UNIVERSIDAD DE SEVILLA

DEPARTAMENTO DE ELECTRÓNICA Y ELECTROMAGNETISMO

DISEÑO CMOS DE UN SISTEMA DE VISIÓN “ON-CHIP” PARA APLICACIONES DE MUY ALTA VELOCIDAD

Memoria presentada por:

FRANCISCO JOSÉ JIMÉNEZ GARRIDO

Para optar al grado de Doctor

Sevilla, 2015

Diseño CMOS de un Sistema de Visión “on-Chip” para Aplicaciones de muy Alta Velocidad

Memoria presentada por:

FRANCISCO JOSÉ JIMÉNEZ GARRIDO

para optar al grado de Doctor

Sevilla, Octubre de 2015

Dirigida por:

Ángel Rodríguez Vázquez

Rafael Domínguez Castro

Tutora:

Belén Pérez Berdú

Departamento de Electrónica y Electromagnetismo

Universidad de Sevilla

CONTENTS

RESUMEN DE LA TESIS

Esta Tesis presenta arquitecturas, circuitos y chips para el diseño de **SENSORES DE VISIÓN CMOS** con procesamiento paralelo embebido. La Tesis reporta dos chips, en concreto:

- El chip Q-Eye
- El chip Eye-RIS_VSoC

y dos sistemas de visión construidos con estos chips y otros sistemas "off-chip" adicionales, como FPGAs, en concreto:

- El sistema Eye-RIS_v1
- El sistema Eye-RIS_v2

Estos chips y sistemas están concebidos para ejecutar tareas de visión a muy alta velocidad y con consumos de potencia moderados. Los sistemas resultantes son, además, compactos y por lo tanto ventajosos en términos del factor SWaP cuando se los compara con arquitecturas convencionales formadas por sensores de imágenes convencionales seguidos de procesadores digitales. La clave de estas ventajas en términos de SWaP y velocidad radica en el uso de sensores-procesadores, en lugar de meros sensores, en la interface de los sistemas de visión. Estos sensores-procesadores embeben procesadores programables de señal-mixta dentro del pixel y son capaces por tanto de adquirir imágenes como de pre-procesarlas para extraer características, eliminar información redundante y reducir el número de datos que se transmiten fuera del sensor para su procesamiento ulterior.

El núcleo de la tesis es el sensor-procesador Q-Eye, que se usa como interface en los sistemas Eye-RIS. Este sensor-procesador embebe una arquitectura de procesamiento formada por procesadores de señal-mixta distribuidos por pixel. Sus píxeles son por tanto estructuras multi-funcionales complejas. De hecho, son programables, incorporan memorias e interactúan con sus vecinos para realizar una variedad de operaciones, tales como:

- Convoluciones lineales con máscaras programables.
- Difusiones controladas por tiempo y nivel de señal, a través de un "grid" resistivo embebido en el plano focal.
- Aritmética de imágenes.
- Flujo de programación dependiente de la señal.
- Conversión entre los dominios de datos: imagen en escala de grises e imagen binaria
- Operaciones lógicas en imágenes binarias.
- Operaciones morfológicas en imágenes binarias.
- etc.

Con respecto a otros píxeles multi-función y sensores-procesadore anteriores, el Q-Eye reporta entre otras las siguientes ventajas:

- mayor calidad de la imagen y mejores prestaciones de las funcionalidades embebidas en el chip;
- mayor velocidad de operación y mejor gestión de la energía disponible;
- mayor versatilidad para integración en sistemas de visión industrial.

De hecho, los sistemas Eye-RIS son los primeros sistemas de visión industriales dotados de las siguientes características:

- Procesamiento paralelo distribuido y progresivo.
- Procesadores de señal-mixta fiables, robustos y con errores controlados.
- Programabilidad distribuida.

La Tesis incluye descripciones detalladas de la arquitectura y los circuitos usados en el pixel del Q-Eye, del propio chip Q-Eye y de los sistemas de visión construidos en base a este chip. Se incluyen también ejemplos de los distintos chips en operación.

THESIS SUMMARY

This Thesis presents architectures, circuits and chips for the implementation of **CMOS VISION SENSORS** with embedded parallel processing. The Thesis reports two chips, namely:

- Q-eye chip,
- Eye-RIS_VSoC chip,

and two vision systems realized by using these chips and some additional "off-chip" circuitry, such as FPGAs. These vision systems are:

- Eye-RIS_v1 system,
- Eye-RIS_v2 system.

The chips and systems reported in the Thesis are conceived to perform vision tasks at very high speed and with moderate power consumption. The proposed vision systems are also compact and advantageous in terms of SWaP factors as compared with conventional architectures consisting of standard image sensor followed by digital processors. The key of these advantages in terms of SWaP and speed lies in the use of sensors-processors, rather than mere sensors, in the front-end interface of vision systems. These sensors-processors embed mixed-signal programmable processors inside the pixel. Therefore, they are able to acquire images and process them to extract the features, removing the redundant information and reducing the data throughput for later processing.

The core of the Thesis is the sensor-processor Q-Eye, which is used as front-end in the Eye-RIS systems. This sensor-processor embeds a processing architecture composed by mixed-signal processors distributed per pixel. Then, its pixels are complex multi-functional structures. In fact, they are programmable, incorporate memories and interact with its neighbors in order to carry out a set of operations, including:

- Linear convolutions with programmable linear masks.
- Time- and signal-controlled diffusions (by means of an embedded resistive grid).
- Image arithmetic.
- Signal-dependent data scheduling.
- Gray-scale to binary transformation.
- Logic operation on binary images.
- Mathematical morphology on binary images.
- etc.

As compared with previous multi-function pixels and sensors-processors, the Q-Eye brings among other the following advantages:

- higher image quality and better performances of functionalities embedded on chip;
- higher operation speed and better management of energy budget;
- more versatility for integration in industrial vision systems.

In fact, the Eye-RIS systems are the first industrial vision systems equipped with the following characteristics:

- Parallel distributed and progressive processing.
- Reliable, robust mixed-signal processors with handled errors.
- Distributed programmability.

This Thesis includes detailed descriptions of architecture and circuits used in the Q-Eye pixel, in the Q-Eye chip itself and in the vision systems developed based on this chip. Also, several examples of chips and systems in operation are presented.

RESUMEN EXTENDIDO, MOTIVACIONES Y CONCLUSIONES

1. INTRODUCCIÓN

Esta Tesis aborda aspectos del diseño de circuitos integrados CMOS para la adquisición y el procesamiento de imágenes. Los **SENSORES DE IMÁGENES** integrados en tecnologías CMOS se identifican por las siglas CIS (CMOS Image Sensors). Estos sensores generan una imagen cuando son irradiados por luz en las bandas del visible y del infrarrojo cercano, aproximadamente [400nm < λ < 1000nm].

La captura de imágenes por sensores CIS implica, entre otros procesos, la *absorción* de los fotones y la *transformación* fotón-carga eléctrica. Estos dos procesos ocurren en dispositivos foto-sensores realizados dentro de un substrato semiconductor, en concreto Silicio. Por tanto este substrato constituye un “plano” activo, fotosensible, sobre el cual la luz es enfocada por una conjunto óptico formado por lentes para crear imágenes nítidas. Esta es la razón por la cual suele usarse el término *plano focal* para referirse a dicho plano.

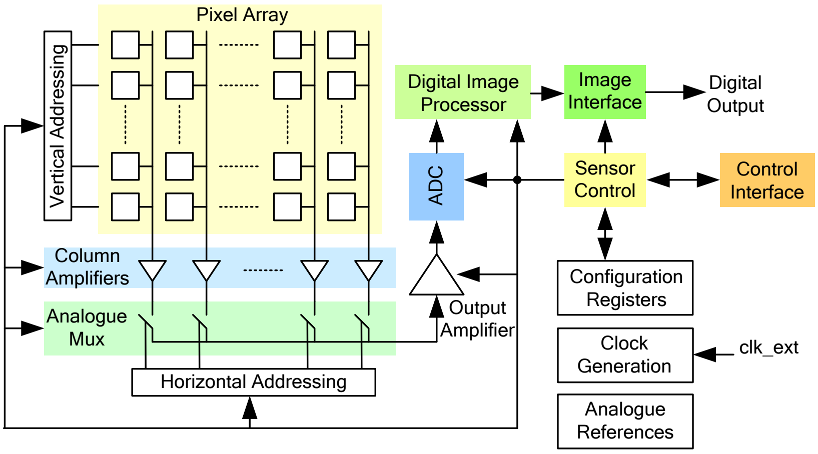
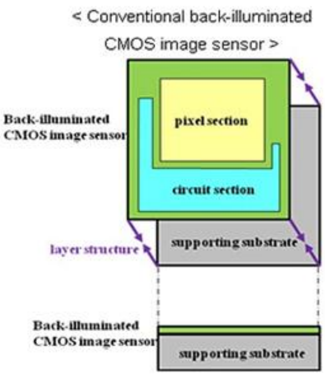


FIGURA 1. SENSOR DE IMAGEN CMOS DIGITAL.

Todas los bloques se supone que están realizados en el mismo substrato semiconductor

El plano focal incluye además circuitos no foto-sensibles, empleados con distintos propósitos. Esto se ilustra en la Figura 1, que describe conceptualmente un diagrama de bloques de un sistema CIS concebido para adquirir imágenes y codificarlas digitalmente [ANAFOCUS]. Cuando los CISs se fabrican en una *tecnología planar* [1], donde todos los circuitos están implementados en el plano focal, sólo la matriz de *píxeles* debe ser foto-sensible. Esta es la parte etiquetada “Pixel Array”, sobre fondo amarillo en la Figura 1. La matriz de píxeles consiste en un conjunto de dispositivos foto-sensores ubicados en una estructura regular, cada uno de los cuales captura una *muestra espacial* de la imagen incidente. El resto de los bloques del sensor se distribuyen por la periferia de la matriz foto-sensora y deben ser cubiertos ópticamente con objeto de reducir el impacto de la generación parásita de pares electrón-hueco foto-inducidos fuera del área foto-sensible o activa.

Los píxeles en sí incluyen también dispositivos foto-sensibles y circuitos no foto-sensibles. La figura de la derecha muestra conceptualmente la distribución de dispositivos en un pixel CIS,



[1] El término planar indica que solo hay un substrato activo, por contraposición a las tecnologías con varios substratos interconectados verticalmente.

donde el área activa, marcada en amarillo, constituye aproximadamente el 45% del área total del pixel. Este porcentaje se denomina “*fill factor*”, mientras que el término “‘*pixel pitch*” se usa para definir las dimensiones del pixel.

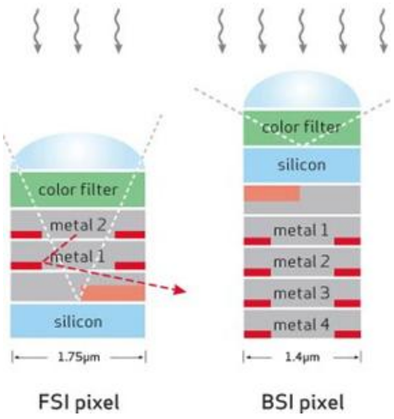
Reducir el “pixel pitch” y el aumentar el “fill factor” marcan tendencia en las actividades académicas e industriales sobre el diseño de CIS. Estas tendencias implican reducir la circuitería no foto-sensibles embebida en el pixel y dominan la investigación y el desarrollo de sensores para electrónica de *consumo* [Font11] [ISW13] [IS14] [Elmg14]. Sin embargo, en aplicaciones de “*machine vision*” los sensores de imágenes pueden contener una parte significativa de circuitería no foto-sensibles dentro del pixel. Los chips en esta Tesis pertenecen a este último grupo. En concreto, dichos chips incluyen circuitería no foto-sensible en el plano focal y el pixel para procesar las imágenes concurrentemente con su adquisición. Esta es la razón por la cual se usa el término **SENSOR-PROCESADOR EN PLANO FOCAL** [Rosk01] [Zara11] para referirse a este tipo de circuitos.

En general, los circuitos empleados en el acondicionamiento y procesamiento de señal en los sensores de imagen pueden ser integrados a diferentes niveles [Ohta08]:

- pixel,
- columna,
- chip.

Los chips presentados en esta Tesis combinan estas tres estrategias [Rodr10b].

Los sensores CIS pueden ser iluminados por el frente (Front Side Illumination – FSI), o desde atrás, a través del substrato (Back Side Illumination – BSI). Las diferencias entre estos dos métodos se ilustran, para un único pixel, en la figura de la derecha. En el caso BSI, la superficie posterior de la oblea de semiconductor incluye micro-lentes ópticas y la luz alcanza directamente el silicio, donde los foto-sensores están integrados, sin interferir con las capas empleadas para interconexión y aislamiento en el IC. En el caso FSI, las micro-lentes ópticas son emplazadas sobre la superficie frontal del chip y la luz debe propagarse a través de las capas y estructuras que constituyen el circuito integrado antes de alcanzar el dispositivo foto-sensible localizado en el substrato de la oblea. El estudio de las ventajas y desventajas de ambas alternativas no es objeto de esta Tesis. Los lectores interesados pueden consultar, por ejemplo, [Prov11]. Los chips descritos en esta Tesis han sido concebidos para operar en modo FSI.



Los sensores de imágenes son las interfaces de entrada (“front-end”) de las **CÁMARAS** y los **SISTEMAS DE VISIÓN**. . Los sensores de imagen suelen estar por tanto *embebido*s en una cámara o en un sistema de visión. La Figura 2 muestra el diagrama de bloques de una cámara incluyendo la óptica, la interfaz de entrada constituida por el sensor de imagen, y un número de bloques encargados del control, el almacenamiento de datos y las comunicaciones.

Aparte de los sensores de imágenes y todas las estructuras funcionales requeridas para el control, comunicación e integración del sistema, las cámaras pueden incluir circuitos y sistemas para *analizar* el flujo de imágenes y *extraer información* del mismo. Estos sistemas son entes complejos tanto en términos de “*hardware*” como de “*software”* y su diseño requiere explotar las sinergias entre ambas facetas — “hardware-software co-design”. En cualquier caso, conviene recordar que por muy compleja que sea una cámara o un sistema de visión, su calidad y capacidad de operación van a depender críticamente de las prestaciones del sensor de imagen usado para la interface de entrada, ya que la calidad de las imágenes adquiridas está básicamente determinada por dicho sensor.

Aunque las contribuciones de esta Tesis, están mayormente en el diseño de chips, las consideraciones sobre el diseño de cámaras y sistemas de visión, incluyendo “*smart cameras”* [Belb09], están presentes en todos los capítulos, con énfasis especial en el Capítulo 5. A modo de ejemplo, la figura del margen izquierdo muestra una de las versiones del sistema de visión Eye-RIS descrito en dicho capítulo. Destaca la compacidad de un sistema que tiene la capacidad de ejecutar tareas de visión de forma autónoma. Esta compacidad, junto con la operación a muy alta velocidad son atributos conferidos por las arquitecturas y los circuitos propuestos en la Tesis.

En este punto es pertinente preguntarnos: ¿por qué merece la pena abordar y estudiar los sensores de imágenes, cámaras y sistemas de visión y las tecnologías relacionadas? La experiencia de la vida cotidiana nos proporciona una respuesta intuitiva. No hay duda de que la capacidad de capturar, analizar e interpretar imágenes ópticas es crucial para la interacción con el entorno de cualquier sistema, natural o artificial, incluyendo los seres humanos. De hecho, los sensores de imágenes de estado sólido se utilizan en un número cada vez mayor de aplicaciones, con una previsión de volumen global de mercado de aproximadamente 3 mil millones de unidades para el año 2015 [Yole14a-b]. Además de la amplia utilización de dispositivos de visión en teléfonos móviles y dispositivos electrónicos personales, otros sectores de aplicación con un fuerte potencial de crecimiento están evolucionando rápidamente, tales como medicina, defensa, automoción, vigilancia, robótica, industria y similares.

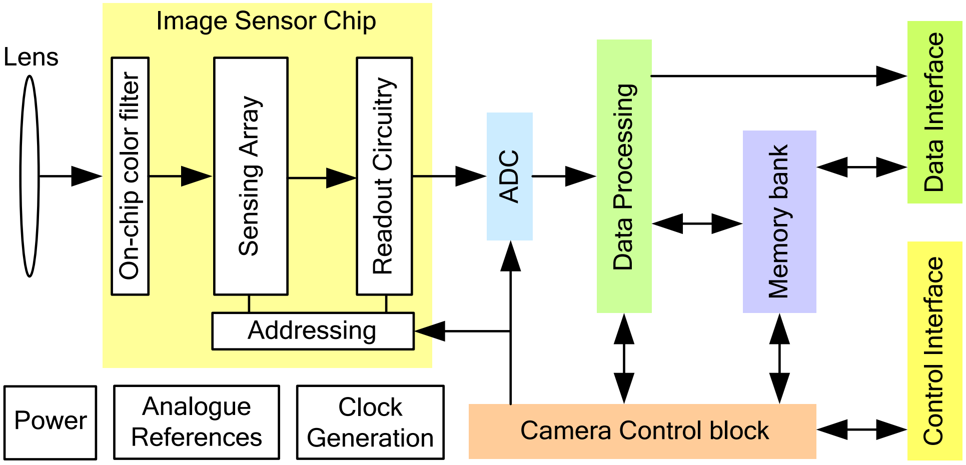


FIGURA 2. DIAGRAMA DE BLOQUES DE UNA CÁMARA.

Entre otras razones, la actual proliferación de sensores de imagen de estado sólido está impulsada por las capacidades de las tecnologías CMOS modernas para integrar sistemas con tamaño reducido, poco peso y consumo de potencia, gran velocidad de operación, y amplias capacidades funcionales y flexibilidad. Estos sensores CIS son parte esencial de los modernos sistemas InfoTech *portátiles*, y una característica esencial en estos sistemas portátiles es poseer un factor SWaP (Size, Weight and Power) pequeño — conferido por el uso de tecnologías CMOS.

Sin embargo, hasta aproximadamente mediados de la década de los noventa, la expansión de los sistemas de visión CMOS se vio obstaculizado por la menor calidad de los dispositivos foto-sensores compatibles con estas tecnologías en comparación con las tecnologías CCD (Charge Coupled Devices) [Naka06] [Ohta08]. En resumen, las imágenes captadas por los foto-sensores CMOS primitivos no eran lo suficientemente buenas para numerosas aplicaciones industriales. En tal escenario las tecnologías CCD dominaron la realización industrial de cámaras y sistemas de visión y las CMOS quedaron relegadas a un segundo plano. Esta desventaja se atenuó de forma significativa con la mejora de los foto-sensores CMOS. Por ejemplo con los "*pinned photodiode"s* se alcanzaron niveles de calidad similares al CCD, haciendo viable al CMOS para múltiples aplicaciones prácticas. De hecho, pese a que las tecnologías CCD puedan quizás adquirir mejores imágenes, las ventajas de las tecnologías CMOS en términos de tamaño, peso, consumo de potencia, coste de producción, velocidad y capacidad para la integración de funcionalidades en sistemas embebidos, entre otras, han sido decisivas para que los sensores CMOS hayan ganado finalmente el duelo frente a los CCDs.

Actualmente, los sistemas de visión CMOS copan más del 90% del mercado de sensores de imagen de área, donde el término *sensor de área* se usa en contraposición a *sensor de línea* o sensor lineal. El primero captura las señales ópticas usando una matríz bidimensional de detectores, mientras que el segundo emplea un vector unidimensional de detectores, como los requeridos para aplicaciones de *escaneado*. Además, el volumen total de mercado para sistemas de visión CMOS está previsto que crezca significativamente durante los próximos años, con una previsión total de mercado próxima a los 17 billones para 2018 [Yole14a-b]. La Figura 3 muestra los volúmenes de producción para sistemas CIS en diferentes campos de aplicación. Las principales ventajas de las tecnologías CMOS pueden resumirse en los siguientes puntos:

- Coste,
- Tamaño,
- Consumo de potencia,
- Velocidad,
- Flexibilidad funcional.

Esta Tesis se centra en el estudio de los sistemas de vision basados en tecnología CMOS y no pretende analizar ni cuantificar las ventajas y desventajas de los sistemas CMOS frente a los sistemas CCD. Los lectores interesados en sensores de imágenes basados en tecnologías CCD pueden consultar, entre otras muchas referencias, a [Theu95].

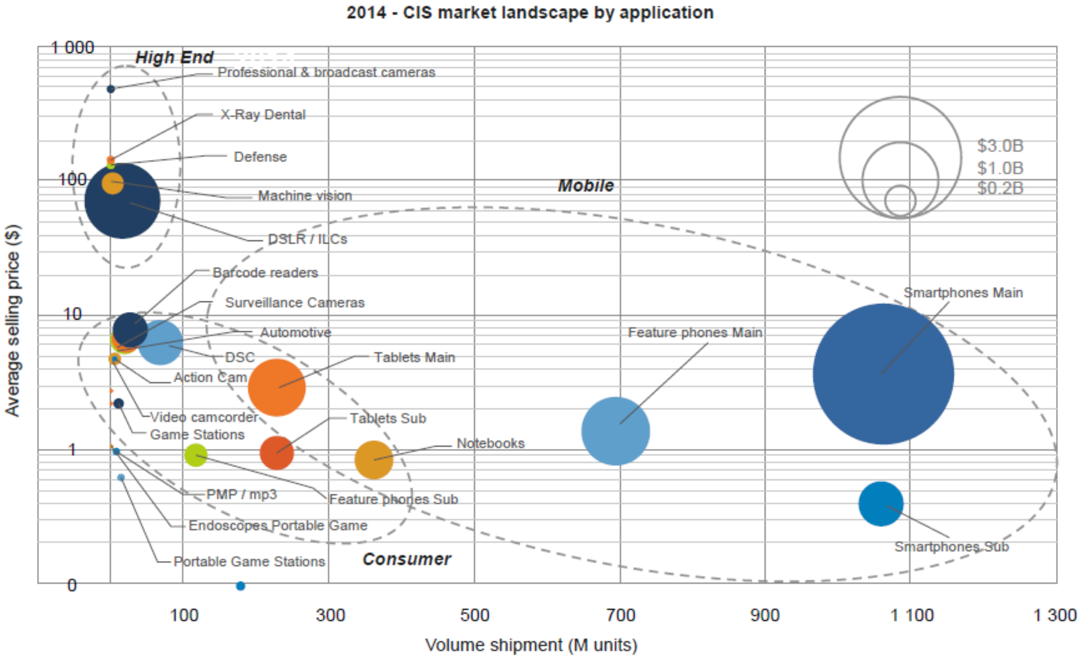


FIGURA 3. PREVISIÓN DE VENTAS DE SENSORES CIS POR CAMPOS DE APLICACIÓN

Predicción realizada por Yole Development

Los avances tecnológicos en los sistemas de visión CMOS han sido impulsados principalmente por las aplicaciones de consumo. Estas condicionan también los desafíos técnicos de industrias y universidades en relación con los sistemas CIS. Como ya se mencionó, el mercado de consumo demanda píxeles con un "pitch" cada vez menor y con *resolución espacial* (número de píxeles) cada vez mayor. Además de este aumento en la resolución, otros retos tecnológicos para los sensores de imágenes CMOS son:

a) mejora de la calidad de imagen a través de la mejora de los circuitos analógicos de lectura; acondicionamiento de la señal, conversión AD (Analógico/Digital) y procesamiento digital de la señal;

b) aumento de la velocidad de transmisión de datos, mejorando los circuitos de comunicación;

c) reducción del área, potencia y coste a través de la integración de circuitos embebidos en un único chip ("system on chip") [ISW13] [Elmg14] [IS14].

Los avances recientes incluyen sensores de varios millones de píxeles con "pixel-pitch" del orden de 1µm, velocidad de datos superior a los 10Gpx/s [Taka13], arquitecturas de lectura y conversión AD reconfigurables [Leñe14] [Kawa08], corrección de imagen [Chen12], gestión térmica y energética, etc.

Todos los desafíos tecnológicos citados anteriormente se centran en la adquisición de matrices de intensidad de luz que definen *imágenes 2-D*. Durante los últimos años también se han desarrollado técnicas y arquitecturas con el objetivo de incluir en los sistemas de visión la estimación de profundidad, es decir, *información 3-D*. Uno de los principales impulsores de estas actividades es el desarrollo de interfaces hombre-máquina para la industria del entretenimiento [Payb14], pero el potencial de estas tecnologías es enorme en campos de como vigilancia, inspección industrial, automoción y medicina. Además de las técnicas basadas en la estereoscopía, la triangulación y similares, se están haciendo esfuerzos significativos en el desarrollo de píxeles CMOS que sean capaces de capturar la información de profundidad a través de técnicas de *tiempo de vuelo* (Time of Flight). Entre las diferentes aproximaciones, los diodos de avalancha, Single Photon Avalanche Diodes (SPADs), están siendo objeto de un creciente interés [Seit11]. Aparte de tecnologías de pixel específicas para ToF (tales como SPADs, doble puerta de transferencia, etc.), las medidas de tiempo de vuelo requieren complejos esquemas de iluminación [Nicl11]— todavía lejos de una solución integrada en un solo chip para capturar imágenes en 3-D. Por otra parte, los sensores 3-D incorporan a fecha de hoy muchas menos capacidades de procesamiento "on-chip" que los actuales sensores de visión 2-D. En esta Tesis, no se abordan problemáticas relacionadas con la adquisición de imágenes 3-D.

Retomando el hilo de las ventajas del uso de tecnologías CMOS, tal vez la más relevante está conferida por la capacidad para diseñar sistemas de visión completos en un único chip ("system-on-chip") en lugar de sólo sensores de imagen. El concepto de cámara integrada en un único chip ("camera-on-chip") fue introducido en [Foss97]. Con referencia al diagrama de bloques de la Figura 2, este concepto implica la incorporación en un único substrato semiconductor del máximo número posible de los bloques funcionales necesarios para implementar una cámara. Por tanto, los sistemas de visión CMOS integrados en un único chip pertenecen a la categoría de sensores inteligentes — "smart sensors"; por ejemplo, sensores con inteligencia embebida [Meij01].

El concepto **SENSORES INTELIGENTES** es genérico y ambiguo. En general, la característica '*inteligente*' hace referencia a cualquier tipo de circuito empleado para procesar las señales adquiridas por los sensores, ya sea en funciones como la lectura de datos y la corrección de errores, o para la extracción de información a partir de las imágenes. En la práctica pueden considerarse diferentes niveles de inteligencia. El más bajo implica básicamente funciones relacionadas con la lectura de las imágenes, corrección de errores y comunicaciones, que son las más utilizadas actualmente en la industria. En el mercado encontramos sensores de imágenes inteligentes con microprocesadores embebidos que generan imágenes de alta calidad completamente corregidas a velocidades del orden de miles de imágenes por segundo (Frames-per-Second, FpS) y con niveles de ruido equivalentes inferiores a un electrón [ANAFOCUS]. Niveles de inteligencia superiores, como los requeridos para extraer e interpretar la información contenida en las imágenes, y generar las decisiones y acciones correspondientes, han sido estudiados y desarrollados en el ámbito académico durante años [Eklu96] [Rodr04] [Abbo08] [Fern11] [Lopi11], y es solo ahora cuando empiezan a implementarse en sistemas industriales [ANAFOCUS]. Entre las razones de este retraso podemos citar la falta de estandarización en las soluciones arquitecturales, en su programación y en los formatos de salida [Khro13]. Sin embargo, los problemas arquitecturales pueden abordarse mediante el uso de arquitecturas *celulares* y, en particular, a través de la utilización del paradigma de *redes celulares no lineales* (Cellular Nonlinear Network, CNN. Este paradigma se adapta a las características específicas del problema, en conreto:

- gran cantidad de datos con una estructura matricial regular que contiene propiedades e información distribuida en la globalidad, y
- que pueden ser extraídas a través de interacciones y operaciones locales,
- basadas en la propagación y procesamiento de señales en el espacio, a lo largo y ancho de cada imagen, y en el tiempo, a lo largo de la secuencia de imágenes [Rosk01] [Chua02].

Dependiendo del nivel de inteligengia integrada, distinguiremos dos tipos de sensores-procesadores CMOS:

- Sensores de imágenes CMOS inteligentes (Smart CMOS Image Sensors) denominados SCIS.
- Sensores de visión CMOS (CMOS VIsion Sensors) denominados CVIS.

La Figura 4 ilustra las diferencias entre SCIS, por un lado, y CVIS, por otro lado. Aunque las entradas de ambos circuitos integrados son imágenes capturadas por foto-sensores localizados en el plano focal, sus salidas son de naturaleza diferente.

En el caso de los sistemas SCIS, los resultados básicos de la cadena de procesamiento son imágenes; bien en escala de grises o en color, en formato analógico o digital, sin o con correcciones, etc. Por el contrario, los resultados de los sistemas CVIS pueden no ser imágenes sino *características* [2] o incluso decisiones basadas en el análisis espacio-temporal de la información contenida en la escena.

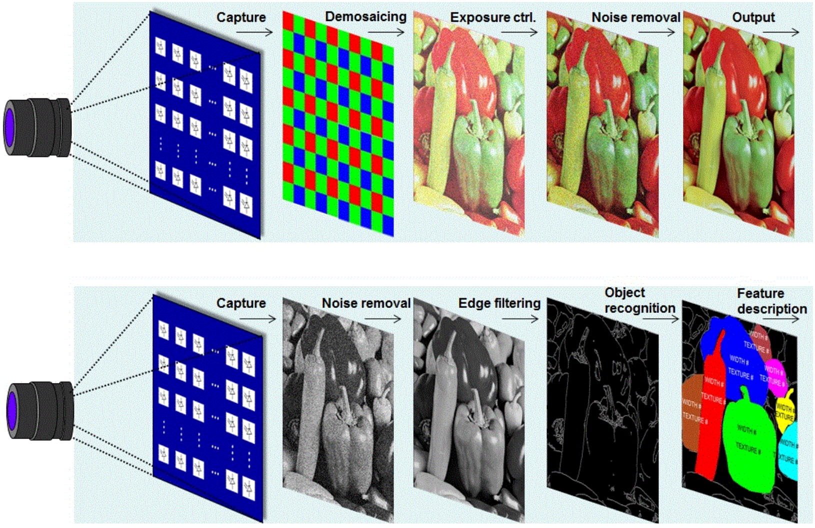


FIGURA 4. ILUSTRACIÓN DE LAS DIFERENCIAS ENTRE LOS SISTEMAS SCIS (ARRIBA) Y CVIS (ABAJO)

Pese a las diferencias funcionales, los sistemas SCISs y CVISs son estructuralmente similares en que ambos combinan dos tipos diferentes de sub-sistemas integrados en un substrato de silicio común:

- Los usados para la adquisición de imágenes: por ejemplo, la captura de la potencia lumínica y su transducción en tensión o corriente. Estos consisten esencialmente en uniones *pn* polarizadas en inversa (*foto-diodos*) y en circuitos de *señal-mixta* que controlan la operación del sensor.
- Los que se utilizan para la lectura de los resultados del proceso de transducción de la luz, acondicionamiento de la señal y conversión AD, los empleados en la corrección y procesamiento, en el control global del sistema y las comunicaciones. Estos sub-sistemas están constituidos por circuitos analógicos, de señal-mixta y circuitos puramente digitales.

Estas similitudes pueden ser explotadas durante la fase de diseño ya que muchos conceptos, arquitecturas y circuitos son compartidos por ambos tipos de sistemas y pueden ser reutilizados por los diseñadores. Este hecho queda reflejado a lo largo de la Tesis.

2. RESUMEN DE LA TESIS

Como se ha mencionado anteriormente, el mercado de sistemas de visión, incluyendo sectores como vigilancia, visión industrial, sistemas de transporte inteligentes, electrónica de consumo, guiado de vehículos, redes inalámbricas de sensores, etc., demanda:

- reducción de los costes de producción a nivel de sistema;
- aumento del nivel de integración, con implementaciones físicas compactas, de tamaños reducidos y con el menor número posible de componente;

[2] Características de una imagen pueden ser interpretadas como manifestaciones de la información contenida en una imagen; por ejemplo, el número de objetos presentes en una imagen, localización de objetos, etc.

- mayor calidad de la imagen y mejores prestaciones de las funcionalidades embebidas en los chips;
- mayor velocidad de operación;
- mejor uso posible del presupuesto de potencia disponible.
- etc.

Esta Tesis describe sensores de visión CMOS (CVISs), y sistemas basados en dichos chips, para alcanzar los objetivos anteriores mediante:

- arquitecturas concebidas para el ***procesamiento paralelo*** de las imágenes;
- el uso de **procesadores de señal-mixta**, en lugar de meramente digitales, **distribuidos** por pixel.

Las razones que subyacen a estas soluciones se detallan a continuación.

El **PROCESAMIENTO PARALELO** es muy adecuado para manejar la gran cantidad de datos presentes en las primeras etapas de la cadena de procesamiento de imágenes. Tal como se ilustra en la Figura 5, en las etapas iniciales de la cadena de procesamiento de la visión, el número de datos puede ser muy elevado pues es proporcional a la resolución espacial y al número de bits por dato de píxel. Además, muchos de los datos son redundantes y por lo tanto inútiles para los fines de extracción de información y la realización de tareas de visión. Sin embargo, el número de datos decrece a medida que se ejecutan los flujos de información y detección de características a través de la cadena de procesamiento. En realidad, en la toma de decisión final del sistema de visión están implicados muy pocos datos. Por ejemplo, en aplicaciones de vigilancia, las decisiones van a depender principalmente de si la escena contiene algún objeto sospechoso o no.

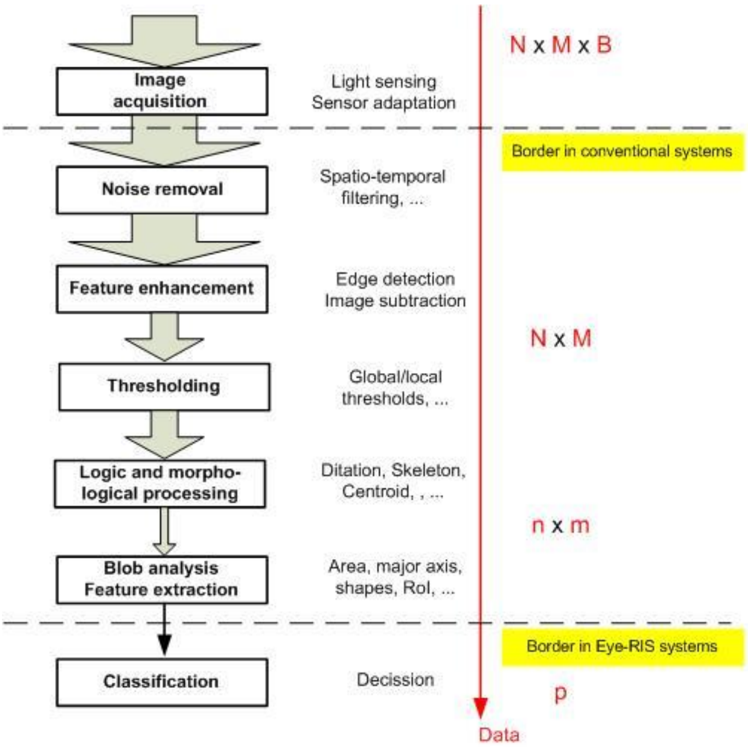


Image acquisition
Light sensing
Sensor adaptation
N x M x B
Border in conventional systems
Noise removal
Spatio-temporal filtering, ...
Feature enhancement
Edge detection
Image subtraction
N x M
Thresholding
Global/local thresholds, ...
Logic and morpho-logical processing
Ditation, Skeleton, Centroid, , ...
n x m
Blob analysis
Feature extraction
Area, major axis, shapes, RoI, ...
Border in Eye-RIS systems
Classification
Decission
p
Data

FIGURA 5.- CADENA DE PROCESAMIENTO EN ALGORITMOS DE VISIÓN.

Empezando arriba con las imágenes captadas por el sensor (datos en crudo), la cantidad de datos decrece y el nivel de abstracción aumenta. N representa el número de filas, M el número de columnas y B el número de bits usado por dato de pixel: $n<N$; $m>M$ and $p<(n,m)$. Las arquitecturas propuestas en esta Tesis mapeas esta estructura de datos en capas mediante el uso de estrategias de procesamiento específicas para cada paso de la cadena.

En las arquitecturas de sistemas de visión convencionales la frontera entre sensores y procesadores está localizada en un punto done la cantidad de datos a transmitir es grande (ver la linea discontinua en la parte superior de la Figura 5). De hecho, los sistemas convencionales descargan imágenes completas de los sensores para su posterior procesamiento, por lo que requieren una cantidad significativa de memoria y tienen una demanda computacional elevada. Por ejemplo, para una imagen con formato de alta definición para TV, con una resolución espacial 1920x1080 y una longitud de palabra de 10 bits por pixel, el número de datos asociados a una escena ("frame") es 20.74Mbits, equivalente a 2.6Mbytes. Por tanto, la velocidad del flujo de datos asociada a la transmisión de esta imagen en un sistema operando según el estandar de video (30FpS) es 622Mbits por segundo. Para sistemas operando a altas velocidades, por ejemplo las aplicaciones de inspección industrial pueden requerir 1000FpS, el tratamiento de esta gran cantidad de datos puede llegar a convertirse en un obstáculo insuperable.

Sin embargo, usando sistemas CVISs como los propuestos en esta Tesis, la frontera o interfaz entre el sensor [3] y los procesadores digitales está localizada en un punto donde la cantidad de datos a transmitir es pequeña, pues se transmiten características presentes en la escena en lugar de los datos de pixeles completos de la imagen (ver la linea discontinua en la parte inferior de la Figura 5). Supongamos que nuestra aplicación objetivo consiste en seguir objetos que se mueven a una velocidad de 40m/s en la escena. Esto requiere capturar y analizar imágenes a una razón de 2000Fps. En el resultado del proceso de captura y análisis el único dato pertinente es la posición predicha de los objetos. Esta es en realidad la única información transmitida al procesador digital. Pero para extraer dicha información se deben realizar antes las siguientes tareas:

- Adquisición de imágenes,
- filtros paso de baja,
- detección de actividad,
- estimación de movimiento,
- seguimiento del objeto,
- lazo de control y
- predicción de posición.

Los sistema CVISs propuestos en esta Tesis implementan estas y otras tareas directamente en el plano focal, evitando así los cuellos de bolleta en la comunicación de los datos y reduciendo los requerimiento de potencia de cálculo memorias asociados a las aqruitecturas convencionales. Consideremos por ejemplo el caso de una operación de *convolución* para un filtrado espacial linear. Esta operación implica 9 multiplicaciones y ocho sumas por pixel. Con lo cual, para el ejemplo previo del estándar de alta definición para TV (resolución espacial 1920x1080 a 30FpS) la velocidad de procesamiento requerida sería 1.06GOpS (<u>G</u>iga <u>O</u>perations <u>p</u>er <u>S</u>econd). Y el filtrado es solo una de las muchas tareas que deben ser llevadas a cabo en las primeras etapas del procesamiento de imágenes.

En resumen, además de los problemas en la comunicación y almacenamiento de datos, los sistemas de visión convencionales neecistan potencias de cálculo elevadas. Es obvio que el problema es abordable con las tecnologías digitales actuales siempre que no existan limitaciones en el consumo de potencia y en el tamaño del sistema. Pero bajo fuertes restricciones de consumo de potencia, tamaño y velocidad de operación, es conveniente considerar arquitecturas alternativas como las porpuestas en esta Tesis.

Además del procesamiento paralelo, otra característica distintiva de las soluciones propuestas en esta Tesis, es el uso de **CIRCUITOS DE SEÑAL-MIXTA** para procesamiento. Esto circuitos operan a mayor velocidad y con mayor eficiencia energética que sus homólogos digitales, bajo el supuesto de que los requisitos de *precisión* sean moderados. Además, procesar en paralelo con circuitos analógicos es ventajoso, frente a la alternativa digital, en términos de la cantidad de operaciones realizables para un consumo de potencia y una velocidad de operación dadas [Carm13]. En resumen, las arquitecturas de sistemas de visión propuestas en esta disertación superan a las soluciones convencionales en términos de la figura de mérito SWaP así como en términos de la relación entre la potencia-de-cálculo y el consumo-de-energía. Estas características, junto con el coste, resultan ideales para reducir la brecha que existe entre los detectores binarios y los detectores de área en aplicaciones de inspección y visión artificial —

[3] Los sistemas CVISs no solo adquieren imágenes sino que las procesan. Por ello, un término más conveniente para ellos es el de sensor-procesador en lugar del de sensor.

ver Figura 6. Es decir, los sistemas propuestos en esta Tesis pueden, por coste, tamaño y velocidad, reemplazar a los sensores uni-dimensionales que se usan actualmente en muchas cadenas de producción, capacidad de procesamiento superiores en varios órdenes de magnitud.

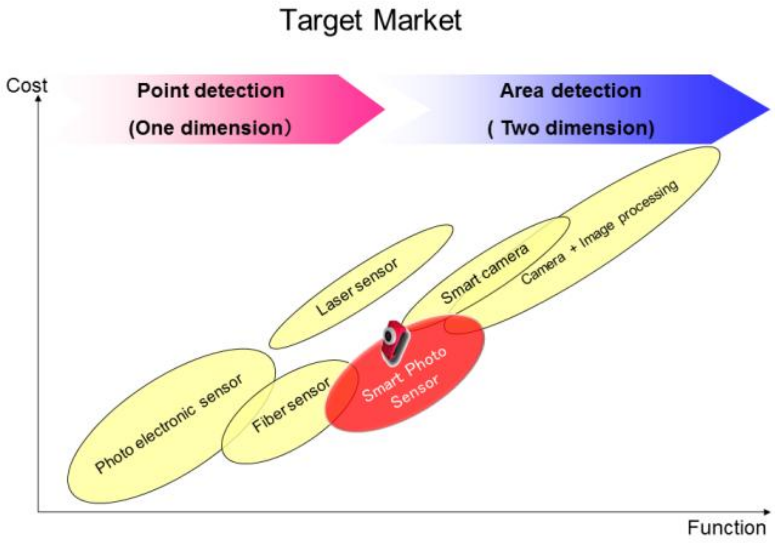


FIGURA 6.- POSICIÓN DEL SENSOR INTELIGENTE TOSHIBA-TELI SPS [TOSH14] EN EL PLANO DE MERCADO OBJETIVO COSTE FRENTE A FUNCIONALIDAD. ESTA CÁMARA DE TOSHIBA-TELI EMPLEA ARQUITECTURAS CVIS CONCEBIDAS EN ESTA TESIS.

Imagen tomada de http://www.toshiba-teli.co.jp/en/products/industrial/sps/sps.htm

El sistema CVIS básico desarrollado en esta Tesis se llama Q-Eye. El píxel implementado en el sensor Q-Eye ha sido dotado de un alto nivel de capacidad de procesamiento y programación con el fin de implementar el mayor número posible de las funciones de procesamiento requeridas en las aplicaciones de *visión temprana* [Toma06]. Esta capacidad de programación del Q-Eye permite el desarrollo de cámaras inteligentes y sistemas de visión embebidos con una funcionalidad versátil. Es decir, el Q-Eye no está diseñado para una tarea específica sino que puede implementar una amplia variedad de algoritmos de visión mediante programación.

Los píxeles del sistema Q-Eye intercambian información con sus vecinos para realizar distintas operaciones, tales como:

- Convoluciones lineales con máscaras programables.
- Difusiones controladas por tiempo y nivel de sañal, a través de un grid resistivo embebido en el plano focal.
- Aritmética de imágenes.
- Flujo de programación dependiente de la señal.
- Conversión entre los dominios de datos: imagen en escala de grises e imagen binaria
- Operaciones lógicas en imágenes binarias.
- Operaciones morfológicas en imágenes binarias.
- etc.

El **CAPÍTULO 3** proporciona una descripción detallada de la arquitectura del píxel del Q-Eye. Esto incluye esquemas para todos los circuitos integrados dentro del píxel, la descripción del comportamiento nominal de los circuitos, el análisis de los errores principales de los circuitos, y ejemplos del impacto de estos errores en la operación del sistema.

El **SISTEMA DE VISIÓN CMOS Q-EYE** es descrito en el **CAPÍTULO 4**, donde se explica en detalle los puertos de comunicación, la programabilidad del sistema, las arquitecturas de los canales de lectura y los bloques misceláneos. Este Capítulo 4 introduce también los aspectos a nivel de sistema que deben considerarse durante la fase de diseño.

La Tesis también describe el desarrollo de sistemas de visión basados en el sensor CVIS Q-Eye. Las diferencias entre estos sistemas y sistemas de visión convencionales han sido presentadas anteriormente haciendo referencia a la localización de la interfaz de datos entre la interfaz física de entrada sensor-procesador y los procesadores digitales estándar dentro de la cadena de procesamiento – ver Figura 5. La Figura 7 y la Figura 8 complementan estas explicaciones.

Por un lado, la Figura 7 corresponde a un sistema de visión convecional. En este sistema, la interface de entrada captura la luz asociada a las imágenes de entrada y trasmite los datos de "frames" completos, representado por el símbolo **F**, para su posterior procesamiento. Como se mencionó anteriormente, la enorme cantidad de datos asociados a una imagen crea cuellos de botella en comunicaciones, memorias y potencia de cómputo.

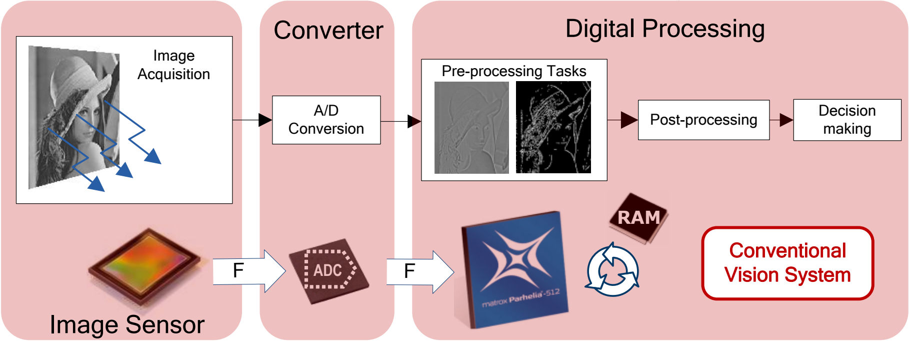


FIGURA 7.- FLUJO DE DATO EN UN SISTEMA DE VISIÓN CONVENCIONAL

Por otro lado, la Figura 8 corresponde a los sistemas de visión considerados en esta Tesis. En este caso, la interfaz de entrada no sólo captura la imagen sino que además procesa la información contenida en la misma. Por tanto, la cantidad de información a transmitir, representada por el símbolo **f** << **F,** a los posteriores bloques de procesamiento es mucho menor. En realidad, en la arquitectura de la Figura 8, el procesamiento se lleva a cabo de forma progresiva, distribuyendose las tareas de procesamiento entre la interfaz de entrada y el núcleo de procesadores digitales. La interfaz de entrada sensor-procesador realiza un pre-procesado de los datos ejecutando los algoritmos de procesamiento en paralelo directamente en la matriz de píxeles. Estos algoritmos deben tratar una gran cantidad de datos y son muy demandantes en términos de potencia y velocidad, por lo que se benefician del uso de circuitos de señal-mixta. Respecto al pos-procesamiento, éste trata una menor cantidad de datos pero requiere lógica compleja con numerosos puntos de decisión, diferentes caminos de ejecución, etc. El pos-procesamiento consiste en complejos algoritmos los cuales requieren programabilidad y re-configurabilidad. La carga computacional no es alta porque la cantidad de datos a procesar es mucho menor, relajándose las especificaciones de los procesadores digitales incluido en el sistema de visión. En definitiva, las diferentes secciones de la arquitectura propuesta se adaptan a la naturaleza de los datos a tratar, y el sistema final es mucho más eficiente que uno convencional.

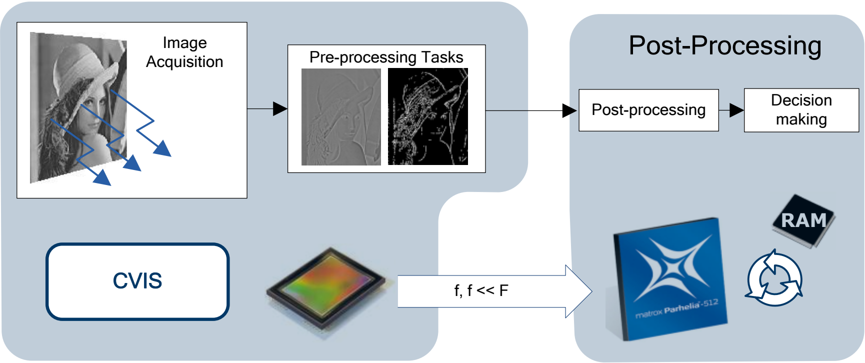


FIGURA 8.- FLUJO DE DATOS EN UN SISTEMA BASADO EN EL CVIS Q-EYE.

En el **CAPÍTULO 5**, se describen dos sistemas de visión desarrollados sobre el Q-Eye, en base al concepto representado en la Figura 8. Dichos sistemas son:

- Eye-RIS v1 y
- Eye-RIS v2 (Vision System on Chip),

Estos sistemas sirven como demostradores de la viabilidad técnica y las ventajas de incorporar CVIS al diseño de sistemas de visión embebidos.

Las diferencias entre ambos sistemas están en el nivel de integración. El segundo sistema (Eye-RIS_v2) integra sobre el mismo silicio el sensor-procesador Q-Eye y el microprocesador RISC, mientras que el primero (Eye-RIS_v1) integra el microprocesador RISC en una FPGA fuera del silicio. El Capítulo 5 incluye también secciones que muestran estos sistemas en operación.

Esta Tesis es completada con un **CAPÍTULO 2** dedicado a introducir conceptos básicos relacionados con la adquisición y procesamiento de imágenes y a explicar los principios de las aproximaciones arquitecturales empleadas a lo largo de esta Tesis.

3. CONCLUSIONES Y CONTRIBUCIONES

Con esta Tesis se demuestra la viabilidad técnica del desarrollo e implementación de sensores CVIS (CMOS Vision Image Sensors) con procesadores paralelos de señal-mixta "per-pixel" y de sistemas de visión basados en dichos sensores para aplicaciones industriales. Las ventajas principales de estos sistemas de visión respecto a las arquitecturas convencionales de sistemas de visión son:

- aumento del nivel de integración de las implementaciones físicas permitiendo en última instancia la realización de sistemas de visión compactos, en un solo chip;
- mayor velocidad de operación;
- mejor uso posible de la potencia y la energía disponibles.

Las contribuciones de la Tesis producen desde la base del diseño e implementación de una arquitectura específica de sistema CVIS – el sensor-procesador Q-Eye. El estudio realizado para el diseño del Q-Eye ha considerado los diferentes niveles de jerarquía y campos de conocimiento que lleva asociado el desarrollo de un sistema de estas características. Dichos niveles van desde la definición de su arquitectura y especificaciones en base a las necesidades determinadas por las aplicaciones objetivo, hasta el análisis y diseño de los bloques de circuito que implementan sus funciones básicas usando técnicas de circuitos integrados.

Partiendo del Q-Eye se han concebido e implementado dos sistemas de visión que usan a dicho CVIS como interface. Estos sistemas son:

- Eye-RIS_v1 y
- Eye-RIS_v2.

Este último es un claro exponente del nivel de integración que puede alcanzarse con las técnicas propuestas en la Tesis pues s trata básicamente de un sistema de visión implementado íntegramente en un único chip (Vision System on Chip).

A continuación se resumen las aportaciones principales de la Tesis:

- Definición de un conjunto de operaciones básicas que pueden implementarse con circuitos a nivel de pixel y que permiten la implementación de las funciones de bajo nivel usualmente utilizadas en las primeras fases del procesamiento de imágenes. Poseer este conjunto de operaciones básicas dota al sensor-procesador de la versatilidad y programabilidad adecuados para desarrollar un elevado número de algoritmos en una amplia gama de aplicaciones. En concreto, el pixel del Q-Eye tiene mayor nivel de programabilidad que la mayoría de sus predecesores, muchos de los cuales incluyen circuitería para desarrollar tareas específicas y no son adecuados por lo tanto para sistemas industriales. Los CVISs ACE y CACE tienen niveles de programabilidad semejantes pero sus implementaciones son mucho más costosa en términos de área y potencia.

- Implementación mediante circuitos de cada una de las operaciones básicas incluidas al nivel de pixel, garantizando un nivel de precisión moderado (8-bits), y optimizando consumos de potencia y área del pixel para conseguir la mayor resolución espacial y el mejor "fill factor" posible. El resultado de ello ha sido que el pixel del Q-Eye tiene uno de los mayores niveles de densidad de dispositivos por área entre todos los pixeles con procesamiento implementados hasta la actualidad. Además, tanto la operación básica como los circuitos que la implementan se han desarrollado para garantizar robustez en la ejecución de los algoritmos de procesamiento. Una operación robusta del sensor significa que los resultados de los algoritmos, y las configuraciones y secuencia de instrucciones asociadas a los mismos, se mantienen tanto de chip a chip, como para condiciones ambientales y de entorno cambiantes dentro de los límites especificados. De hecho, una demostración de esto, es que los sistemas de visión Eye-RIS basados en el sensor inteligente Q-Eye son los primeros sistemas de visión industriales con procesamiento masivamente paralelo de señal-mixta en el plano focal que son fiables, robustos y con errores controlados susceptibles de calibración. Esto ha propiciado su comercialización.
- Con objeto de garantizar la precisión de 8 bits, se ha llevado a cabo un análisis exhaustivo de los errores y no-idealidades asociadas a las implementaciones de las operaciones básicas. También se ha desarrollado una estrategia para implementar las funciones de procesamiento de bajo nivel de manera que se obtenga la menor acumulación posible de errores. Para ello se han generado modelos de las operaciones básicas a nivel de pixel y sus no-idealidades, para posteriormente estudiar el efecto de estos errores en los algoritmos de procesamiento. Esto se traduce en una gran precisión y reproducibilidad de la operación del Q-Eye "en campo".
- En base al pixel desarrollado, se ha diseñado el sensor CVSI Q-Eye en una tecnología 180nm estándar CMOS de 1.8V. Las funcionalidades más relevantes implementadas en el mismo se resumen a continuación:
  - Este sensor-procesador adquiere y procesa imágenes de forma concurrente, extrayendo características a muy alta velocidad a partir de la información espacio-temporal contenida en la secuencia de imágenes. La alta velocidad de operación viene conferida por el uso de procesamiento masivamente paralelo.
  - El Q-Eye se ha implementado en una arquitectura de procesador tipo Single Instruction Multiple Data (SIMD) que saca el máximo rendimiento de la capacidad de procesamiento de la matriz de píxeles multi-función.
  - Las instrucciones a ejecutar por dicha matriz se aplican a través de una interfaz de programación y bloque de control. Estos últimos contienen memorias para almacenar micro-instrucciones, las cuales definen la secuencia de operaciones a desarrollar por la matriz de pixeles con procesamiento embebido. El contenido de dichas memorias puede ser actualizado por un "host" externo. Esta arquitectura permite que el micro-procesador del sistema de visión, en el cual se integra el sensor inteligente diseñado, pueda definir los algoritmos de procesamiento de imágenes de bajo nivel a desarrollar por el Q-Eye, el cual actúa como "hardware" específico de procesamiento o co-procesador.
  - Las imágenes resultantes de este procesamiento son descargadas a través de la interface de imágenes. Esta interface se ha diseñado con un elevado grado de configurabilidad, pudiéndose cargar y descargar tres tipos de imágenes:
    - ✓ Imágenes binarias que representan la información comprimida resultante del procesamiento de bajo nivel y que contienen las características necesarias para el procesamiento de alto nivel.
    - ✓ Coordenadas de eventos o píxeles activos. Este protocolo de descarga es óptimo en velocidad para imágenes con puntos activos dispersos.
    - ✓ Imágenes en niveles de gris con 8 bits de resolución. Esta interfaz se ha definido principalmente con propósito de test. Pero puede usarse como interfaz de entrada salida al utilizar al Q-eye sólo como co-procesador de imágenes en niveles de gris.

  La capacidad de cargar y descargar imágenes permite utilizar al sistema Q-Eye sólo como co-procesador de imágenes (binaria y en niveles de gris)

  - Para la implementación física de la interface de imágenes se han diseñado bloques de señal mixta tales como:
    - ✓ Convertidores AD Flash de 8 bits de resolución.
    - ✓ Convertidores DA de 8 bits de resolución basados en decodificadores.
    - ✓ Comparadores.
    - ✓ Circuitos "Sample & Hold".

  Como resultado del diseño se han descrito los aspectos más importantes a tener en cuenta para el diseño e la implementación de todos los sub-bloques y estructuras funcionales, sus no-idealidades y los efectos de las mismas sobre la calidad de imagen.

- La correcta operación de los circuitos de señal-mixta incluidos en la matriz de procesamiento e interfaz de imágenes requieren tensiones de referencias y corrientes/tensiones de polarización que han sido implementadas internamente a partir de un band-gap de alta precisión, convertidores DA y buffers. En relación con la polarización de los bloques de señal-mixta se ha desarrollado una estrategia de polarización y distribución de las referencias que garantiza un comportamiento robusto de los circuitos.
- Con el objetivo de dotar al sistema Q-Eye con funcionalidades adicionales útiles para la integración en un sistema de visión se ha incluido en el Q-Eye bloques como:
  - Sensor de temperatura
  - Convertidor AD de aproximaciones sucesivas de propósito general conectado a distintos nudos internos que pueden se interesantes para un control automático del tiempo de exposición, procesos de calibración, etc.
  - "Power on Reset" para la aplicación de un "reset" inicial durante el encendido del sistema Q-Eye, garantizando un estado inicial conocido.
  - Unidad de generación de los relojes internos.

  Estos bloques, junto con la generación interna de referencias, se integran en el Q-Eye ("on-chip") para reducir el número de componentes externos ("off-chip") requeridos para la operación del sensor consiguiendo el mayor nivel de compacidad posible en las cámaras y sistemas de visión que incorporen el Q-Eye.

- Se ha definido una arquitectura para un sistema de visión basado en el Q-Eye adecuada para su integración completa en un único chip, considerando la problemática que conlleva el diseño de sistemas embebidos. Esta arquitectura ha tenido en cuenta las características del sensor-procesador relativas a su configuración/programación y potencia de cálculo para realizar un co-diseño "hardware-software" óptimo. En una primera aproximación al problema se ha desarrollado el sistema de visión Eye-RIS_v1 para verificar la arquitectura del sistema de visión propuesta. Este sistema de visión está compuesto en líneas generales por el sensor inteligente Q-Eye, un controlador que actúa de interface ente el sensor-procesador y el micro-procesador del sistema, circuitería digital de procesamiento específico, memorias de programa y datos e interfaces de comunicación. En este caso, tanto el micro-procesador como la circuitería de procesamiento específica y el controlador van implementados en una FPGA.
- Una vez verificada la arquitectura propuesta para el sistema de visión, se ha llevado a cabo el diseño e implementación física de una versión de dicho sistema en un único chip, utilizándose la misma tecnología 180nm estándar CMOS de 1.8V. Este sistema de visión "on-chip" se denomina Eye-RIS_Vsoc. Durante la fase de desarrollo del mismo se han identificado y estudiado todos los aspectos críticos desde un punto de vista del diseño e implementación, confirmándose que son similares a los presentes en el desarrollo de cualquier sistema de visión embebido:
  - Área: esta limitación implica una restricción en el número de recursos disponibles como son las memorias de programas y datos, condiciona la resolución espacial y la precisión del procesamiento, limita el número de "pads" para los interfaces de comunicación y alimentaciones, acota el número y complejidad de los procesadores, etc.
  - Restricciones temporales: El sistema de visión tiene un conjunto de procesadores y controladores que deben realizar un determinado algoritmo en un tiempo determinado por la velocidad de operación requerida en la aplicación. Por ello, los

procesadores y controladores tienen que realizar cada uno de ellos una determinada secuencia de tareas en un tiempo dado, y sincronizarse entre ellos buscando el mayor grado de paralelismo posible para optimizar la velocidad de operación.

- Consumo de potencia: Un presupuesto restrictivo del consumo de potencia implica una limitación en la precisión de los algoritmos y una limitación en la potencia de cálculo con su correspondiente penalización en la velocidad de operación. En le caso de los circuitos integrados, el consumo de potencia implica un aumento de temperatura del silicio. Para los sensores de imágenes este aumento de temperatura degrada significativamente la calidad de la imagen debido al efecto de la corriente oscura.
- Integridad de las señales: En el caso de un sistema integrado en un único chip, circuitos analógicos y circuitos digitales se encuentran integrados sobre el mismo silicio. La actividad de estos últimos supone una fuente de ruido para las alimentaciones, lo cual puede suponer una limitación en la precisión alcanzada por los circuitos analógicos.

En el caso de una integración en silicio estas restricciones y condicionamientos son aún más restrictivos. Durante la fase de integración se han dado soluciones a estas limitaciones haciendo técnicamente viable el sistema de visón "on-chip" propuesto. Dicha implementación ha demostrado el nivel de integración y compacidad alcanzado con la aproximación CVIS propuesta basada en el sensor inteligente Q-Eye con procesamiento analógico masivamente paralelo en el plano sensor.

CHAPTER 1 - INTRODUCTION AND MOTIVATIONS

1.1. INTRODUCTION

This Thesis deals with the design of chips conceived for sensing and processing images using CMOS technologies. **IMAGE SENSORS** built in CMOS technology are usually referred to as CIS (from CMOS Image Sensors). CISs deliver images when irradiated by light at frequencies covering roughly the visible and the near infrared bands of the electromagnetic spectrum: [400nm < λ < 1000nm]. Image capture by CIS involves, among other processes, photon absorption and photon-to-charge transformation; these processes happen both at photo-sensor devices embedded into a *silicon* semiconductor substrate. Therefore this substrate becomes an *active*, photosensitive "plane" where light gets focused by optical lenses. This is the reason why this active semiconductor plane is sometimes called *focal plane*.

The focal-plane includes also non-sensitive circuits employed for different purposes. This is illustrated in Figure 1 which depicts a conceptual block diagram of a modern CIS chip conceived to deliver *digitally-encoded* images [ANAFOCUS]. When implemented in a *planar* technology all these blocks are embedded in the focal plane although only the so-called *pixel* array (part shadowed in yellow in Figure 1) must be photo-sensitive. The pixel array consists of a regular arrangement of photo-sensitive devices each capturing a spatial sample of an incoming image. The rest of chip functions are customarily located in the chip periphery and must be masked to reduce the impact of parasitic photon-induced electron-hole generation outside the photosensitive area.

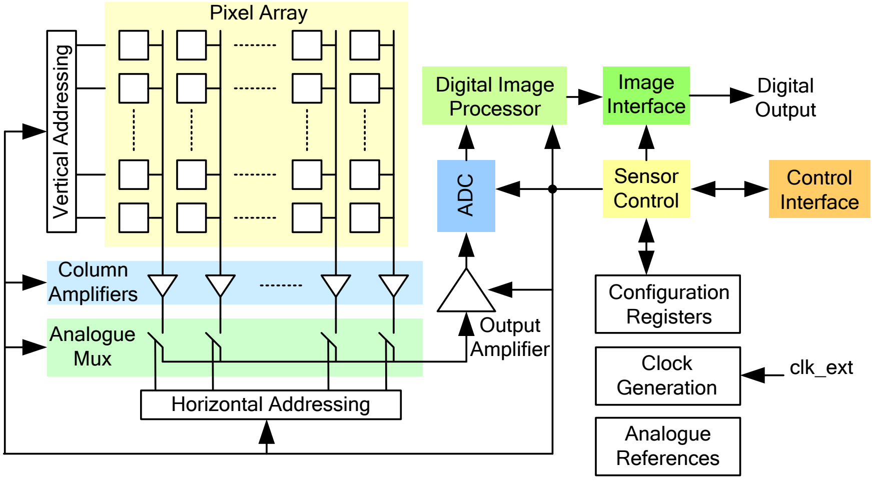
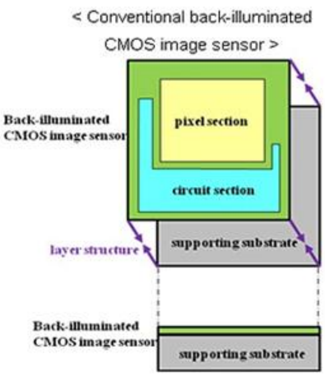


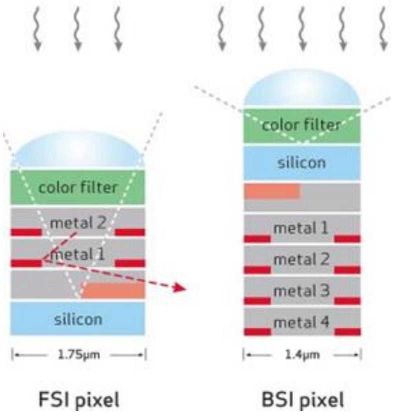
Figure 1. **DIGITAL CMOS IMAGE SENSOR.**

Regarding pixels themselves, they also include sensitive and non-sensitive devices and circuits. The figure at the right inset shows a conceptual footprint of a modern CIS pixel where the active, photo-sensitive area, (in yellow) amounts typically to 45% of the total pixel area. This percentage is called *fill factor*, while the term *pixel pitch* is employed to denote the length of the lateral pixel side. Some prevalent trends of CIS industry are decreasing pixel pitch and increasing fill factors. The latter one involves decreasing the amount of non-photosensitive circuits embedded at pixels. This trend is mostly observed in CISs devised for consumer applications [Font11] [ISW13] [IS14] [EImg14]. However, CISs employed for machine vision may contain significant portions of non-sensitive circuits within pixels. Chips presented in this Thesis belong to this latter category. Actually, their pixels embed circuits intended for processing images right at the focal plane and concurrently with image capture. This is the reason why these circuits are sometimes referred to


< Conventional back-illuminated CMOS image sensor >
Back-illuminated CMOS image sensor
pixel section
circuit section
layer structure
supporting substrate
Back-illuminated CMOS image sensor
supporting substrate

as **FOCAL-PLANE SENSOR-PROCESSORS** [Rosk01] [Zara11]. In the more general case, non-sensitive circuit structures employed for sensor signal conditioning and processing can be incorporated at different levels [Ohta08]:

- per-pixel,
- per-column,
- per-chip.



Chips in this dissertation combine all these three strategies [Rodr10a, b].

CIS chips can be illuminated either at the top (Front Side Illumination – FSI) or at the bottom (Back Side Illumination – BSI). Differences between these two cases are illustrated for a single pixel at the right figure inset. In the case of BSI, optical lenses are placed at the bottom surface of the semiconductor wafer and the light reaches directly the silicon (where photosensors are placed) without interfering with IC layers employed for interconnection and isolation. In the case of FSI, optical lenses are placed at the top surface and light must propagate throughout the top chip layers and structures before reaching the photosensitive silicon. Discussion of pros and counters of these alternatives is not a topic of this dissertation and can be found elsewhere [Prov11]. Chips reported in in this Thesis are conceived to operate as FSIs.

Image sensors are placed at the *front-end* of **CAMERA SYSTEMS**. Figure 2 shows the block diagram of a camera system including optics, a front-end image sensor chip and a number of blocks for controlling, data storage and data communications. Image sensors cannot be used as isolated components but must be embedded into camera systems, called simply cameras for practical usage. Besides image sensors and all the functional structures required for control, communication and system integration, camera may also include circuits and systems to analyze image flows and extract information from them. Despite how complex a camera system may be, it is clear that its ultimate image handling abilities will critically depend on the performance of the front-end image sensor because imaging quality is basically conferred by the image sensor. Although this Thesis is mostly focused on the design of image sensory-processing chips, consideration pertaining to camera design will also be made throughout the chapters. By the way of example, the figure at the left inset shows the appearance of one version of the Eye-RIS camera system described in Chapter 5 of this dissertation.

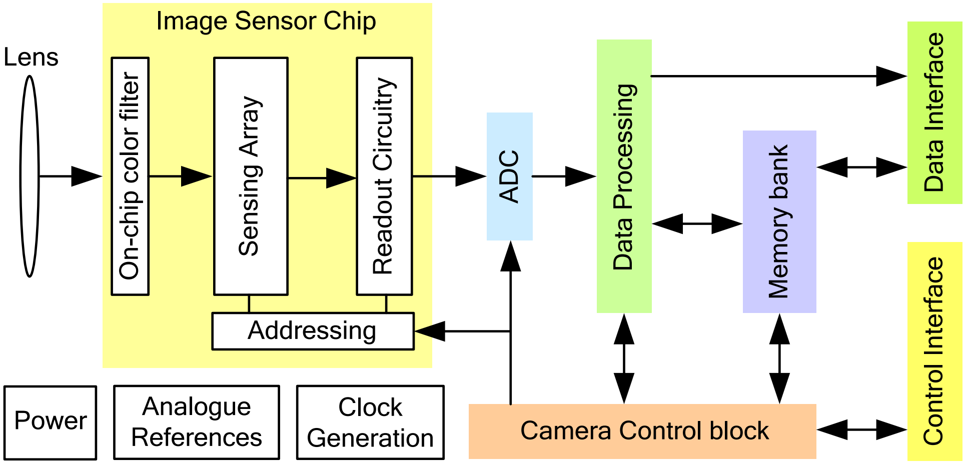


Figure 2. CAMERA SYSTEM BLOCK DIAGRAM.

We may wonder why are imagers, cameras and related technologies worth addressing? Daily life experience supports an intuitive answer to this question. Indeed, daily life shows that the ability to capture, analyze and interpret optical images is likely to be crucial for systems to interact with man and the environment. Actually, *solid-state* image sensors are employed at an ever increasing number of applications. Thus, besides the extensive usage of imaging devices in cell phones and personal electronic appliances, other application sectors with large growing rate potentials are rapidly evolving such as medical, military, automotive, surveillance, robotics, machine vision, and the like.

Among other reasons, the currently witnessed proliferation of solid-state image sensors is driven by the potential of modern CIS technologies to design *systems* with reduced SWaP (Size, Weight and Power), low *cost*, large *speed* and large *functional* capabilities and *flexibility.* However, until approximately the mid 90´s, the expansion of CMOS imaging systems was hampered by the lower quality of their compatible photo-sensing devices as compared to CCD (Charge Coupled Devices) counterparts [Naka06] [Ohta08]; i.e. images captured by earliest CMOS photo-sensors were not good enough for many industrial applications. In such a scenario CCD was the choice for imaging systems and the potential advantages of CMOS were sort of academic nuances. This drawback was attenuated with the advent of improved CMOS photo-sensors, specifically based on *pinned photodiodes*, and the subsequent enhancement of the quality of captured images above the feasibility thresholds required for practical applications. Once these thresholds were surpassed, and despite that CCDs may still provide larger quality at image acquisition, the CMOS advantages regarding SWaP, production cost, speed and embedded functionality became unbeatable weapons at the CMOS-CCD duel. Actually, CMOS-systems cope todays more than 90% of the market of *area* imager sensors, where the term area sensor is used here in contraposition to *linear* sensors; the former capture optical signals using a bi-dimensional detector array, while the latter employ a one-dimensional detector array as required for instance for scanning applications. Also, the total volume market for CIS is expected to grow significantly during the next few years, with an overall market forecast of around 3 billion units for 2015 and 17 billion for 2018 [Yole14a-b]. Figure 3 illustrates the share of CIS image acquisition systems shipment by applications as reported by Yole Development. This Thesis focuses on CMOS-based systems and does not attempt to analyze and quantify pros and counters of CMOS versus CCDs. Readers interested in solid-state imaging technologies with CCDs may refer among others to [Theu95].

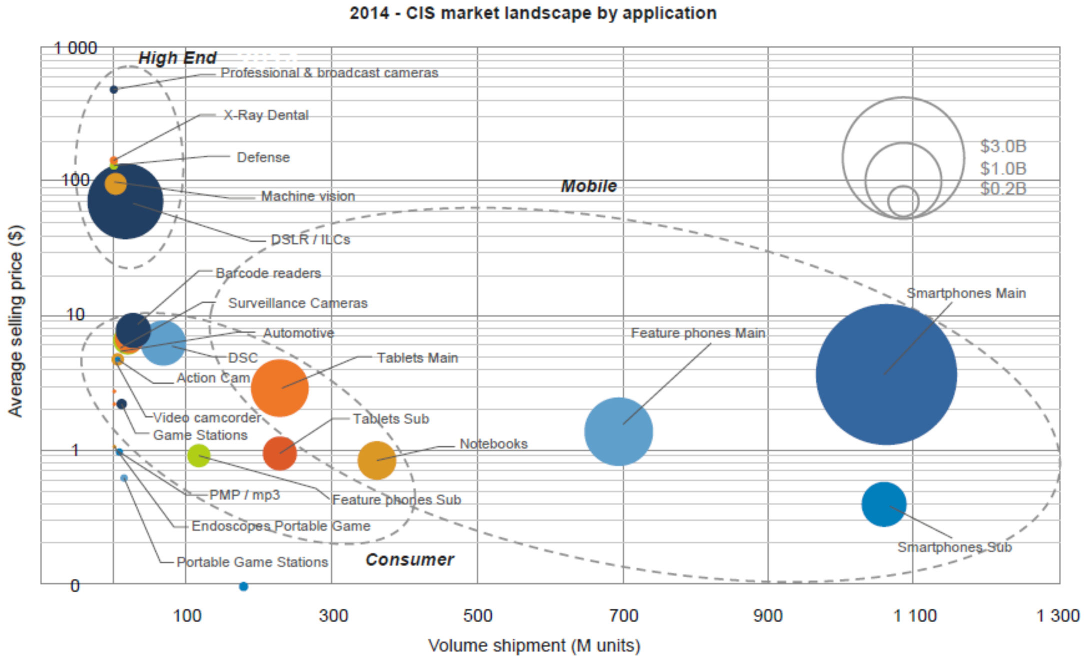


Figure 3. CIS SHIPMENT FORECAST BY APPLICATIONS (YOLE DEVELOPMENT).

Major advantages of CMOS can be summarized in the following items:

- Cost
- Size.
- Power Consumption

- Speed
- Functional Flexibility

Advances in CIS are basically driven by consumer applications. These applications also condition the technical challenges pursued by industry and academia regarding CISs. On the one hand, as already mentioned, consumer markets call for ever smaller pixel pitch, and ever larger spatial image resolution (number of pixels). Besides the quest of reduced pitch, other prominent CIS challenges include:

- improving the image quality through enhanced readout, signal conditioning and image enhancement circuitry;
- improving the image downloading speed through enhanced communication circuitry and;
- reducing the area, power and cost through on-chip circuit embedding [ISW13] [EImg14] [IS14].

Recent advances include multi-million-pixel sensors with a pixel-pitch on the 1μm range, data rates above 10Gpx/s [Taka13], advanced, reconfigurable A/D conversion and readout architectures [Leñe14] [Kawa08], image correction [Chen12], thermal and energy management, etc.

All previous challenges are focused on the acquisition of light intensity maps defining 2-D images. During the last few years we have also witnessed ever increasing activities towards adding the estimation of depth, i.e. 3-D information, to the 2-D scenes. One of the main drivers of these activities is the development of human-machine interfaces for the entertainment industry [Payb14], but the potentials of these technologies to surveillance, automotive, industrial inspection and medicine show large potentials for huge development. Besides techniques based on stereoscopy, triangulation and the like, significant efforts are being made towards modifying the CMOS pixels so that they are able to capture time information and estimate depth through Time-of-Flight (ToF) techniques. Among different approaches, Single Photon Avalanche Diodes (SPADs) are receiving significant interest [Seit11]. Despite the specific pixel technology employed (SPADs, double transfer gate, etc.), ToF measurements require complex active illumination schemes [Nicl11] — still far from the single-chip solution for 3-D imaging. Also, 3-D sensors are lacking behind mainstream CIS ones regarding the incorporation of on-chip processing circuitry.

Perhaps the most important asset of using CMOS is that it enables the design of complete imaging *systems on-chip* instead of just image sensors. An earlier realization of this possibility was the concept of *camera-system on-chip* introduced in [Foss97]. Referring to the block diagram of Figure 2, this concept involves embedding as many as possible of the functional blocks required to implement a camera within the same, common, silicon substrate. Therefore, CMOS imaging systems-on-chip belong to the general category of *smart sensors*; i.e. sensors with embedded *intelligence* [Meij01].

The concept of **INTELLIGENT SENSORS** is an undefined, ambiguous one. Generally speaking, the smartness feature embraces whatever kind of circuit structure employed to handle the signals acquired by sensors, despite such handling being used either for readout, for error correction, for calibration or for extracting information from the images, among others. Different levels of intelligence can be considered in practice. The "lowest" involves basically functions related to image readout, error correction and communications, and is the most exploited by industry up to now. Todays, smart CISs with embedded microprocessors that deliver fully-corrected digital images to the outside at thousand´s Frames-per-Second (FpS) and with noise level below one-equivalent-electron are available [ANAFOCUS]. Higher intelligence levels, as required to extracting and interpreting the information contained into images and prompting sub-sequent *reaction* commands, have been explored for years at academia [Eklu96] [Rodr04] [Abbo08] [Fern11] [Lopi11], and industrial applications are recently ramping up [ANAFOCUS]. One important reason for that is the lack of standardization regarding both architectural solutions as well as output formats [Khro13]. The latter can be addressed by resorting to the usage of cellular architectures and, particularly, to the use of the Cellular Nonlinear Network (CNN) architectural paradigm as it fits to the specifics of the problem addressed, namely: large, regular array of data with properties emerging out the data collections and which can be disclosed through local interactions and signal propagation in space and in time [Rosk01] [Chua02].

Depending upon the level of intelligence embedded, two types of CMOS sensing-processing chips can be distinguish, namely:

- SCIS which stands for Smart CMOS Image Sensors, and
- CVIS which stands for CMOS VIsion Sensors.

Figure 4illustrates differences between SCIS and CVIS at conceptual level. While the inputs of both chip types are images captured by photo-sensors placed in the *focal-plane* their primary outputs are of different nature.

In the case of SCIS, basic outcomes of the processing chain are just images; either grey-scale images or color images, in analog format or in digital format, *raw* or *corrected*, etc. On the contrary, the outcomes of CVIS may not be images but either image *features* [4] or even decisions based on the spatial-temporal analysis of the information contained into the scene. Both SCIS as well CVIS chips are conceived to be employed at the front-end of complex hardware-software camera systems. Considerations pertaining to the design of these camera systems, including topics like *smart cameras* [Belb09], are very wide and beyond the scope of this overview.

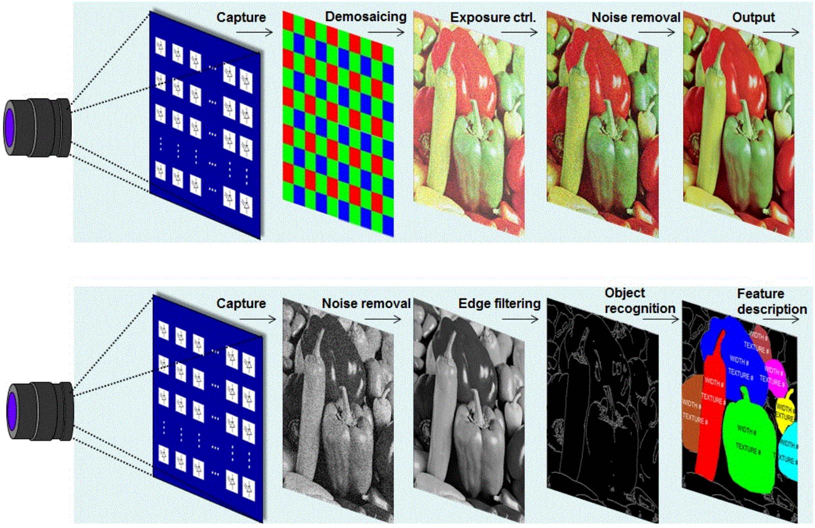


Figure 4. **ILLUSTRATING DIFFERENCES BETWEEN SCIS (TOP) AND CVIS (BOTTOM).**

Despite their differences regarding function, SCISs and CVISs are structurally similar; specifically, both combine two different types of subsystems on a common silicon substrate:

- Those employed for sensing: i.e. for capturing light power and transducing it into either voltages or currents. They usually consist of reverse-biased *pn*-junctions (called photo-diodes) together with some associated *mixed-signal* circuitry to control the sensor operation.
- Those employed for reading the outcome of the light transduction process, for signal conditioning, for Analog-to-Digital Conversion (ADC), for image correction and processing, for control and for communications. They consist of different analog, mixed-signal and digital circuits and subsystems.

These similarities can be exploited during the design phase as many concepts, architectures and circuits are shared by both kind of systems and can be re-used by designers. This fact is reflected throughout this dissertation.

1.2. OUTLINE OF THE THESIS

As stated in previous sections, vision markets (including surveillance, machine vision, intelligent transport systems, consumer electronics, vehicle guidance, vision-enabled wireless sensor networks, etc.) demand the following from sensor design:

- reduced production costs in system development;

[4] By image features can be interpreted as characteristics of the information contained into images; for instance the number of objects included into an image, the location of maximum spots, etc.

- increased integration level in physical implementations (compact vision systems);
- largest possible image quality combined with per-chip embedded functional performance;
- largest possible operation speed;
- best possible usage of the available power budget.

This monograph reports CVISs, and vision systems based on them, to meeting these challenges through:

- ***parallel processing*** architectures with,
- **per-pixel** distributed, **mixed-signal processors**.

Rationale underlying these architectural strategies are described in the paragraphs below.

PARALLEL PROCESSING is very well suited to handle the large amount of data present at early stages of the vision processing chain. As Figure 5 illustrates, the number of data available at initial steps of the vision processing chain may be huge for sensors with large spatial resolution and large number of bits per pixel data. Also, many of the data are redundant and hence useless to the purposes of extracting information and performing vision tasks. However, as information flows across the processing chain and *features* are extracted from the incoming images, the number of data decreases. Actually, very few data are typically involved in final decision-making; for instance, in surveillance applications decisions can be made depending on whether scenes include some suspicious object or not.

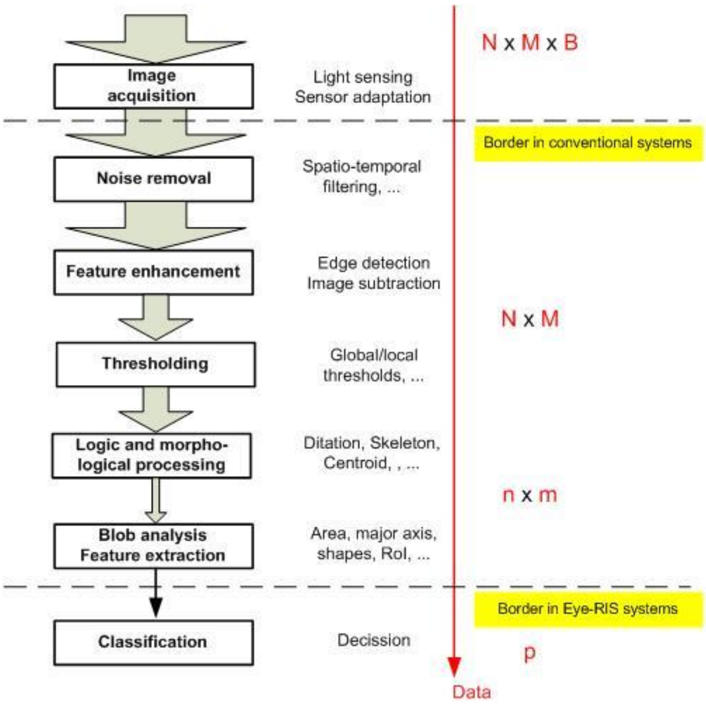


Image acquisition
Light sensing
Sensor adaptation
N x M x B
Border in conventional systems
Noise removal
Spatio-temporal filtering, ...
Feature enhancement
Edge detection
Image subtraction
N x M
Thresholding
Global/local thresholds, ...
Logic and morpho-logical processing
Ditation, Skeleton, Centroid, , ...
n x m
Blob analysis
Feature extraction
Area, major axis, shapes, RoI, ...
Border in Eye-RIS systems
Classification
Decission
p
Data

Figure 5. PROCESSING CHAIN OF VISION.

The amount of data decreases and the abstraction level increases as data evolve from the sensor interface (raw data); *N* represents the number of rows, *M* the number of columns and *B* the number of bits used per pixel data: $n < N$; $m < M$ and $p < (n, m)$. The a architecture proposed in this Thesis maps this layered data structure by using processing strategies fitted to each step in the chain.

In conventional vision system architectures the border between sensors and core digital processors is placed at a point where the amount of data is large. This border is marked by the dashed line drawn at the top in Figure 5. It means that conventional vision system architectures download *complete image frames* from sensors, thereby posing significant memory and computational demands on the processing side. For instance, for an image with high definition TV format, with a spatial resolution 1920x1080 and a word length of 10bits per pixel, the number of data associated to a frame amounts to 20.74Mbits, equivalently 2.6Mbytes, and the throughput associated to the transmission of this image in a system operating at stadard video rate of 30FpS is 622Mbits per second. If frame rates in the range of 1000FpS are required, as it happens for instance in some inspection applications, handling the corresponding huge amount of data may become an unsourmountable obstacle.

However, using the herein proposed CVISs shifts the border between front-end sensors [5] and core digital processors to a point where features instead of frames are transmitted. The number of data at this border, marked by the dashed line at the bottom in Figure 5, is much smaller than at the top one. Assume for illustration purposes that we target to track objects moving at 40m/sec within a scene. It requires capturing and analysing images at 2000F/s rate. At the outcome of the capture/analysis process the only pertinent data are the predicted positions of the objects. This is actually the only information required by the core digital processors. But to extract this information the following tasks must be completed:

- Image acquisition,
- low-pass filtering,
- activity detection,
- motion estimation,
- object tracking,
- loop control, and
- position prediction.

CVISs in this Thesis performs these tasks directly at the sensory front-end, thereby precluding data communication bottlenecks and reducing the computational payload of digital processors. A simple exmaple will help us gaining insignt in this payload: Consider the case of a simple *convolution* operation required for a linear spatial filtering. It involves 9 multiplications and 8 additions per pixel. Thus, for the the previous example of high definition TV images at 30FpS a processing speed equal to 1.06GOpS (Giga Operations per Second). And, this is just one of the many tasks that must be completed at early vision stages, Hence, besides communication and memory bottlenecks, conventional vision system architectures also create processing bottlenecks. Of course, if there is no power size constraints and system size is neither a problem, you may scale conventional architectures to face these bottlenects. But, under speed and SWaP constraints it may be advisable seeking for alternative architectures.

MIXED-SIGNAL CIRCUITS. Besides parallelism and distributed processing, the usage of mixed-signal processing circuits is another main attribute of the architectures proposed in this Thesis. Mixed-signal circuits are intrinsically faster and have larger power efficiency than digital counterparts under the assumption that accuracy requirements are moderate. Also, parallel analog processing reports advantages in terms of the number of operations attainable for given power consumption [Carm13]. All-in-all, it means that the vision system architectures proposed herein outperforms conventional solutions in terms of SWaP figures as well as in terms of operations/power values. These features, together with their cost, make them ideally suited to bridge the gap between point detectors and area detectors for machine vision and inspection applications – see Figure 6.

The core proposal of this Thesis is a CVIS called Q-Eye. The Q-Eye pixel has been endowed with a high level of programmability in order to implement the largest possible number of early processing functions required by vision applications [Toma06]. This programming capability of Q-Eye permits the development of smart cameras and embedded vision systems with a versatile functionality, not designed to perform a specific task. They can be programmed to carry out several image processing algorithms. Actually, Q-Eye pixels interchange information with their neighbors to realize a variety of operations such as:

- Linear convolutions with programmable masks.
- Time- and signal-controlled diffusions (by means of an embedded resistive grid).
- Image arithmetics.
- Signal-dependent data scheduling.
- Gray-scale to binary transformation.

[5] CVSIs do not only sense but also process images. Hence, the correct word to refer them should be front-end sensor-processors instead of just front-end sensors.

- Logic operation on binary images.
- Mathematical morphology on binary images.
- etc

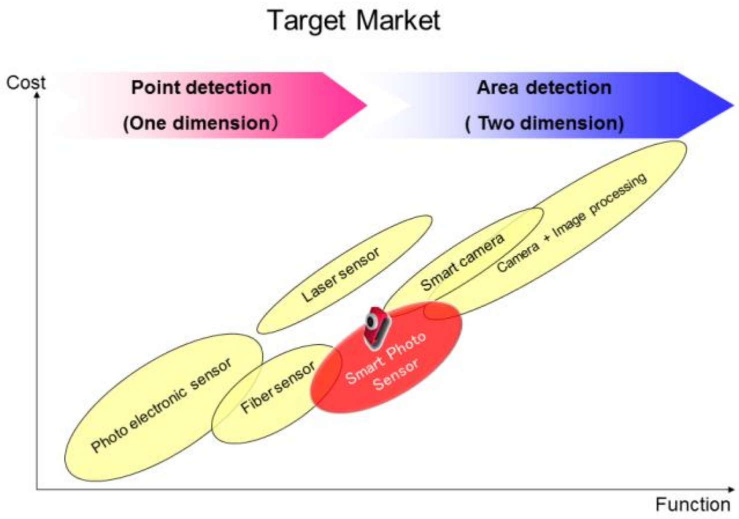


Figure 6. Positioning of the Toshiba-Teli SPS at the Cost-vs-Function target market plane.

This camera from Toshiba-Teli employs the CVISs and vision architectures devised in this Thesis [Toshiba]. Image taken from http://www.toshiba-teli.co.jp/en/products/industrial/sps/sps.htm

CHAPTER 3 provides a detailed architectural description of **Q-EYE PIXEL**. This includes schematics for all circuits embedded within the pixel including circuit details for all function embedded within the pixel, analysis of the main circuit errors and illustrations of their impact on ssytem operation.

The **Q-EYE CVIS CHIP** is described in **CHAPTER 4**, where communication ports, system programmability, readout architecture and miscellaneous block are explained. Also, the issues at system level to be considered during the design phase are introduced.

This Thesis also reports **VISION SYSTEMS** built using the Q-Eye CVIS at the front-end. Differences between these systems and conventional vision systems have been outlined above with reference to the location of the border between front-end sensor-processors and cores processors within the processing chain of vision – see Figure 5. These differences are further highlighted via Figure 7 and Figure 8.

On the one hand, Figure 7 corresponds to a conventional vision system, where the front-end just senses input images and full frames, represented by the symbol **F**, are downloaded from this front-end for processing. As already mentioned, the huge amount of data associated to full frames create communications, memory and computational bottlenecks.

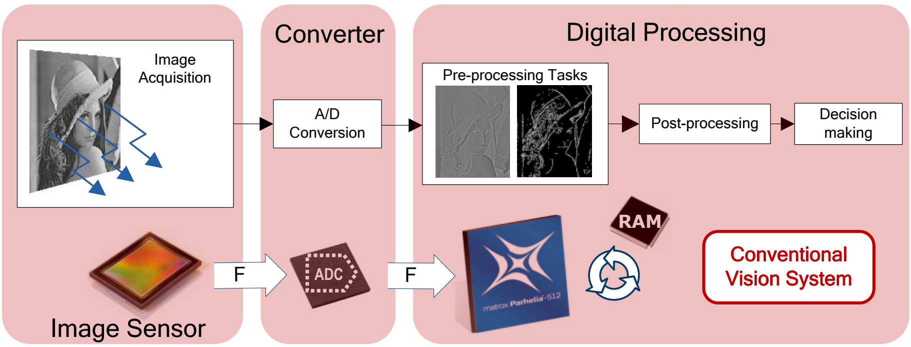


Figure 7. DATA FLOW IN CONVENTIONAL VISION SYSTEM.

On the other hand, Figure 8 corresponds to the vision systems considered herein. Notice that the front-end does not only sense but also *pre-process* the information. Hence, it sends much smaller amount of data, represented by the symbol **f** << **F**, for ulterior processing. Indeed, in the architecture of Figure 8, processing is made progressivley by distributing processing tasks between the front-end and the core porcessor sections. The front-end already realizes a ***pre-processing*** of the data by executing parallel-processing algorithms directly in the pixel array. These algorithms involve many data and are very demanding in terms of power and speed thereby benefiting from the usage of mixed-signal techniques. Regarding ***post-processing***, this stage handles less data but require complex logic with numerous decision points, alternate paths of execution, etc. The post-processing stage consists of complex image processing algorithms which require programmability and reconfigurability. The computational payload is not so high because the amount of data to process is less, therby relaxing the specifications of processors included in the vision system. All-in-all, the diiferent sections of the proposed architecture are fitted to the nature of the data they handle, and the whole system is much more efficient than the conventional one.

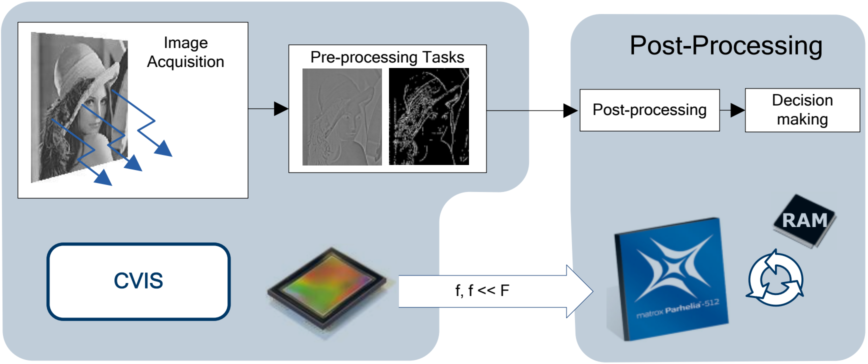


Figure 8. **DATA FLOW IN VISION SYSTEM BASED ON THE Q-EYE CVIS.**

CHAPTER 5 is devoted to present a couple of vision system built according to the concept of Figure 8 and using the Q-Eye CVIS as sensor-processor front-end. These visions systems are respectively called:

- Eye-RIS_v1 and
- Eye-RIS_v2.

Differences between these systems rely in the degree of embedding. Namely the second one embeds the Q-Eye and a digital RISC microprocessor in a single silicon substrate, while the first employs an off-chip RIS microprocessor implemented of FPGA. This chapter also shows the diiferent chips and systems in operation.

The Thesis is completed with a Chapter, namely **CHAPTER 2**, devoted to introduce basic concepts related to image acquisition and processing.

CHAPTER 2 – BASIC CONCEPTS, DEFINITIONS AND TERMS

CSISs, SCISs and CVISs are complex systems whose design embraces quite different disciplines, including: solid-state photo-sensors, analog signal conditioning, readout and analog-to-digital conversion, digital image correction, digital microcontrollers, memories, etc. Therefore, the design of this type of advanced systems-on-chip requires the concourse of multidisciplinary teams. Also, because correct operation calls for close interaction among the different parts, global knowledge is mandatory for successful design and correct operation. This is particularly pertinent for CIS embedding smartness (SCIS and CVSI) and for those employed at high-end applications and/or requiring high-speed or low-noise image capture, on-chip image correction, scene interpretation, high dynamic range capture, etc. All these features demand architectural and circuital innovations and pose significant challenges to designers. These challenges are even more demanding when, as it happens in this Thesis, sensing is combined with image processing. Actually, pixels, chips and systems proposed throughout the monograph belong to general category of advanced sensory-processing imagers. Hence, besides the multi-disciplinary challenges involved in pure imagers, others including image processing, mixed-signal processing circuits, digital processing circuits, hardware-software co-design, etc. must be considered as well.

Core chapters of this Thesis describe respectively the architecture and design of a sensory-processing CVIS, called Q-Eye, and the architecture and implementation of two vision systems, called Eye-RIS, based on this CVIS. Thus Chapters 3 and 4 cover respectively the Q-Eye pixel and the Q-Eye chip and Chapter 5 covers the design and applications of two versions of the Eye-RIS systems. Previous to the presentation of the architectures, the circuits and the methods backing the design and usage of Q-Eye and Eye-RIS chips, this chapter is intended to present basic concepts related to imagers and vision systems.

2.1. FROM LIGHT RADIATION TO PICTURE ELEMENTS

2.1.1. IMAGERS: CONCEPT AND BASIC TERMS

IMAGE SENSORS are aimed to convert optical scenes into electrical images. The concept is illustrated in Figure 1 where the light coming from a *scene element* gets focused by an *imaging system* into a sensing area, called *image plane*, consisting of an array of *photo-detectors*.

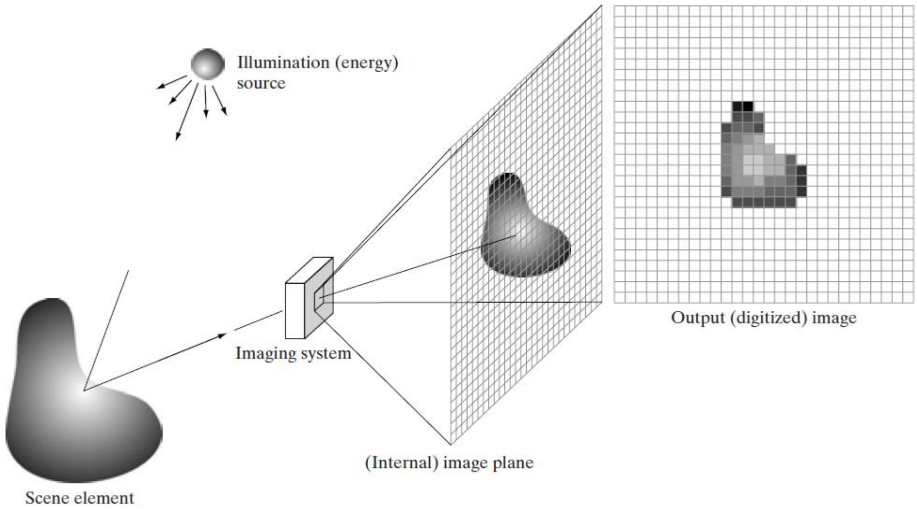


Figure 1. CONCEPT OF IMAGE SENSING.

Figure from [R. C. Gonzalez and R. E. Woods, Digital Image Processing, Prentice Hall, 1992]

Each site in the array (usually called ***pixel***) embeds a photo-sensing device to obtain a spatial sample of the image when excited by the photons that impinge on the area corresponding to the detector. Samples are usually called *picture element*s. While sensitive element units may also be called picture elements, we prefer using the term pixel to refer to the physical entities and the term picture element to refer to the values provided by these entities.

At the outcome of the photo-sensing and transduction processes the input optical scene gets sampled in space and encoded into a two-dimensional *electrical image* described by a two-dimensional mathematical function,

$$I(i,j,t) = x_{ij}(t) \quad for \begin{cases} 1 \le i \le N \\ 1 \le j \le M \end{cases} \qquad \textbf{Eq. 1}$$

where *N* indicates the number of *rows* in the image (the image *height*), *M* indicates the number of *columns* in the image (the image *width*), and *t* is the time variable. Obviously, practical image sensors must include sub-systems to allow electrical images be read by external systems – READ-OUT CIRCUITRY. Two basic subsystems involved on that are:

- ***Pixel addressing*** circuits,
- ***Read-out*** circuits.

Figure 2 shows a conceptual architecture of a CMOS image sensor including these two entities where electrical images are read as analog signals. The pixel array defines the sensing area and is a main agent for image quality. Vertical and horizontal access circuits are used to reaching pixels and reading pixel signal values. Usually, *shift registers* are employed to control addressing. A *decoder* can be used to access pixels randomly. The readout circuit in Figure 2 consists of a one-dimensional array of switches and a Sample and Hold (S&H) circuit [Ohta88] [Seit00]. Noise cancellation circuits, such as Correlated Double Sampling (CDS) [Enz96], are typically employed there.

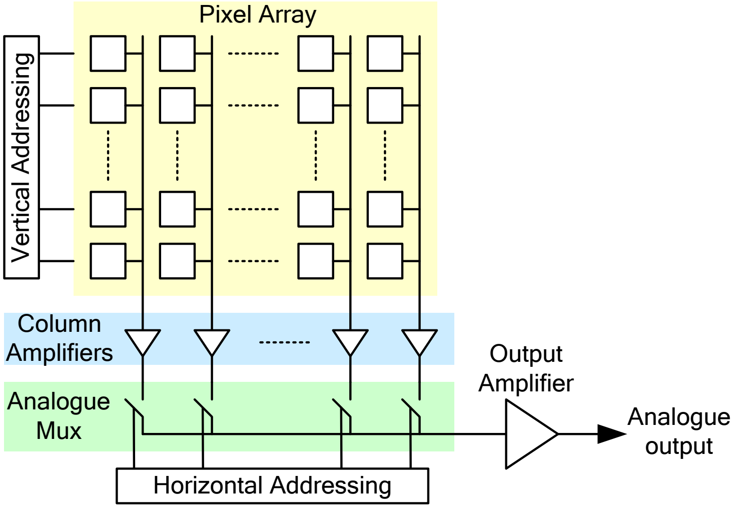


Figure 2. CONCEPTUAL BLOCK DIAGRAM OF AN ANALOGUE READ-OUT CMOS IMAGE SENSOR.

Most modern CIS employ Analogue-to-Digital Converters (ADC) in the readout circuitry and employ sophisticated error correction circuits in the read-out channels, usually in per-column basis [Rodr15]. Further details about these *digital* CIS will be presented in section 2.5.1.

MONOCHROMATIC AND COLOUR IMAGES. In the case of ***monochromatic*** (gray-scale) images each sample consists of a value which represents the light intensity captured by photo-detector placed in the corresponding array site (see Figure 3-left). These values are carried either by analog voltages or by Digital Numbers (DN) obtained through analog-to-digital conversion of corresponding voltage levels. In some cases, currents may be employed to carry picture element values [Berm04]. In the case of ***color*** images, the mathematical representation requires several bi-dimensional functions instead of just one. Typically, three functions are employed each encoding one out of three main color channels: red, blue and green – illustrated in Figure 3-right.

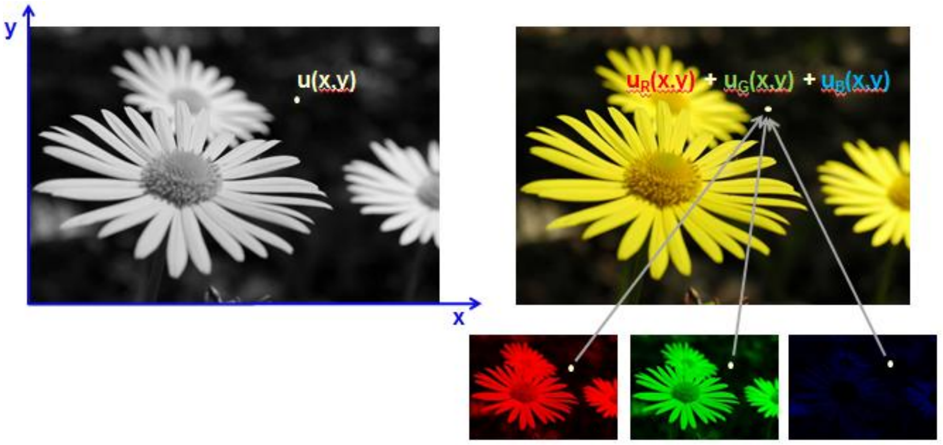


Figure 3. (LEFT) MONOCHROMATIC IMAGE: 2D FUNCTION U(X, Y) WHOSE AMPLITUDE AT COORDINATES (X, Y) IS A POSITIVE SCALAR QUANTITY AND WHOSE MEANING IS DETERMINED BY THE IMAGE SOURCE; (RIGHT) POLYCHROMATIC (COLOR) IMAGE: COLLECTION OF FUNCTIONS U_K(X, Y) DEFINED ON A 2D SPACE, AND CORRESPONDING TO DIFFERENT SPECTRAL COMPONENTS

Electrical images do contain ***information*** which is encoded in the variations of the picture element values across ***space***, on the one hand, and over ***time***, on the other hand [Gonz92] [Seit00] – see Figure 4. These latter variations are typically *synchronous* with a clock and hence happen at regular intervals of this clock period. Moreover, picture element values are commonly obtained through processes where electrical charges get accumulated into capacitors over the clock period interval. Although some applications employ asynchronous, continuous-time readout, *integration* and *discrete-time* operation are typically employed because the instantaneous values of picture elements are extremely weak unless the illumination levels are very high.

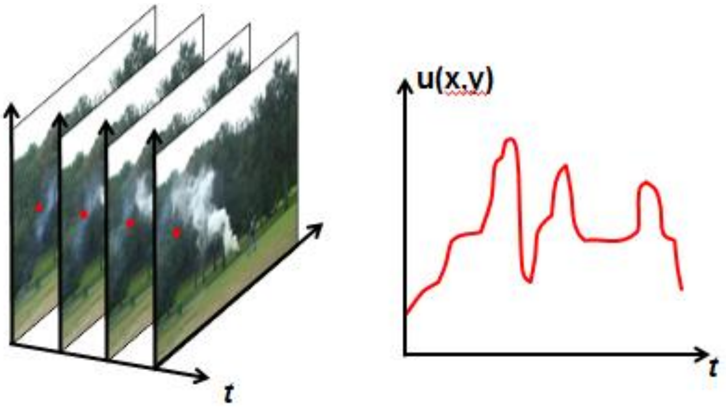
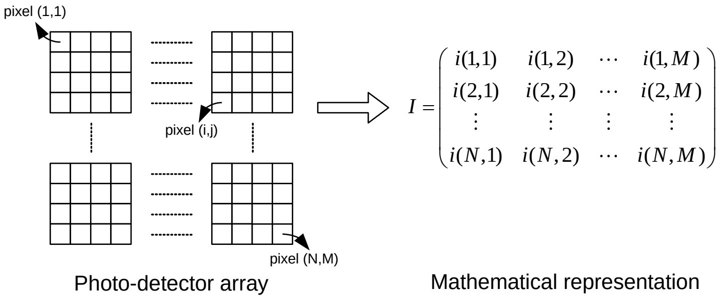


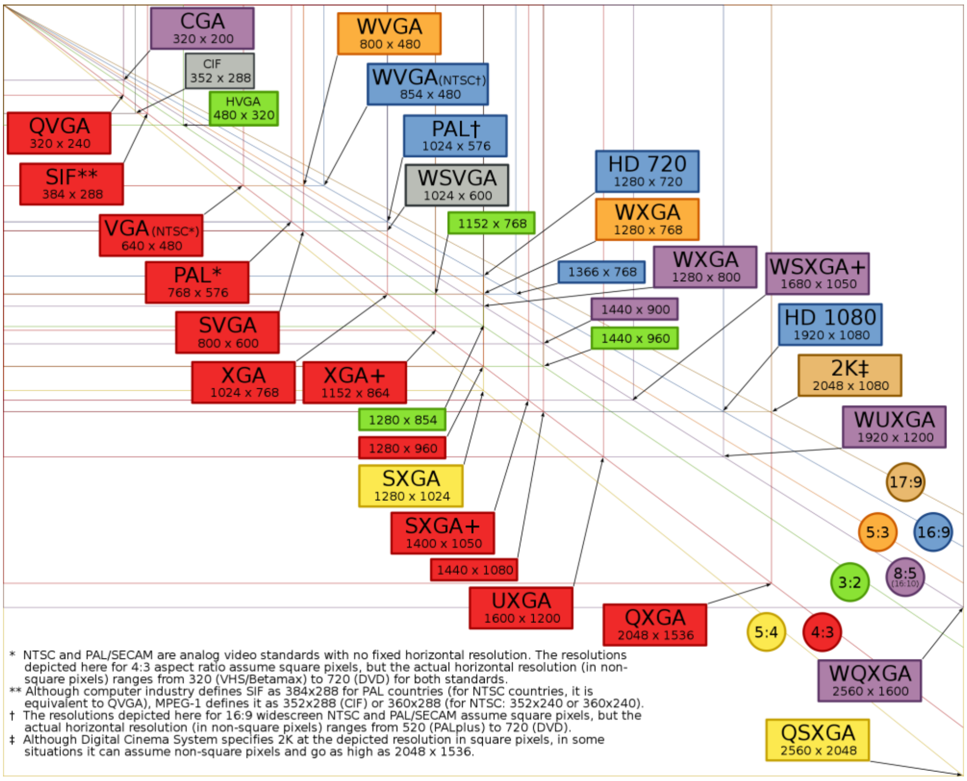
Figure 4. ILLUSTRATING VARIATIONS OF PICTURE ELEMENTS OVER TIME

Let us compile a few useful **DEFINITIONS** before closing this section:

- **IMAGE SIZE**, also referred to as image resolution or *spatial resolution*. This is expressed as the number of pixels obtained by multiplying the width (number of columns: *M*) by the height (numbers of rows: *N*). Different formats used in practical applications involve different number of pixels (*M* x *N* value) and *form factors* (*M*-to-*N* ratio). For instance: CIF is 352 x 288, VGA is 640 x 480, QSXGA is 2560 x 2048, etc. Figure 5 shows a chart of spatial resolution formats taken from Wikepedia Commons.



$$I = \begin{pmatrix} i(1,1) & i(1,2) & \cdots & i(1,M) \\ i(2,1) & i(2,2) & \cdots & i(2,M) \\ \vdots & \vdots & \vdots & \vdots \\ i(N,1) & i(N,2) & \cdots & i(N,M) \end{pmatrix}$$



* NTSC and PAL/SECAM are analog video standards with no fixed horizontal resolution. The resolutions depicted here for 4:3 aspect ratio assume square pixels, but the actual horizontal resolution (in non-square pixels) ranges from 320 (VHS/Betamax) to 720 (DVD) for both standards.
** Although computer industry defines SIF as 384x288 for PAL countries (for NTSC countries, it is equivalent to QVGA), MPEG-1 defines it as 352x288 (CIF) or 360x288 (for NTSC: 352x240 or 360x240).
† The resolutions depicted here for 16:9 widescreen NTSC and PAL/SECAM assume square pixels, but the actual horizontal resolution (in non-square pixels) ranges from 520 (PALplus) to 720 (DVD).
‡ Although Digital Cinema System specifies 2K at the depicted resolution in square pixels, in some situations it can assume non-square pixels and go as high as 2048 x 1536.

Figure 5. **CHART OF SPATIAL RESOLUTION FORMATS.**

Chart from Wikipedia Commons

- **FRAME** is somewhat equivalent to electrical image as it denotes the set of picture element matrices obtained after each period of the clock controlling the operation; i.e., after each *snapshot*.
- **FRAME RATE** is the inverse of the clock period which determines the temporal updating of the picture elements; i.e. it is the frequency at which electrical images are updated in time. Most imaging systems operate on a frame-by-frame basis which involves obtaining and downloading complete frames. However, smart imagers may operate in *asynchronous* way by detecting *events*, thus precluding the downloading of complete frames [Delb06].
- **PIXEL PITCH** denotes the physical dimensions of the pixels. Pixels employed used for last generation consumer electronic imagers are in the 1μm range [Font11] [ISW13], while pixels of industrial CVISs devised for factory automation may have tens-of- μm pitch; 35μm in the case of the Eye-RIS system [Rodr10a] [Rodr10b]. This larger pitch is required to perform parallel processing on pixel and reports significant speed advantages in the analysis of image flows by reducing the amount of data of interest – similar to what animal retinas do [Rosk01] [Zara11].
- **FILL FACTOR**, is the percentage of the pixel area which is occupied by sensitive devices. The fill factor value depends on the amount of no-sensitive circuitry embedded on-pixel as well as on the technology employed. For instance, sensing chips can either be irradiated at the front (*frontside* illumination) or at the back (*backside* illumination) [Font11]; 3-D chips consisting of vertically integrated physical layers can be used to separate the sensing devices from the processing ones [Garr08]; sensing materials can be deposed on top of CMOS wafers [ISORG]; etc.
- **REGIONS OF INTEREST**. It denotes a sub-set of the total picture elements set.
- **IMAGE SCALES** are downscaled, subsampled versions of the electrical images that preserve the features of the original ones.

2.1.2. Basic Radiometric Concepts

As already mentionned, electrical images are formed in a process that involves the irradiation of photo-sensor devices with an ElectroMagnetic Radiation (EMR) and the subsequent transformation of irradiated energy patterns into picture element values. This transformation happens point-wise across a spatial distribution of sensitive elements placed at the focal plane of the incoming EMR flow. The figure at the inset illustrates the concept for a monochromatic image.

2.1.2.1. Patterned Radiance Maps and Scene Formation

SCISs, CVISs are excited by optical *scenes* and meant to sense them. Optical scenes consist of modulated EMR power patterns. The formation of optical scenes and electrical images involves different disciplines, such as *radiometry*, *materials science* – particularly light-matter interactions, *optics* and, of course *electronics*.

Object features are encoded within optical scenes in the form of spatial-temporal variations of the EMR *radiant flux* Φ; where the radiant flux, measured in Watts, represents the rate of change (power) of the EMR energy. When the photo-sensor array gets the radiation ,characterized by a patterned flux, electrical images are generated whose picture element values capture the spatial-temporal variations of the incoming scene. This generation involves the *absorption* of the radiant flux by the semiconductor and the subsequent transformation of the flux photons into electrical charges.

Figure 6, adapted from [Fiet12], illustrates the generation of a scene power flux under *outdoor* lighting conditions – similar concepts apply for *indoor* condition. The illustration shows a chain of processes:

i) an EMR source (the sun in this illustration) *radiates* energy;

ii) part of this energy is lost and other part *irradiates* an object of interest;

iii) this object produces a reflected radiant flux;

iv) part of this flux reaches the camera lens and finally the sensors placed at the camera focal plane.

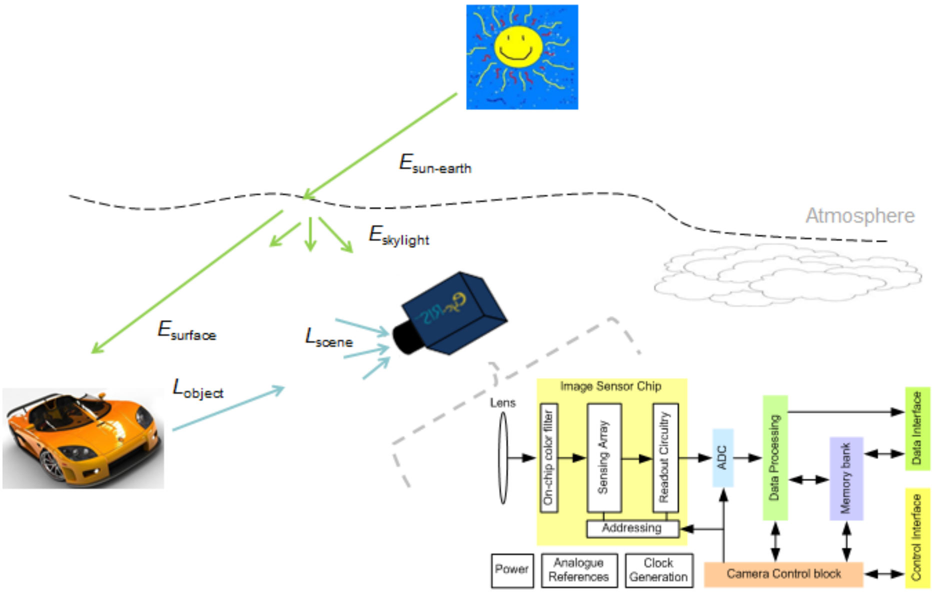


Figure 6. **Illustrating the generation of a scene power flux under outdoor lighting conditions.**

This figure has been adapted from R.D. Fiete, *Modeling the Imaging Chain of Digital Cameras*. SPIE Press 2012.

Let us start with the EMR source. Its *irradiation rate* is characterized by the so-called radiant *exitance* (or *emittance*) which, in the case of the sun, can be approximated by using a temperature *T*=5,800ºK in the expression derived by Max Planck´s for the blackbody radiation:

$$M_{BB} = (\lambda, T) = \frac{2\pi hc^2}{\lambda^5} \cdot \frac{1}{e^{hc/\lambda kT} - 1} \quad \frac{W}{m^2 \cdot m}$$ **Eq. 2**

This magnitude is a spectral density which shows dependence with the wavelength λ. Note that all radiometric variables (emittance, irradiance, etc.) can be given either as spectral densities depending on λ or as integrated variables over the whole spectrum of interest. For instance, the integrated emittance would be

$$M = \int_0^\infty M(\lambda) \cdot d\lambda$$ **Eq. 3**

Figure 7 shows the emittance of the sun, the irradiance at top of the atmosphere and irradiance at sea level as a function of the wavelength, jointly with a plot that illustrates the sensitivity of silicon. It highlights good sensitivity within the *visible spectrum region*, roughly:

λ ∈ [400nm, 700nm], and

ν (frequency) ∈ [428THz, 750THz]

and a reasonable sensitivity within the near-infrared region. These sensitivity figures support most of the practical SCISs and CVISs applications, although new applications using structures sensitive in the THz range are emerging – not based on photon absorption [Pfei10].

Other EMR sources (such as incandescent lamps, fluorescent lamps,...) show different spectral emittance curves which in some cases do not fit the blackbody model. Either case, despite the EMR source being used, the radiant flux reaching the objects of interest differs from that emitted by the source, the difference calling for the definition of another spectral variable called *radiant incidance* (or *irradiance*), *E*(λ).

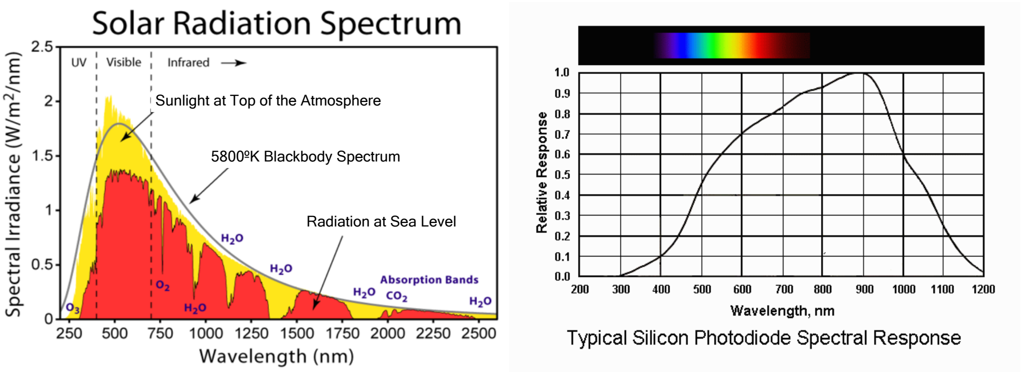


Figure 7. **EMITTANCE OF THE SUN AS A FUNCTION OF WAVELENGTH AND SENSITIVITY OF SILICON PHOTO-DETECTORS**

Figures from Wikipedia Commons

To calculate the irradiance from the emittance the following must be taken into account:

- The angular distribution of the emitted flux across the space, which is accounted for by using a spectral density per *solid angle* Ω, measured in *stereoradians* (sr). Bear in mind that the solid angle sets a relationship between the radius *d* of a sphere and the area *A* of a region of the sphere surface, namely:

  $$A = \Omega \cdot d^2$$

  Hence, the use of flux spectral densities per solid angle enables accounting for the distance between the source and the object of interest and for the area of the object of interest.

- The orientation of the surface corresponding to the object of interest, measured as the angle θ_{surf} from the normal to the surface to the incident ray.

- The combined effect of distance and orientation, is accounted for by using another radiometric variable, called spectral *radiant sterance* (or *radiance*), that applies both to the emittance and the irradiance,

$$L(\lambda)=\begin{cases}\dfrac{dM(\lambda)}{d\Omega\cos(\theta_{surf})}\\ \dfrac{dE(\lambda)}{d\Omega\cos(\theta_{surf})}\end{cases}\quad\frac{W}{m^2\cdot m\cdot sr}\qquad\textbf{Eq. 4}$$

- The attenuation in the transmission of the energy from the source to the object of interest, including perturbations and scattering – illustrated in Figure 6 through the interactions with the atmosphere.

In the case of Figure 6, the irradiance of the sunlight on the surface of the object of interest can be expressed as [Fiet12],

$$E_{surf}(\lambda)=M_{sun}(\lambda)\cdot\frac{4\pi r_{sun}^2}{4\pi d_{sun-earth}^2}\cdot\tau_{atm}(\lambda)\cdot\cos(\theta_{surf})\qquad\textbf{Eq. 5}$$

where $\tau_{atm}(\lambda)$ accounts for the attenuation in atmosphere and the calculation considers that:

i) the area of the whole surface of a sphere of radius *r* is $4\pi r^2$;

ii) the solid angle of a whole sphere is 4π;

iii) the flux emitted by the sun is the same in all radial directions;

iv) the sun is seen as a point from the earth, being $d_{\text{sun-earth}}$ the mean distance from the sun to the earth (1.4960E8Km).

As illustrated in the bottom-left part of Figure 6, the object-of-interest, gets irradiated both by the sunlight, represented by $E_{\text{surf}}(\lambda)$ in Eq.5 and by scattered light, represented by $E_{\text{skylight}}(\lambda)$ in Figure 6. Their combined action produces a reflected flux with radiance

$L_{object}(\lambda)$

The exact functional dependance of this spectral density and hence the calculation of the reflected flux may be involved. Calculations can be simplified in practice by assuming the object surface to be *Lambertian* and hence that the reflected energy is evenly distributed over the π stereoradians spanned by the hemisphere above the object surface. When this assumption, and taking into account that the scene has also components coming directly from skylight, the scene radiance, equal for all angular orientations is given by [Fiet12],

$$L_{object}(\lambda)\Big|_{(x,y,t)}\approx\frac{1}{\pi}\Big[\rho(\lambda)\cdot\big\{E_{surf}(\lambda)+E_{skylight}(\lambda)\big\}\Big]\Bigg|_{(x,y,t)}+\frac{1}{\pi}E_{skylight}(\lambda)_{(x,y,t)}\Bigg|\qquad\textbf{Eq. 6}$$

Where $\rho(\lambda)$ is the *reflectance* of the object.

2.1.3. LENSES

As Figure 6 illustrates, before the patterned radiant flux impinges on the array of semiconductor devices where it is transformed into an electrical image, the following processes happen:

- The collection and the transmission of the optical flux by a *lens*, and the subsequent optical focusing of the flux on the sensitive, focal plane, surface
- The transmission and the filtering of the flux by different kind materials placed on top of the irradiated semiconductor surface. Some of these materials are placed intentionally as it happens for instance for *microlenses* and *colour filters*.

To help understanding some basics regarding the impact of the optical system on the overall performance, Figure 8 sows a conceptual lens system. Significant parameters are the *transmittance* of the lens $\tau_{\text{lens}}(\lambda)$, the *magnification factor M=q/p*, the *focal length f*, the *diameter of the aperture* D_{aperture}, and the *F-number* F_N.

One basic relationships among these parameters is obtained by using the Gauss´s thin-lens equation,

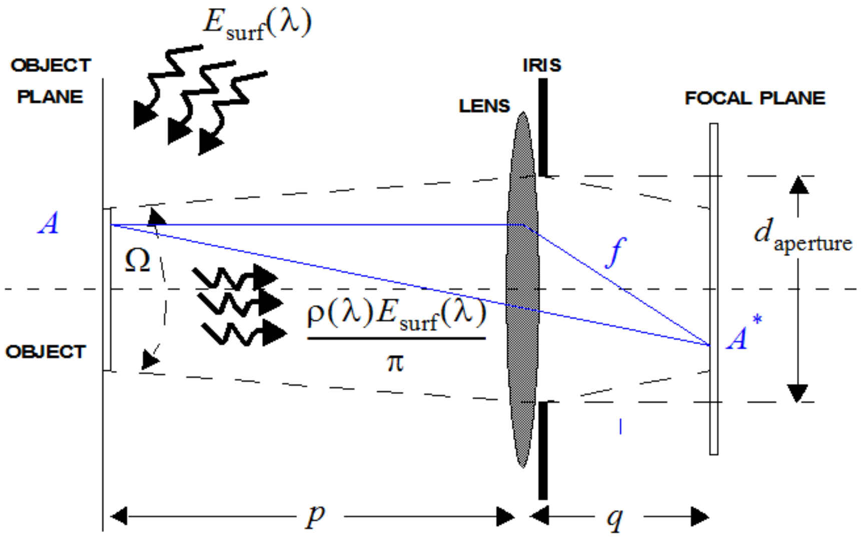


Figure 8. CONCEPTUAL LENS SYSTEM.

This figure has been adapted from R.D. Fiete. Modeling the Imaging Chain of Digital Cameras. SPIE Press 2012.

$$\frac{1}{f}=\frac{1}{p}+\frac{1}{q}\bigg|_{Gauss \quad lens \quad equation}$$ **Eq. 7**

Another basic relationship accounts for the fact that the F-number of an optical system, modeled by an equivalent thin lens, is the ratio of the focal length *f* to the diameter of the aperture,

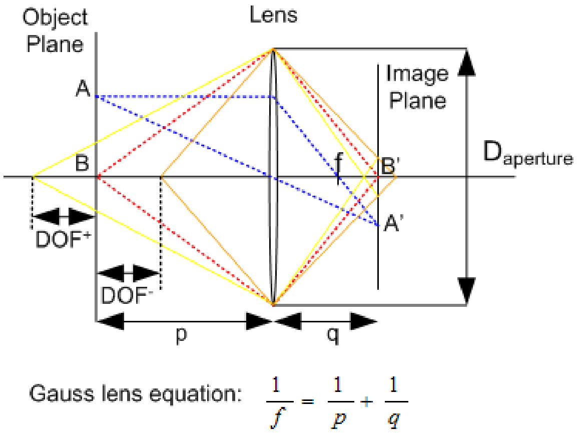
$$F_N=\frac{f}{D_{aperture}}$$ **Eq. 8**

From there the following derived relationships are obtained,

$$p=f\cdot\frac{M+1}{M} \qquad q=\frac{f\cdot p}{p-f}$$

$$\frac{D_{aperture}}{q}=\frac{f}{M\cdot F_N}\cdot\frac{1}{p}$$ **Eq. 9**

A primary impact of the lens is the **reduction of the power** that reaches the sensor plane, which can be calculated as,



$$L_{sensor}(\lambda)=L_{in}(\lambda)\cdot\frac{\tau_{lens}(\lambda)}{4\cdot(F_N)^2}$$ **Eq. 10**

This equation shows that in order to increase the power reaching the sensor an optical system with small F-number should be used.

Another important concept from an application point of view is the **Depth Of Field (DOF)** which is defined as the variation range of the object distance *p* that generates an acceptable circle of confusion. This is a very qualitative definition, since that acceptable value of the circle of confusion is strongly dependent on the application. It is usual to assume the size of pixel as the size for the circle of confusion denoted by coc.

$$coc = \frac{\left|S_2 - S_1\right|}{S_2} \cdot \frac{f^2}{F_N\left(S_1 - f\right)} \qquad \textbf{Eq. 11}$$

Figure 9 describes lens and ray diagram for calculating the circle of confusion diameter coc for an out-focus subject at distance S_2 when the camera is focused at S_1.

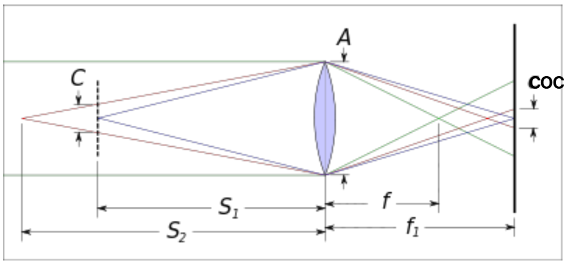


Figure 9. **CIRCLE OF CONFUSION CONCEPT**

Figures from Wikipedia

Besides the loss of light power through the set of lenses, the optical system introduces **crosstal**k. Due to diffraction effect when light crosses the optical system diaphragm, a point in the object plane is transformed into a spot on the sensor plane, whose size X_{spot} is a function of the wavelength λ of light and the F-number of the optical system [Bore98]:

$$X_{spot} = 1.22 \cdot \lambda \cdot \left(F_N\right) \qquad \textbf{Eq. 12}$$

This means that pixels with a pitch significantly lower than X_{spot} will present high crosstalk which is characterized by the **Modulation Transfer Function (MTF).** Therefore, if cross-talk is to be avoided, small pixel pitches require optical systems with lower F-number. However, an optical system with a lower F-number results in lower depth of field (and therefore in a different source of blurring under specific circumstances), a higher cost of the optical system and often in larger nonlinear aberration. The latter together with MTF are important parameters to consider for camera performances.

The generation of picture element values following the exposition of semiconductors to light power embraces two processes:

i) Photons get absorbed by the semiconductor thereby producing electron-hole pairs,

ii) The so-generated electrical charges either recombine themselves or are swept by an electric field thereby producing either a photo-current or the accumulation of a net charge.

These processes are described in the next section of the Chapter.

2.2. BASIC PHOTOSENSORS

2.2.1. PHOTON ABSORPTION

Regarding **PHOTON ABSORPTION**, when light hits the semiconductor, a part of the incident light is reflected while the rest is absorbed in the semiconductor producing electron-hole pairs inside the semiconductor. These electron-hole pairs are denoted by photo-generated carriers. The amount of photo-generated carriers depends on the semiconductor material and is described by the *absorption coefficient* α - defined as the ratio of decrease of light power $\Delta P / P$ when the light travels a distance Δz within the semiconductor:

$$\alpha\left(\lambda\right) = \frac{1}{\Delta z} \cdot \frac{\Delta P}{P} \qquad \textbf{Eq. 13}$$

From here, the light power can be calculated as a function of semiconductor depth:

$$P(z) = P_o e^{-\alpha \cdot z} \qquad \textbf{Eq. 14}$$

where Po is the light power at the surface of semiconductor ($z = 0$). This equation shows an exponential decrease of the light power as light penetrates further within the semiconductor. The decrease rate gets better highlighted by using the inverse of the absorption coefficient, called *absorption length* L_{abs}

$$L_{abs} = \alpha^{-1} \qquad \textbf{Eq. 15}$$

Thus, the absorption length L_{abs} depends on wavelength. In the case of silicon, for visible region (400nm, 700nm), the absorption length lies within (0.1µm,10µm). The absorption length is an important figure of silicon photo-sensing device, see Figure 10.

Regarding the second process mentioned above, namely the **SWEEPING OF PHOTO-GENERATED CHARGES**, CISs typically employ pn-junction structures (photodiodes in short) to achieve this – see Figure 11-left. When a photodiode is illuminated, the current from the cathode to the anode I_L is obtained as the combination of the diffusion current I_D and the photo-generated current, according to the expression:

$$I_L = I_{ph} - I_D = I_{ph} - I_s \cdot \left(e^{\frac{e \cdot V}{n \cdot k_B \cdot T}} - 1 \right) \qquad \textbf{Eq. 16}$$

Where I_{ph} is the photo-generated current – see drawing at the right in Figure 11.

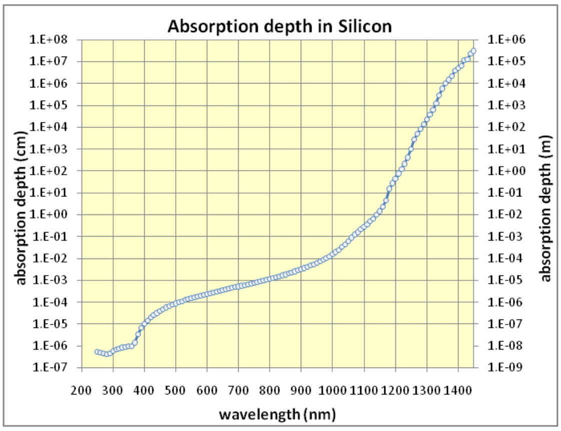


Figure 10. ABSORPTION LENGTH OF SILICON

This figure has been acquired from www.pveducation.org

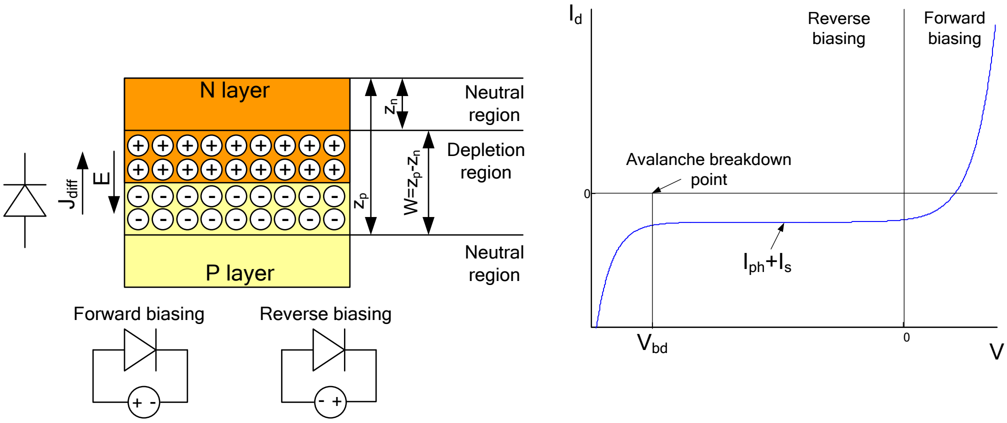


Figure 11. PN-JUNCTION STRUCTURE AND STATIC CURRENT-VOLTAGE CHARACTERISTICS UNDER ILLUMINATION.

Semiconductor pn-junctions can operate in three different modes pertinent to photo-sensing operation, namely

- **SOLAR CELL MODE**: In this mode, no bias is applied to the photodiode. This one acts as a battery under lighting conditions. It produces a voltage across the junction. This voltage is obtained from the open circuit condition $I_L = 0$:

$$V_o = \frac{n \cdot k_B \cdot T}{e} \cdot \ln\left(\frac{I_{ph}}{I_s} + 1\right) \qquad \textbf{Eq. 17}$$

  Examples of CISs whose photo-sensors operate in solar cell mode are presented in [Yang11].

- **AVALANCHE MODE:** The photodiode is strongly reverse biased. The photocurrent suddenly increases due to the phenomena called avalanche, where impact ionization of electrons and holes occurs and the carriers are multiplied. This mode supports the operation of imagers based on Single Photon Avalanche Diodes (SPADs) [Nicl05].

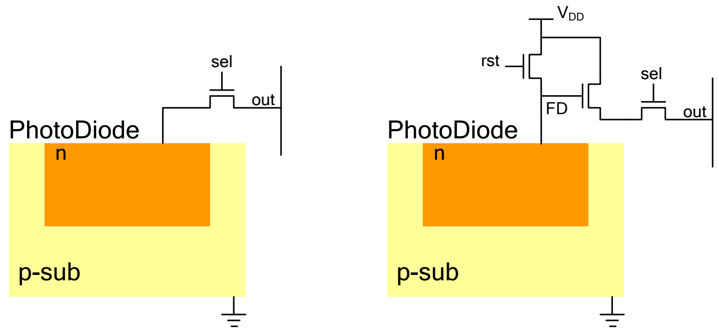
- **PHOTODIODE MODE:** The photodiode is reverse biased. In this case, the exponential term in equation Eq.16 can be neglected, and thus the current through the diode is:

$$I_L \approx I_{ph} + I_s \approx I_{ph}$$

Photo-sensors employed in the systems reported in the Thesis operate in photodiode mode as most of CISs. Within this mode there are two possibilities depending upon whether the photodiode is connected to a resistive load while it is hit by light left floating. The latter, called **ACCUMULATION MODE**, is the most commonly employed.

2.2.2. ACCUMULATION MODE PHOTO-SENSORS

In this mode, the reverse biased photodiode is electrically floated and, photo-carriers are generated and swept to the surface due to the potential in the depletion region. The accumulation of electrons forming the photocurrent makes the capacitor discharging from a reset value. The potential voltage decreases when electrons accumulate. By measuring the voltage drop, the total amount of light power can be obtained. Figure 12 shows two basic photodiode pixel structures conceived to work in accumulation mode. Let us particularly focus on Figure 12(b). For sensing operation, the node labeled Floating Diffusion (FD) is reset to VDD and the photodiode is cleaned of carries; the only electrical charge. Next, the reset transistor is turned OFF and the integration phase starts. Note that in this case, voltage at FD before integration cannot be read out which, as we will see later on has implications on the sensor operation.



a) Passive pixel topology b) 3T active pixel topology

Figure 12. **PHOTODIODE PIXEL OPERATING IN ACCUMULATION MODE.**

There are two important parameters associated to the behavior of photo-sensing element: sensitivity and quantum efficiency.

The **SENSITIVITY** is defined as the amount of photocurrent I_{ph} produced when a unit of power P_o is incident on the material:

$$R_{ph} = \frac{I_{ph}}{P_o} \qquad \textbf{Eq. 18}$$

The **QUANTUM EFFICIENCY** is defined as the ratio of the number of generated photo-carriers to the number of input photons. The number of input photons per time unit is:

$$\frac{P_o}{h \cdot \nu}$$

The number of generated carriers per time unit is:

$$\frac{I_{ph}}{e}$$

being e the electron charge. Thus, the quantum efficiency is determined by the equation:

$$QE = \frac{I_L / e}{P_o / (h \cdot \nu)} = R_{ph} \cdot \frac{h \cdot \nu}{e} \qquad \textbf{Eq. 19}$$

Incident photons penetrate into the semiconductor according to their energy. Photons with smaller energy or longer wavelength penetrate into the semiconductor, while photons with larger energy or shorter wavelength are absorbed near the surface. The photons absorbed in the depletion region are swept by the electric field and accumulated in the potential well.

By using the definition of absorption coefficient given by Eq.13, the light intensity depend on the position within the silicon:

$$dP(z) = -\alpha(\lambda) \cdot P_o e^{-\alpha(\lambda) \cdot z} dz \qquad \textbf{Eq. 20}$$

Considering that each absorbed photon in the depletion region generates an electron-hole pair, the quantum efficiency is equivalent to:

$$QE = \frac{\int_{z_n}^{z_p} \alpha(\lambda) \cdot P_o \cdot e^{-\alpha(\lambda) \cdot z} dz}{\int_0^{\infty} \alpha(\lambda) \cdot P_o \cdot e^{-\alpha(\lambda) \cdot z} dz} = \left(1 - e^{-\alpha(\lambda) \cdot W}\right) \cdot e^{-\alpha(\lambda) \cdot z_n} \qquad \textbf{Eq. 21}$$

Where W is the width of depletion region and z_n is the distance from the surface to the edge of depletion region, see Figure 11**.**

The width of depletion region is given by the equation:

$$W = \sqrt{\frac{2 \cdot \varepsilon_{Si} \cdot (N_d + N_a) \cdot (V_{bi} + V_{bias})}{e \cdot N_a \cdot N_d}} \qquad \textbf{Eq. 22}$$

ε_{Si} is the dielectric constant of silicon, N_a and N_b are the impurity concentrations, V_{bias} the biasing voltage of pn-junction and V_{bi} the built-in potential of pn-junction given by:

$$V_{bi} = k_B \cdot T \cdot \ln\left(\frac{N_d \cdot N_a}{n_i^2}\right) \qquad \textbf{Eq. 23}$$

Where n_i is the intrinsic carrier concentration for silicon.

The limits of depletion region are determined by the equations:

$$z_n = \frac{N_a}{N_{a+N_d}} \cdot W \qquad \textbf{Eq. 24}$$

$$z_p = \frac{N_d}{N_a + N_d} \cdot W \qquad \textbf{Eq. 25}$$

The quantum efficiency curve is dependent on the impurity profile of the n-type and p-type regions as well as position of pn-junction z_j.

Only carriers photo-generated in the depletion region are accounted for; some portion of the photo-generated carriers outside the depletion region diffuses and reaches the depletion region, but in the calculation these diffusion carriers are not account for.

The red and Infra-Red (IR) light penetrates most deeply and reaches the p-substrate layer, where it produces minority carrier electrons. In p-substrate region, there is a weaker electric field, and the mechanism which moves the photo-generated carriers is the diffusion. Part of these photo-generated carriers are recombined and do not contribute to the signal charge, while others arrive at the edge of the depletion region and accumulate in the potential well, contributing to the signal charge. The extent of the contribution depends on the diffusion length of electrons produced in the p-substrate.

The surface-interface states produce deep levels in the middle of the band-gap, carriers near this states (photo-charges generated too close to the surface in the n-diffusion) are easily trapped in these levels. The lifetime in these states is generally long and trapped carriers are finally recombined there. Such trapped carriers do not contribute to the signal charge. Blue light suffers from this effect because the absorption length corresponding to these wavelengths is lesser (Figure 10), and thus has smaller quantum efficiency than longer wavelengths.

The surface of semiconductor is source of other important parameter, the dark current. Dark current is constituted by the carriers not photo-generated. This means that they are generated under dark conditions. The effect of this dark contribution will be studied in sections 3.3.4.1 and 3.4 in chapter 3.

The dark current in photodiode has several sources: the diffusion current (I_s), the surface leak current, impact ionization current, tunnel current, generation-recombination current, etc. In order to decrease the degradation of quantum efficiency for shorter wavelengths and reduce the dark current, the pinned photodiode (buried photodiode) has been developed. The structure of pinned photodiode is depicted in Figure 13**.**

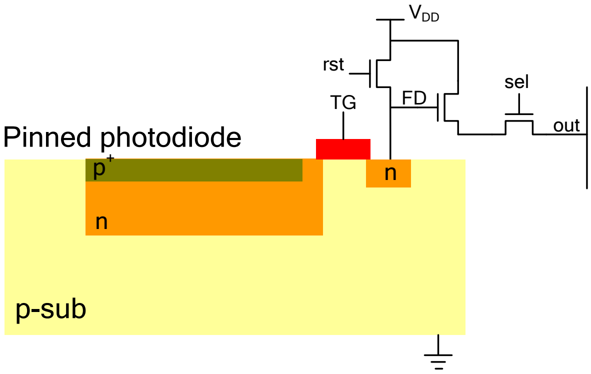


Figure 13. **STRUCTURE OF PINNED PHOTODIODE.**

The topmost surface of the photodiode has a thin p$^+$ layer being buried the photodiode under the surface. In the p$^+$-n-p$^-$ structure both p layers are on substrate potential (GND). As the voltage applied to the n-layer is increased, the depletion regions of both pn-junctions grow toward each other. At a certain voltage, the pinned voltage V$_p$, the depletion regions meet and no more majority carriers can be extracted from the device. The device is fully depleted [Fox98]. The potential then remains fixed inside the device and cannot be increased any further, the voltage is pinned.

The potential profile at the surface is strongly bent so that the accumulation region is separated from the surface where the trapped states are located.

Eventually, the photo-generated carriers at shorter wavelengths are quickly swept to the accumulation region by the bent potential profile near the surface and contribute to the signal charge, improving the quantum efficiency for shorter wavelengths and reducing the dark current contribution associated to the surface, because the charge collection region is separated from the silicon surface through the p^+ layer.

Other benefit of this structure lies in the complete charge transfer from a large accumulation area into the small Floating Diffusion capacitance (FD) through the transfer gate, allowing the implementation of a high electron-voltage conversion factor. In standard photodiode of 3T pixel architecture (Figure 12b), the increment of photodiode area to increase the number of collected electrons involves an increment of capacitance associated to the photodiode. Therefore, the electron-voltage conversion factor decreases. The total number of collected photons increased, but the conversion factor decrease; therefore, a trade-off between collecting area and conversion factor exists. For pinned photodiode, the collecting area and conversion capacitance are independent, allowing the increment of collecting area without increment of conversion capacitance.

For light sensing operation, the pinned photodiode is initially fully depleted. During the integration phase, photo-generated majority carriers are stored in the depletion region (accumulation mode), decreasing the potential of pinned photodiode below pinned voltage V_p. For readout process, the floating diffusion (FD) is first reset to VDD. This reset voltage may now first be read out for true CDS (removing the KTC or reset noise). Next, the transfer gate (TG) is turned on and the complete photo-charge is transfer to conversion capacitance (FD). The complete transfer takes place if the voltage on FD remains above the pinning voltage while the pinned photodiode operates below this voltage.

2.2.3. Other CMOS Photo-sensors

In addition to the accumulation mode, the reverse biased photodiode can operate connected to a MOS transistor working in sub-threshold region, which converts the photocurrent to voltage. This topology, depicted schematically in Figure 14**,** is called logarithmic pixel. High dynamic range images are achieved via logarithmic compression during conversion to voltage, using the exponential I-V characteristic of MOS transistor in sub-threshold.

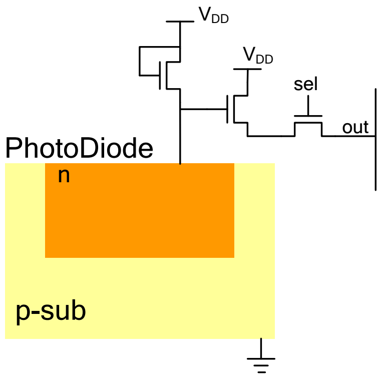


Figure 14. PHOTODIODE PIXEL OPERATING IN LOGARITHMIC MODE.

The structure of photo-gate, depicted in Figure 15, is similar to the MOS capacitor structure. Photo-generated carriers are accumulated in the potential well generated under the gate when a bias voltage is applying on it. For readout, the complete charge can be transferred via the transmission gate TG onto the floating diffusion FD after that had been reset. Thus, most of the photosensitive area does not contribute to the charge-voltage conversion capacitance unlike to the photodiode structure described in the previous section. The resulting high conversion coefficient (denoted by conversion gain) allows for a high sensitivity. However, the sensitivity is mostly lost by the low transparency of the overlying gate material.

As an additional advantage, this reset-transfer structure permits true CDS operation including subtraction of reset noise.

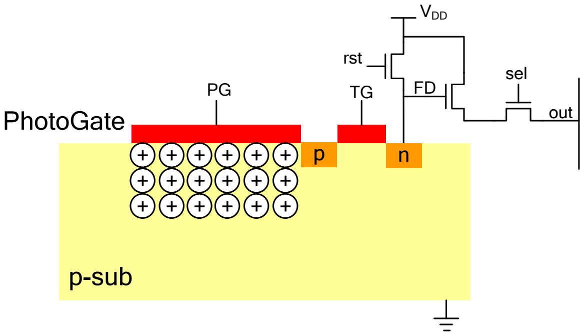


Figure 15. **PHOTO-GATE PIXEL.**

2.3. PIXELS

Pixels are central to the design of imaging devices, including SCISs and CVISs. Pixels include photo-sensitive devices along with devices used for addressing, readout, etc. In same cases, particularly for CVISs, pixels may include a significant percentage of no-sensitive devices. When this happens it is common to use the term cells, instead of pixels, for each of the elements in the front-end array.

Most of the silicon area of SCISs and CVISs chips is occupied by pixels. For instance, pixels take around 80% of the area of last generation, high-speed SCISs devised for machine vision, and approximately 50% of last generation CVISs devised for inspection and factory automation. Note in the illustrative cases of Figure 2 that all pixels are identical and that they are arranged into an uniform, rectangular array. Although this is the most common pixel arrangement, in some special cases the grid may not have a rectangular shape and/or the elements can be unequal. A particularly relevant exception follows a foveated distribution which resembles that of animal retinas. These distributions shows a central region, called fovea, composed of densely packed, small pixels and an increase of the pixel size (correspondingly a decrease of the pixel packing) towards the periphery. Figure 16 shows practical examples of foveated pixel distributions [Pard96].

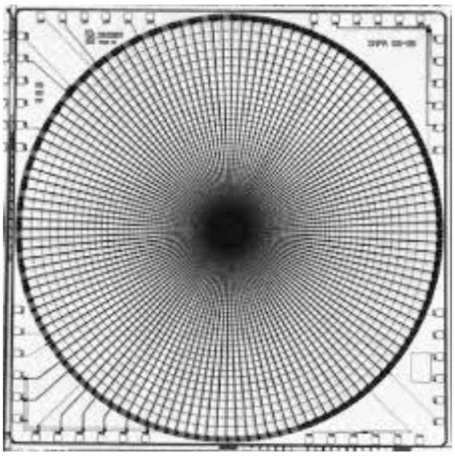


Figure 16. **EXAMPLES OF FOVEATED PIXEL ARRANGEMENTS.**

2.3.1. PASSIVE PIXELS

CMOS pixels include non-sensitive devices (for instance, MOS transistors) in addition to the photo-diode (sensitive area). These non-sensitive devices are included in relation to addressing and readout functions. According to the introduction of active elements at pixel level, CMOS pixels can be divided into two main groups:

- passive pixel sensors (PPS), [Weck67], and
- active pixel sensors (APS), [Nobl68].

The passive pixel (PP) consists of a photodiode and just one switching transistor (selection transistor). This transistor is used as a charge gate, transferring the content of the pixel to a Charge Integration Amplifier (CIA), see Figure 17.

Figure 17 describes the PPS image sensor including the addressing circuitry and readout path. Most modern PPS architectures use a CIA per column in the array.

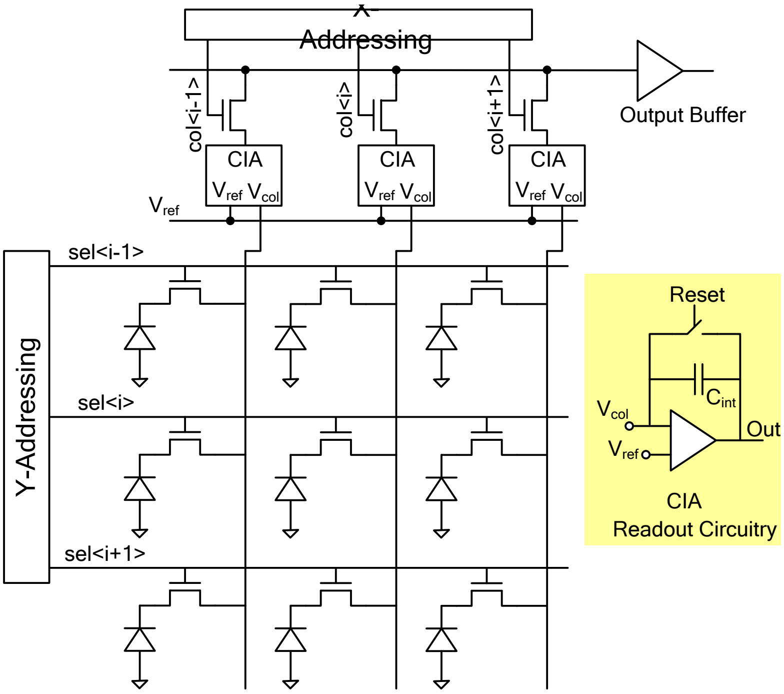


Figure 17. **PPS ARCHITECTURE.**

The voltage V_{ref} is used to reset the photo-diode into the reverse bias stage. After reset operation, the selection transistor is turned off during the integration time, where the photodiode collects charges proportionally to the amount of incident illumination. When the selection transistor is turned on again, the photo-generated charge is transferred from the photodiode to the readout circuit. The total charge that flows to the data column is equal that charge collected during the integration period. This charge is integrated on the capacitance associated to data column and output as a voltage through the CIA circuitry. After reading the signal voltage, during the reset phase where the selection transistor is turned on, the column bus and photodiode voltages return to V_{ref} via the charge amplifier. At the end of reset process the selection transistor is turned off, the charge stored in C_{int} is removed by the reset transistor in the readout circuit, and the integration process start again. Figure 18 shows the control diagram for a PPS.

The out voltage is given by the expression:

$$V_{out} = V_{ref} - \frac{Q_{photodiode}}{C_{\text{int}}} \quad \textbf{Eq. 26}$$

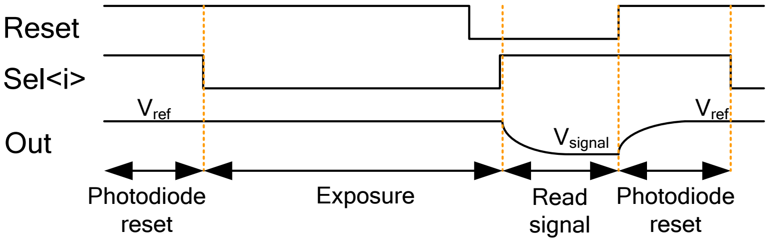


Figure 18. **PPS CONTROL DIAGRAM.**

The passive pixel has the highest fill factor. But, its structure has major problems due to its large capacitive loads at the input of CIA corresponding to the data column.

Previous result in Eq.26 has not considered the finite gain of amplifier in the readout circuit. Since the large bus is directly connected to each pixel during readout, the finite gain of amplifier implies that the load capacitance associated to the large bus affects to the charge integrated in C_{int} reducing the conversion factor.

This large capacitance of column bus means that the RC time constant is very high and the readout speed is slow. In addition, passive pixel readout noise is typically high. The thermal noise generated during the reset process of photodiode has associated a noise power expressed in charge given by the expression:

$$n_e^2 = k_B \cdot T \cdot C \qquad \textbf{Eq. 27}$$

Where k_B is the Boltzmann constant, T the absolute temperature and C the sampling capacitance.

A PPS has a large sampling capacitance (data column capacitance) and hence large noise cannot be avoided.

Because of all these factors, PPS does not scale well to larger array sizes or faster pixel readout rates.

In order to overcome the limitations presented in the preceding paragraphs an active buffer is introduced at pixel level giving rise to the active pixel concept, [Foss92].

2.3.2. In-pixel readout electronics (APS)

An active pixel (AP) includes a local active buffer [Ando97] which is usually a source follower transistor together with the reset and selection transistors. Figure 19 describes the general structure of an active pixel.

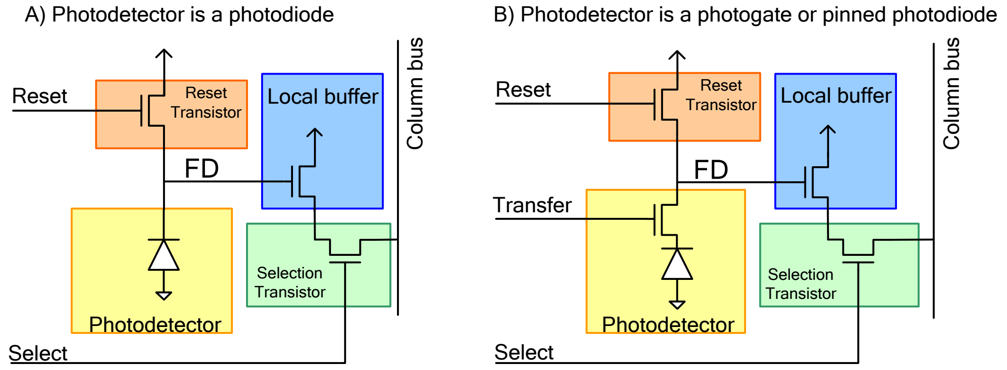


Figure 19. ACTIVE PIXEL SCHEME.

Active pixels typically have a fill factor (relation between the sensing area and the total pixel area) of only 50-70%. This fact implies a reduction in the effective sensitivity area; and therefore, a reduction in the photo-generated charge. However, APS topologies present advantages also. For instance, the reduced capacitance in each pixel leads to lower read noise for the array, what means an increment in the dynamic range and signal to noise ratio.

The active pixels can be divided into three types, depending on the photo-sensitive structure implemented: photodiodes, photo-gates and pinned photodiodes.

The active pixel employs a photosensitive device and a local readout circuit of three transistors: one reset transistor (Reset), one row select transistor (Select) and one source-follower transistor.

Usually, the photodiode AP has a structure called 3T pixel (consist of three transistors) and the pinned photodiode or photo-gate has a structure called 4T transistor, where the transfer gate is considered a transistor, see Figure 19.

The charge-to-voltage conversion occurs at the capacitance of integration node (Floating Diffusion, FD), which comprises the input source-follower transistor and all other parasitic capacitances connected to that node. In the case of photodiode pixel, this capacitance consists of photodiode capacitance, diffusion capacitance of reset transistor and input capacitance of Source Follower. But, in the case of pinned photodiode pixel, the capacitance associated to the photodiode does not contribute to the FD capacitance. Furthermore, the photodiode capacitance is null while photodiode stay pinned (the depletion regions are met).

The pinned photodiode pixel has a significant advantage over the photodiode pixel. This advantage of pinned-photodiode pixel is an integration node capacitance independent from the photo-sensitive area. In case of photodiode pixel, the integration node capacitance includes the capacitance associated to the photodiode, which depends on area and perimeter of latter. Then, larger photodiodes means more sensitive area but also an increment in the integration node capacitance, which implies a reduction in the conversion gain. Therefore, the responsivity is supposedly independent of detector size.

On the contrary, in a pinned photodiode pixel, the photo-sensitive area can be increased to collect a larger number of electrons and the electron-voltage conversion capacitance remains fixed because photo-detection and photo-conversion regions are separated and the pinned photodiode has associated a null capacitance while stays in pinned state (voltage of photodiode remains above the pinned level).

In both pixels, the Source Follower (SF) transistor acts as a buffer to isolate the integration node and the output data column; the load of this buffer (the active-current-source load) is located on each column, which is common for all pixels of column, in order to keep the fill factor high. The reset transistor cleans the photodiode and initializes the sensing node (FD) before charge-to-voltage conversion. It is usually implemented with a NMOS transistor; this allows a higher fill factor. However, an NMOS transistor with VDD on both gate and drain can only reach a voltage of VDD-$V_{T,}$ because the NMOS transistor operates in sub-threshold at this level, and thereby decreasing the dynamic of the pixel.

Generally, 3T pixel operation can be divided into two main stages: the reset phase (consisting of photodiode reset and integration node initialization) and phototransduction phase. For 3T pixel architecture, the reset transistor controls the integration time cleaning the photodiode and initializing the integration node.

During the reset phase, the reset transistor is turned on and carriers of photodiode flow through the reset transistors towards the pixel power supply, cleaning the photodiode. Moreover, the conversion capacitance (C_{FD}) is set to a reset voltage. After that, the reset transistor is turned off and the photo-generated carriers are collected by the photo-diode, the phototransduction phase begins at this moment in parallel with the charge-to-voltage conversion. During phototransduction phase of 3T pixel, the capacitance associated to the Floating Diffusion node is discharged during the integration time by the photo-generated current, which is proportional to the incident light power.

In the case of 3T pixel array, the operation of pixels is performed applying a rolling control technique, described in Figure 20. All pixels in each row are reset and read out in parallel, but the different rows are processed sequentially. This operation mode is denoted by rolling shutter.

After phototransduction phase, the signal value is read out to one Sampled & Hold (S&H) circuit of the Correlated Double Sampling block (CDS). Then, the pixel is reset, and the reset value is read out to a second S&H of the CDS circuit. The CDS circuit, which can be located at the bottom of each column, subtracts the photo-generated signal value from the reset value to obtain the output signal. Its main purpose is to eliminate Fixed Pattern Noise caused by random variations in the threshold voltage of reset and pixel amplifier transistors (Source Follower). In addition, it should reduce the 1/f noise in the circuit and eliminate the reset noise in a true CDS operation. However, for 3T pixel, the CDS circuit subtracts the signal pixel value from the reset value of the next frame. In this case, the CDS operation is not truly correlated, the read noise is limited by the reset noise on the photodiode. In order to implement a true CDS, a memory is required to store the correlated reset voltage. Indeed, the fact that reset phase entails the photodiode reset operation and the initialization of sense node in a 3T pixel makes impossible a true CDS operation.

The operation associated to 4T pixel architecture can be divided into four main stages: the photodiode reset phase, the accumulation or phototransduction phase, the FD reset phase and the charge-to-voltage conversion or transfer phase. The pixel operation is described in next paragraph.

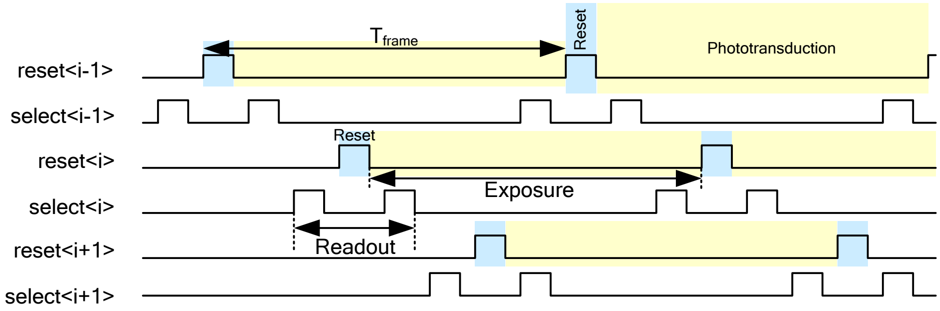


Figure 20. CONTROL SIGNALS FOR 3T PIXEL IN ROLLING SHUTTER MODE.

First, the photodiode is cleaned turning on the transfer and reset transistors. The photo-generated carriers flow to the pixel power supply during the photodiode reset phase. When the transfer transistor is turned off, the photo-generated carriers are accumulated in the photodiode (accumulation phase). Just before transferring the photo-generated charge, the integration node FD is reset turning on the reset transistor. The reset value is read out for CDS operation turning on the selection transistor. After the reset readout is finished, the signal charge accumulated in the photodiode is transferred to the integration node through the transfer transistor, which is turned on. Following, the signal voltage resulting from charge-to-conversion process is read out.

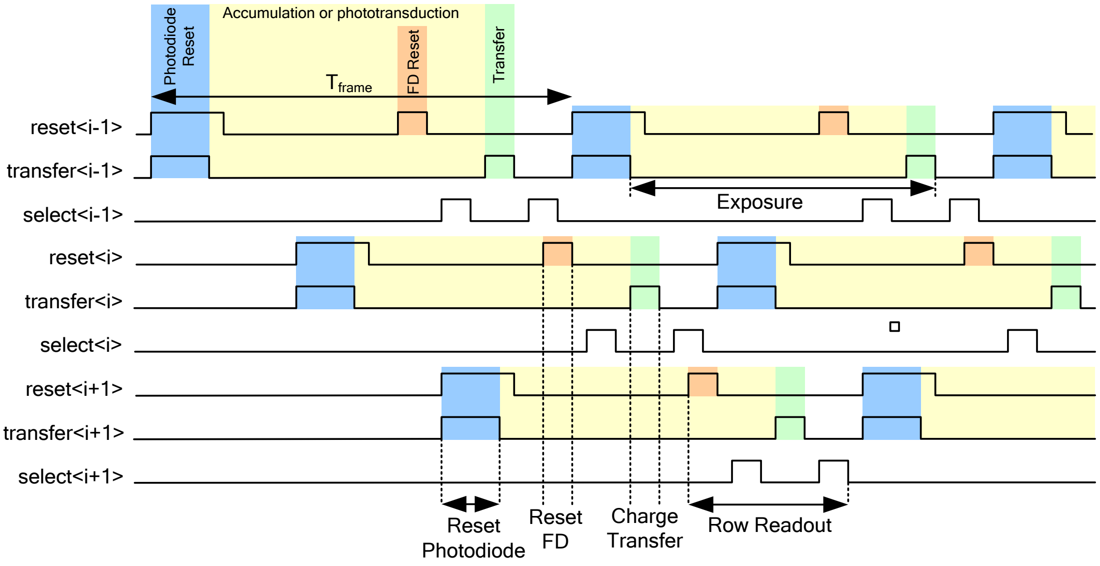


Figure 21. CONTROL SIGNALS FOR 4T PIXEL IN ROLLING SHUTTER MODE.

The pinned photodiode pixel can operate following a rolling technique described in Figure 21. A true CDS operation can be implemented when the 4T pixel works in rolling shutter mode. Since, before transferring the charge from the photodiode to the FD node, this FD node is initialized to the reset voltage and then this value is read out. After that, the transfer is carried out and signal voltage resulting from the integration in FD capacitance is read. In this case, the thermal noise of reset transistor is completely correlated in two readout phases and it can be eliminated. The 4T-APS achieves low noise operation and thus its noise performance is comparable to CCDs.

The rolling shutter mode has an important disadvantage in high speed applications where there are scenes with objects moving at high speed, see Figure 22. In this case, the objects appear distorted in the sensed images because the start and end of the light collection for each row is slightly delayed from the previous row. These objects will be perfectly captured in a global shutter operation.

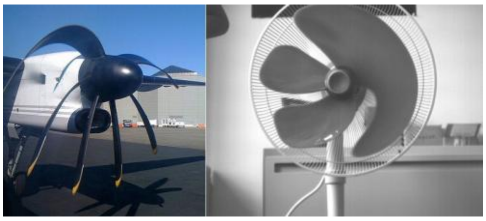


Figure 22. ROLLING SHUTTER MODE FOR HIGH SPEED APPLICATIONS

Figures from Wikipedia Common

The introduction of transfer gate in pinned photodiode pixels makes possible to carry out a global operation in the pixel array. This means that all pixels can operate in parallel. The reset phase of photodiode, exposure and charge transfer phase are carried out in parallel by all pixels of the array, at the same time. The readout process of array follows a rolling technique, see Figure 24. This operation mode is denoted by global shutter. The transfer gate makes possible to implement an electronic global shutter method, [Yadi91]. This technique uses an analogue memory element inside each pixel. In the case of pinned photodiode pixel, the FD node is the analogue memory. For a photodiode pixel, an additional transistor is required in order to implement a 4T structure as it is described in Figure 23. In this case, the shutter transistor operates in a similar way than transfer transistor in pinned photodiode pixel.

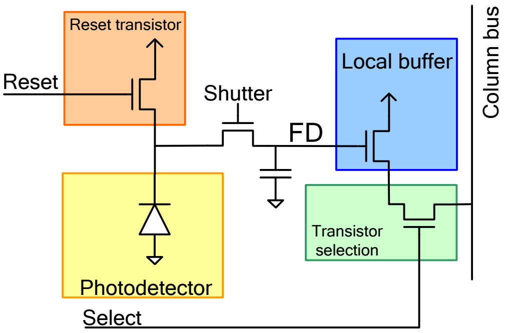


Figure 23. SNAPSHOT STRUCTURE FOR PHOTODIODE PIXEL.

4T structure allows all pixels of array to expose at the same time. After the integration time the signal charge is stored in the analogue pixel memory until readout. An important issue that should be addressed in the global shutter technique is the shutter efficiency. The charge stored in the internal pixel memory is affected by:

- the incident light due to the parasitic sensitivity of diodes associated to the diffusions of FD node
- leakage current of these diffusion.

The effects due to this phenomenon are appreciable in applications with high level of luminosity, short exposure times and low speed in the readout operation. Under these conditions, the value of parasitic exposure corresponding to the floating diffusion could not be negligible respect to the exposure associated to the photodiode operation. In this case, gradients in the output image can be detected. Exposure parameter is given by the product of light power by the integration time.

In order to reduce the impact of FD parasitic sensitivity, layout techniques in the implementation of pixel are considered to cover at maximum level the FD node and minimize the leakage currents.

For instance, in the case of pixel described in Figure 23, shutter efficiency can be increased using a PMOS transistor as a shutter, if it is well separated from the photodiode. Unfortunately, a PMOS shutter decreases the fill factor and, due to increased parasitic capacitances, also decreases the conversion gain.

The global shutter mode does not allow the true CDS operation. Therefore, the reset noise cannot be eliminated. In order to carry out a true CDS, two analogue memories in pixel are called for, one memory to store the reset value before integration and another to store the photo-generated signal.

Regarding the concurrency of acquisition operation and readout process which is demanded in high speed applications, the readout process cannot be carried out in parallel with the photodiode reset phase in the case of pinned photodiode pixels operating in global shutter mode, see Figure 24. This means that the photodiode operation is not completely concurrent with the readout process. In order to achieve the completely concurrence, an additional transfer gate is included in the pixel to clean the photodiode without transferring charge to the FD node, see Figure 25.

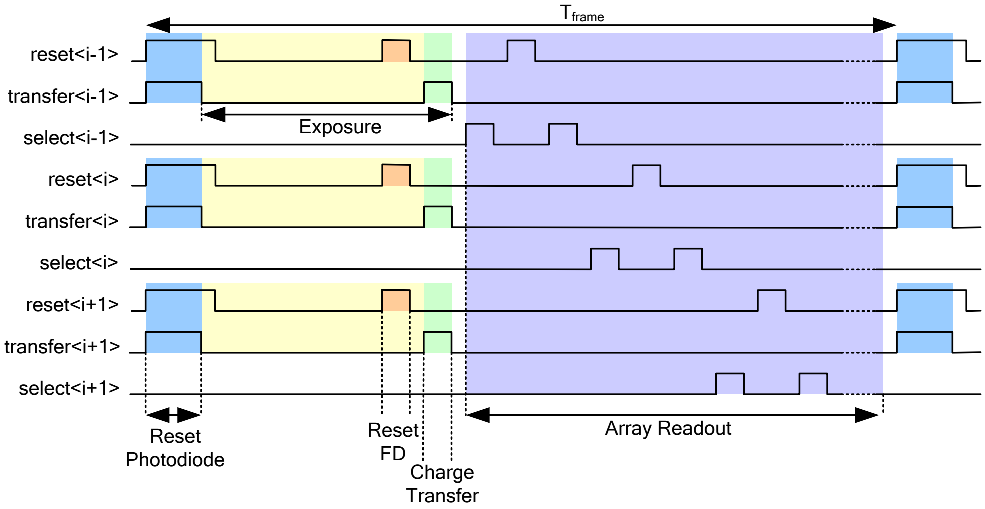


Figure 24. CONTROL SIGNALS FOR 4T PIXEL IN GLOBAL SHUTTER MODE.

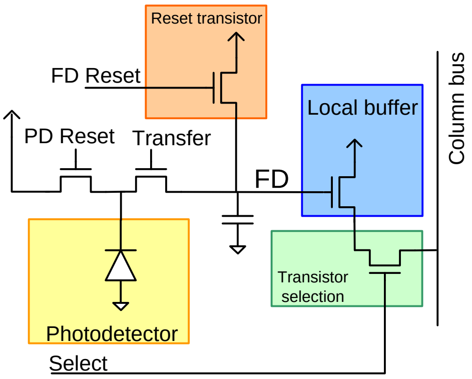


Figure 25. 5-T PINNED PHOTODIODE PIXEL STRUCTURE

In addition, this fifth transistor or second transfer gate is used to avoid a known blooming effect in the photodiode, Figure 26 shows this phenomenon. For this reason, this second transfer gate is denoted by anti-blooming transistor also.

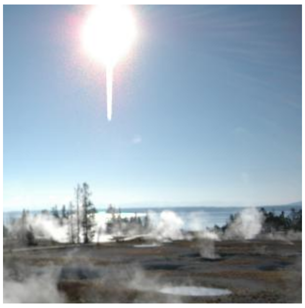


Figure 26. BLOOMING EFFECT

Figures from Wikipedia Common

Blooming occurs when the charge in a pixel exceeds the saturation level and the charge starts to fill adjacent pixels. The excess of charge in the photodiode goes to the substrate and can be collected by electric fields of neighboring photodiodes.

Connecting the gate of antiblooming transistor during exposition to a non-zero value, but enough low to not affect the full well capacity of photodiode, avoid that the electros collected in the photodiode exceed the saturation level.

In the case of 3T photodiode pixel, the blooming effect can be avoided using the reset transistor in a similar way than antiblooming transistor. The gate of reset transistor during exposition will be connected to a non-zero value that does not affect the full well capacity.

Currently, nearly all CMOS sensors employ converters on the same die. There are three general approaches to implement ADC with active pixel sensors. The first approach is a sensor-level ADC, where a single converter reads the whole APS array. This method requires a very high-speed ADC, especially if a very large array is implemented. A widely used ADC architecture in CMOS image sensors is the pipelined approach, [Sira04] and [Paul96].

The second architecture is the column-level ADC, where an array of converters is placed at the bottom of the APS array and each converter is dedicated to one or more columns. All these ADCs operate in parallel, so a low-medium speed converter can be designed, depending of the array size. The disadvantages of this approach are the necessity of fitting each ADC within the pixel pitch and the mismatch among the converters on different columns which is source of Column Fixed Pattern Noise.

The last option is the pixel-level ADC, also denoted as Digital Pixel Sensor, where every pixel has its own converter, [Klei01] and [Jehy03]. The ADCs in the pixel array operates in parallel, allowing the design of a very low speed converter. Therefore, the active circuitry included in the pixel has been increased in this approach, entailing a reduction of fill factor. But, massively parallel conversion and high-speed digital readout become possible eliminating analogue readout bottlenecks completely. This architecture offers better scaling with CMOS technology due to reduced analogue circuit performance demands and the elimination of column and row Fixed-Pattern Noise. The FPN introduced by the pixel ADC is a two-dimensional spatial noise similar to the introduced by the photodiode.

This architecture benefits traditional high speed imaging applications and enables new imaging enhancement capabilities such as multiple sampling for increasing sensor dynamic range. An important disadvantage is the decrement of fill factor, because there is a considerable percentage of pixel area dedicated to the readout circuitry which is not photosensitive.

Tasks related with the early processing level in a Vision System require a high data throughput in the communications and high computational power. These features imply using parallel processing architectures in many high speed and high resolution applications.

An additional progress in the parallelization process of operations associated to the early-processing tasks developed in a Vision System is the introduction of processing circuitry at pixel level (in analogue domain and digital domain), in order to increase the operation speed of Vision System, optimizing the power consumption associated to the computational power required in early processing level.

2.3.3. IN-PIXEL PROCESSING ELECTRONICS

The CMOS imagers make possible to introduce processing circuitry at sensor level with the objective of enhancing the sensor operation and developing the early stages of image processing, which come right after signal acquisition. This approach gives rise to the concept of CMOS Vision Sensor (CVIS), read Chapter 1.

At the beginning of a vision system development, an important step is to decide whether processing circuitry should be inserted into the pixel or placed in the periphery of the array or implemented off-chip. When processing circuitry is included into the pixel (Smart Pixel), additional functions can be implemented in the focal plane, 2-D processing is possible, and neighboring pixels can be easily shared in neural networks. These systems are suitable for real-time applications, which demands high speed operations and usually with moderate accuracy. On the other hand, the fill factor is drastically reduced, making these systems unsuitable for applications where high spatial resolution and very high image quality are required. There is a tradeoff between achievable accuracy and area-power consumption required by the analogue processing circuitry. The area of circuits impacts directly on the fill factor and spatial resolution of image sensor. When the processing circuitry is place in the periphery of pixel array the achievable accuracy is larger because the area associated to the analogue circuitry can be larger, but the speed operation will be smaller because the parallelism level is minor. In this case, the 2-dimensional processing (convolutions, filtering, shifting, etc) is not directly implementable.

The pixel array including processing circuitry gives rise to a multiprocessor architecture, the massively parallel array processor integrated in the sensing plane (Focal Plane Processor). This approach consists of a multiprocessor arrangement in which each unitary processor occupies a node in a two-dimensional lattice, corresponding to a pixel of sensing plane.

Different operations can be included at pixel level. Usually, the circuitry which is introduced in the pixel carries out a specific operation, for instance, circuitry with the objective of improving the intra-frame dynamic range [Yang94][Yadi93], circuitry to detect borders [Yuzo05][Chul10], circuitry for tracking purpose [Indi99][Serr95], Winner-Take-All[6] function has an important role in tracking systems. Most of the existing WTA circuits can be integrated with APS sensors. Usually, when WTA circuits are used in two-dimensional tracking systems, the image processing is included in the pixel.

Vision systems containing smart sensors (CVIS) with a specific function implemented on-chip are designed to carry out a specific task. In order to give greater versatility and ability to program different algorithms and tasks in the smart sensor (programmability and reconfigurability features), several circuits can be introduced in the pixel to develop a group of basic operations. In this way, different algorithms belong to the early processing can be programmed in the smart sensor based on these basic operation set. Examples of such basic operations are arithmetic operations as addition and subtraction, thresholding, gain (multiply by a constant value), convolutions, diffusions, combinational operations (AND, OR, NOT, etc.), local memories, etc, read section 3.1 of Chapter 3. CVIS systems with this kind of pixels are ACE systems (ACE4K and ACE16K [Liña02a]), CACE1K system [Carm01] and Q-eye system which has been developed and described in detail in this Thesis.

A feasible integration of parallel processing with program capability is the SIMD architecture (Single Instruction stream – Multiple Data stream) [Flyn99]. The smart sensors presented above are based on this SIMD architecture. In a SIMD processor, several parallel data paths are centrally conducted by a global control unit, read section 3.1.2 in Chapter 3. The same operation is executed in all data paths, but onto different data. The SIMD architecture is a multiprocessor arrangement in which control hardware is concentrated into a common control unit. Each elementary processor realizes the same operations as the others, upon different data sets. Conditional operations can be carried out by excluding some elementary processor from the execution of an instruction using a binary mask.

The massively parallel array processor has a SIMD architecture where every elementary processor develops the same operation onto its particular data set.

The elementary processor size depends on the accuracy requirements and the number of functions implemented in the pixel. For digital circuits, one more bit resolution implies an increment of circuit area, because the word length is larger and the number of registers increments due to the registers corresponding to the additional bit. For analogue circuitry, larger devices are required in order to achieve higher accuracy. Usually non-ideal behaviors rely on device matching and there is an inverse relation between the statistical deviations of process parameters and device size [Pelg89]. Therefore, in the case of in-pixel processing, the accuracy specification determines the area of pixel and spatial resolution of smart sensor.

For moderate accuracy (8bits resolution), analogue processing represents a faster and more compact approach with a better use of power budget, consequently a more efficient alternative, in comparison with conventional digital circuitry [Moha94]. In the technical literature, VLSI implementations based in Analog Parallel Array Processors (APAP) have been reported [Liña98] [Espe94b] [Domi97], these approaches present a higher computing power rates per area compared to the figure obtained by standard digital implementations. Also, according to the energy consumption per operation, the former are more efficient than latter.

[6] Winner-Take-All (WTA) algorithm selects and identifies the highest input (which corresponds to the brightest pixel of the sensor) and inhibits the rest.

Moreover, maintaining a high accuracy level in analogue processing circuitry, which coexists in the same substrate with an important digital circuitry required for control, scheduling and processing, is a difficult and challenging objective. Therefore, CVIS systems suitable for high speed processing are restricted to applications in which accuracy requirements are moderate.

As mentioned previously, the introduction of processing circuitry in the pixel implies a reduction of fill-factor. Several techniques can be applied to overcome this disadvantage: micro-lenses are widely used to compensate this loss of fill-factor, backside thinning is currently being implemented by several silicon founders (Backside-Illuminated image sensor) [Jane02].

In addition, hybrid sensors appear as a promising option to address the fill-factor issue. It is known that CCD technology is characterized by performance with near-perfect photon interaction and collection of signal carriers. CMOS technology has not demonstrated such high performance. But on the other hand, CMOS arrays allow very fast signal acquisition through parallel processing with low noise, low power, and excellent electronic-circuit compactness. Imaging companies are taking these fine qualities from each technology and they are integrating them into a hybrid system. Hybrid sensors combine the advantages of CCD and CMOS and outperform both them. The Hybrid technology consists of sensing array made in an optimized detection layer on the top of a CMOS layer including the readout circuit [Bai00] [Ando02]. For example, a CCD-to-CMOS hybrid array can be constructed by bump-bonding a high performance CCD to a CMOS signal-processing array. This new architecture is described in Figure 27. It is also possible to mate a backside-illuminated CMOS pixel array to a CMOS readout array [Jane03]. This arrangement allows that CMOS pixel array can be optimized and fabricated independently from the CMOS readout/processing array. Both chips are fabricated independently and the hybrid is thinned and packaged after hybridization process.

The concept of compacting several physical layers, each one associated to a particular hierarchy level of sensing/processing chain is a suitable approach to implement smart sensors including parallel processing and with a fill-factor figure similar to standard CMOS sensors [Garr08], see Figure 27. The readout layer containing a CMOS readout circuit per pixel can include processing circuitry in each pixel. In addition, to attain high resolution images with this approach, a set of n x m pixels in the detection layer can be processed by one processing cell in the CMOS layer. The sensing array is implemented in a physical layer independent from the CMOS processing layer. This fact implies two different substrates for each layer and eliminates the noise influence of processing layer onto the performance of photodiode. Also, the analogue processing circuitry can be designed with a higher accuracy level because this circuitry has more area available. In this new approach, CVIS systems including processing at pixel level can be considered for high accuracy applications (higher than 9bits).

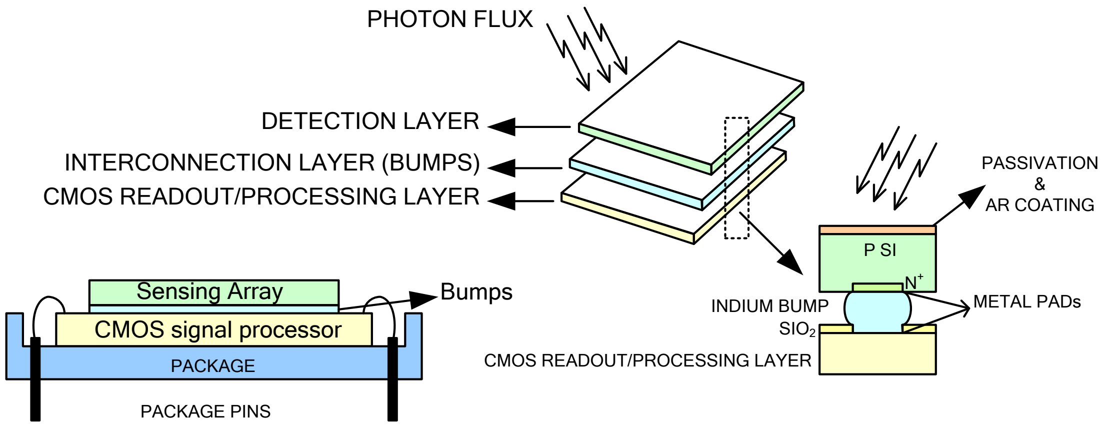


Figure 27. **CMOS HYBRID IMAGE SENSORS.**

Hybrid imagers are an exciting and promising emerging technology which offers optimal features for implementing the Focal Plane Processing approach (in-pixel processing).

2.4. DIGITAL IMAGERS, CAMERA SYSTEMS, VISION SYSTEMS

2.4.1. CONCEPT OF DIGITAL IMAGERS AND CAMERA SYSTEMS

DIGITAL IMAGERS. One of the main assets of CIS technologies comes from the possibility of embedding read-out and processing circuits on the silicon substrate where photo-sensors are built. Indeed, modern CIS-chips substrates incorporate: photo-sensors, addressing circuitry, readout amplifiers, ADCs, image sensor control circuitry and digital processors. Figure 28 shows a conceptual architecture for a modern digital CIS. The larger the number of functions embedded at the sensor the lesser components are needed camera system and the better SWaP factors and manufacturing costs are achieved during camera integration.

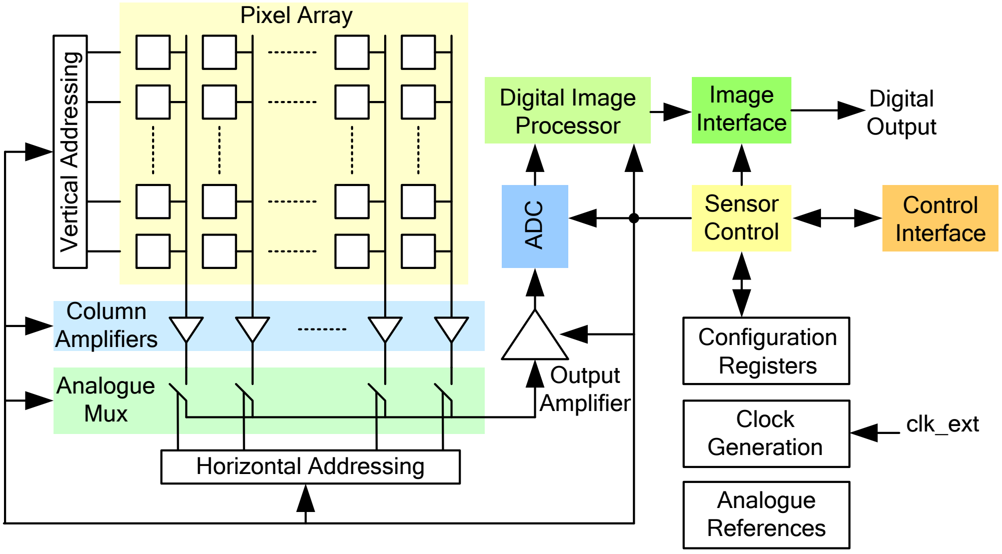


Figure 28. DIGITAL CMOS IMAGE SENSOR.

Architectures like that in Figure 28 enable intensive on-chip interactions between the analog signals delivered by the sensors and digital signals delivered by the ADCs. Thus the embedded **Digital Image Processor** (DIP) can be employed to compensate and correct non-ideal sensor behaviors, and hence to obtain better image quality. Also an internal **sensor control** block is responsible of generating control signals corresponding to all internal blocks. This control block communicates with an **external host** (usually the master control block of the camera system embedding the imager) through the **control interface**, which usually consists of:

i) SPI (Serial Port Interface) or UART (Universal Asynchronous Receiver-Transmitter);

ii) an input trigger signal used to launch expositions; and

iii) several output signals to indicate the state of image sensor; for instance, exposition running, readout process busy or sensor busy. Typical parameters to be configured through the sensor control block include: exposure time, analogue gain applied by the analogue readout path, analogue offset, shutter mode (rolling or global), digital offset, digital gain, region of interest, etc.

The operation of the internal **clock generation block** in Figure 28 is based on a PLL with low jitter. On-chip PLLs are mandatory for large frame rates in order to the frequency demands posed by the **image interface** circuitry, the ADC circuitry and the digital processing circuitry. Commonly LVDS (Low-Voltage Differential Signaling) ports operating at high frequencies (around 1GHz) are employed to reach the data throughput demanded by high-speed applications. All **analogue references**, both currents and voltages, are generated from a master reference which embeds a band-gap circuit.

Figure 28 includes one ADC for the whole pixel array. Thus, pixels are read-out sequentially:

i) rows are selected one after the other;

ii) for each rows, columns are selected one after the others and read-out. This sequential operation poses a constraint on the frame-rate.

For high frame rate applications, several ADCs are employed to parallelize read-out operation. Most common strategy employs a data converter per channel – see Figure 29. Solutions employing several ADCs per channel and ADCs per pixel are considered as well [Rodr15]. Independently of the imager architecture, it is obvious that the electrical performance of the ADCs directly affect to the image quality. Characteristics to be taken into account are the ADC range, resolution, the temporal noise, the conversion speed, non-linearity, etc. ADC architectures and transistor sizes must be properly selected to preclude ADC performances of ADC destroying the targeted image sensor performance.

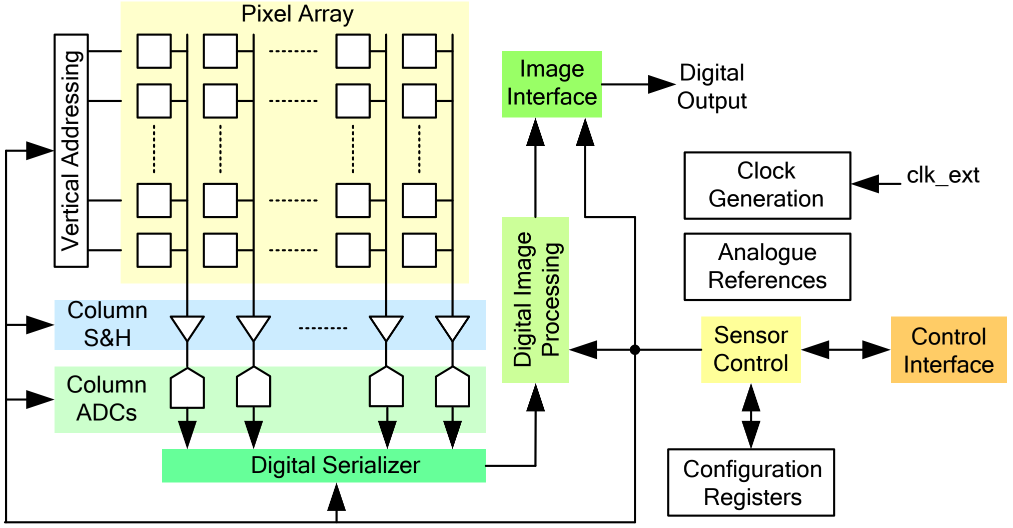


Figure 29. DIGITAL CMOS IMAGE SENSOR WITH PER-COLUMN ADCS.

Architectures with different readout channels working in parallel can produce *artifacts* due to mismatches among the physical circuits implementing the channels. Mismatches produce *systematic errors* which manifest as image artefacts, including Fixed Pattern Noise (FPN).These and other errors will be addressed in section 2.5.5.2. At this point it suffices to mention that on-chip embedding of DIPs allow us implementing different kind of error correction algorithms before delivering digital image data to the outside.

CAMERA SYSTEM. As already stated in Chapter 1, image sensors are placed at the *front-end* of camera systems. Figure 30, which is copied here from Chapter 1 for easier reading, shows the block diagram of a camera system including optics, a front-end image sensor chip and a number of blocks for controlling, data storage and data communications. Image sensors cannot be used as isolated components but must be embedded into camera systems, called simply cameras for practical usage. Besides image sensors and all the functional structures required for control, communication and system integration, camera may also include circuits and systems to analyze image flows and extract information from them.

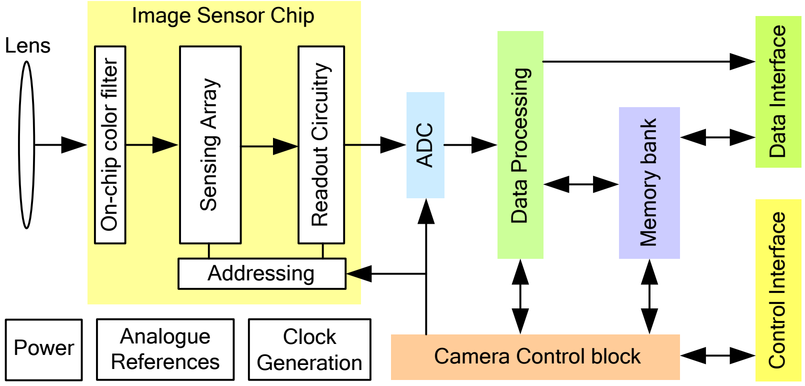


Figure 30. CAMERA SYSTEM BLOCK DIAGRAM.

The block diagram above assumes that ADCs are off the sensing chip. Figure 31 shows a corresponding block diagram in case a digital CIS with embedded ADCs is employed.

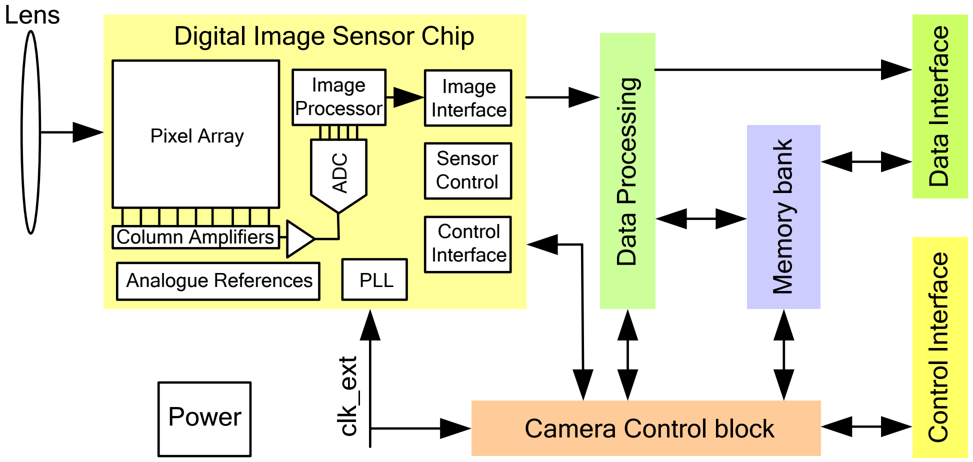


Figure 31. **BLOCK DIAGRAM OF CAMERA USING A DIGITAL IMAGE SENSOR.**

The ***data processing block*** in Figure 31 completes usually different processing tasks related to *color imaging* and *image enhancement*, including: i) color interpolation, ii) color correction or saturation, iii) gamma correction and iv) image enhancement. After that, the ***data interface block*** first handles these processed data to format them according to the selected communication protocol (for instance Camera Link or GigE Vision), and then transmit the capture image to an external host. For instance, the output generator can be an NTSC/PAL encoder to provide standard TV-compatible output, or a video compression engine to provide compressed video streams for communication over network, or digital video output generator such as a Firewire encoder. The image processing functions and interfacing are usually implemented on FPGAs or carried out by DSPs.

The other blocks in Figure 31 are briefly described in the bullet points below:

- The ***camera control block*** is responsible of: i) controlling the operation of the different blocks and in the whole system and in the image sensor (image sensor, ADC, data processing, data interface), and ii) for synchronizing them towards correct system operation. This includes control of clock generation and analog references generations – both crucial for proper imager operation. The control of camera is implemented using either FPGAs, microprocessors or microcontrollers.
- Through the ***control interface*** an external host can configure and control the camera operation. Communication ports, such as Ethernet or RS232 provide the basis for networked camera functionality or camera configuration and firmware upgrading through a PC respectively.
- Digital processing of signals requires storing data in memories. The operation of microprocessors requires memories as well. Therefore, the camera system includes a ***memory bank*** in order to store data and programs.

Notice that camera system (Figure 31) and digital image sensors have similar functional blocks. Main differences are in the data interface and the control interface. If digital image sensors are designed to implement one of several standards communication protocol in order to connect them directly to a frame grabber or PC, then the digital image sensor becomes a **CAMERA-ON-CHIP** [Foss97].

SMART CAMERA. The primary function of a smart camera is to conduct autonomous analysis of the content of an image or video and achieve a high-level understanding of what is happening in the scene. The smart camera produces a high-level understanding of the image scene and generates application-specific data to be used in an autonomous and intelligent vision-based system. Smart cameras are meant to convert data to knowledge by processing information through Application-Specific-Information-Processing (ASIP) algorithms, and to transmit only high abstraction level results that are relevant to the application. Thus, processing in a smart camera goes beyond providing better quality images for human viewing, as it happens in standard cameras and camcorders. Indeed, smart cameras are particular types of *embedded vision systems* intended for decision making in vision-enabled automatic control system [Belb09]. To that purpose smart cameras include sub-systems to directly interact with automation equipment via its discrete I/Os and built-in industrial protocols, like EtherNet.

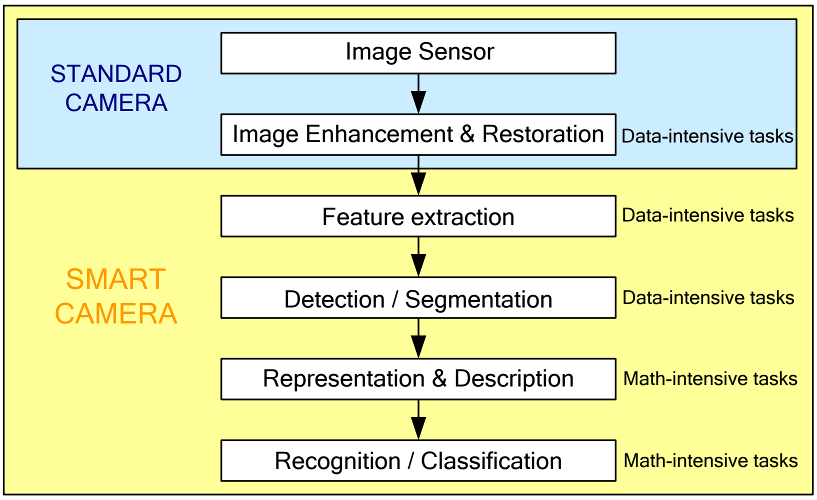


Figure 32. **SMART CAMERA CONCEPT.**

Figure 32 illustrates differences between smart and normal cameras. The data processing block of smart cameras completes early-processing and high-level image processing. Main components of a smart camera are typically the same than those of a normal camera. However, the former embeds powerful signal processing units to perform image feature extraction and/or pattern analysis based on application-specific requirements. Also a smart camera has a output generator which produces a coded representation of image features and/or results from the pattern matching, or in some cases, control signals for other devices (e.g. alarm triggering signal) or actions (e.g. sending a picture of the number of a car which is speeding to police).

There are two major differences between a smart camera and a standard or normal camera. The first is at ***architectural level***. A smart camera usually has a special image processing unit containing one or more high performance processors to run ASIP algorithms seeking to extract information from images. This processing hardware is usually more powerful than that employed for normal cameras where only improving quality matters. The other main difference is at ***data interface level***. A smart camera outputs either the features extracted from the captured images or a high-level description of the scene, which is transmitted into an automated control system. But, for normal cameras the output is the processed version of the captured image for human consumption. For this reason, normal video cameras have large output bandwidth requirements (in direct proportion to the resolution of image and frame rate), while smart camera can have very low data bandwidth requirements at the output.

2.4.2. CONCEPT OF VISION SYSTEMS AND BASIC VISION OPERATIONS

The concept of vision system was brought to scene in this chapter as part of the description of smart cameras. Vision systems are targeted to analyze images and to extract useful information from them in order to solve a determined task. Application areas are large, including: industrial robots, autonomous vehicle, surveillance, people counting, machine vision, computer-human interaction, artificial intelligence, intelligent transportation systems, etc.

Processing algorithms employed in vision systems are based on **COMPUTER VISION** science [Shap91] [Davi12][Szel11]. Computer vision covers a wide range of topics which are often related to other disciplines. For instance, *artificial intelligence* is defined as the study and design of intelligent agents, where intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. The information about the environment could be provided by a vision system, acting as a vision sensor and providing high-level information about the environment. Others fields that are closely related are signal processing, mathematics (statistics, optimization, geometry), image processing, image analysis, etc. Many methods in computer vision are based on these fields. Neurobiology plays an important role, specifically the study of *biological vision system*. There are extensive studies of eyes, neurons and the brain structures devoted to processing of visual stimuli. Results of these studies have been a coarse and complicated description of how biological vision systems operate in order to solve certain vision tasks. Based on these descriptions, a subfield within computer vision has been developed, where artificial vision systems are designed to mimic the processing and behavior of biological systems. Also, some of the learning-based methods developed within computer vision have their background in biology.

Some main functions carried out by general vision system architecture are listed in the bullet points below:

- **IMAGE ACQUISITION.** Digital images are produced by one or several cameras. Generally, depending on the type of sensor, the resulting image data are either ordinary 2D images, or 3D volumes, or image sequences. The pixel values can represent several physical measures such as: light intensity in one or several spectral bands (gray images or color images), depth, absorption or reflectance of sonic or electromagnetic waves, nuclear magnetic resonance, etc.
- **PRE-PROCESSING**, including image *enhancement* and *restoration*. In general terms, they are similar in that both pursue to deliver images that are better suited for given application than original provided by the sensors. More specifically, restoration attempts also to reconstruct or recover an image that has been degraded by using some a priori knowledge of the degradation phenomenon.
- **FEATURE EXTRACTION.** Image features of various complexity levels can be extracted: edges, ridges, localized interest points, corners, blobs, etc. A key problem here is choosing those features that will achieve significant data reduction while preserving the information that will allow subsequent algorithms to distinguish items of interest in images.
- **DETECTION/SEGMENTATION.** Identification of relevant points and regions of the image for further processing, subdividing the image into its constituent parts or objects. Segmentation is one of the most important elements in automated image analysis because it is at this step that objects or entities of interest are extracted from the image for subsequent high-level processing, such as description and recognition.
- **REPRESENTATION AND DESCRIPTION.** After an image has been segmented into regions, the resulting aggregate of segmented pixels must be represented in a form suitable for further computer processing. Basically, there are two methods for representing a region:
  - Representation based on its **external characteristics**
  - Representation in terms of its **internal characteristics**

  Generally, an external representation is chosen when the primary focus is on shape characteristics (*morphological* characteristics), while an internal representation is selected when application is based on reflectivity properties (*color* and *texture*).

  Choosing a representation scheme, however, is only one part of the task of making the data useful for high-level processing algorithms. The next task is to describe the region based on the chosen representation. For example, in an external representation based on boundary of regions, the boundary can be described by features such as its length, the orientation of the straight line joining the extreme points, the number of concavities in the boundary, etc.
- **RECOGNITION (classification and estimation).** Pattern recognition is the assignment of some sort of output value or label to a given input value or instance, according to some specific algorithm. Classification is one of tasks included in the pattern recognition field, which attempts to assign each input value to one of a given set of classes. Pattern recognition assumes that the image may contain one or more objects and that each object belongs to one of several predetermined types or classes.
- **SEMANTIC LEVEL.** The main tasks at this level include possible joint analysis of recognition phase results from one or several cameras, other sensory and data base inputs, data fusion, event description and control signal generation. The algorithms interpret the object classification outputs and make decisions.

As already mentioned in Chapter 1 (see Figure 5 and its associated wording), vision involves a chain of processing functions through which get progressively reduced. On the one hand, representation, description, recognition and semantic level may be roughly grouped under the label of **HIGH-LEVEL PROCESSING** functions. These correspond to math-intensive tasks involving typically small data sets. On the other hand, image acquisition, pre-processing, feature extraction and detection/segmentation are roughly grouped as **LOW-LEVEL PROCESSING**. These are data-intensive tasks where the amount of data associated to the images is very large requiring a high data throughput in the communications and high computational power for applications demanding high resolution and high frame rate.

The relevance of each processing function within the vision processing chain is largely application-dependent; and applications may range from relatively simple targets, such as counting bottles in a production line, to building robots that are to able comprehend the world around them based on the combination of vision and other senses. This application-dependence has an obvious impact on implementation, both at the hardware and the software level.

2.4.3. HARDWARE-SOFTWARE CO-DESIGN

Vision systems consist of *hardware* and *software* components. The value of system can be measured by some objectives that are specific to its application domain (performance design, manufacturing cost, ease of programmability) and it depends on both the hardware and the software components. Hardware/software co-design pursues meeting system-level objectives through the synergistic combination and concurrent design of hardware and software. Hardware and software must be designed together to guarantee proper operation under demanding power, cost and reliability constraints.

Hardware/software system designers faces the same problems as ASIC designers: characterizing the components, understanding the operation of component networks and choosing a network topology based on the requirements. While a great deal of research has addressed methods for pure hardware designs and pure software designs, the increasing number of applications requiring embedding calls for new hardware/software design methodologies capable of meeting performance goals under SWaP and cost constraints.

In general, different co-design problems are found and design approaches are developed depending on the classes of digital systems. To characterize these systems, several general criteria are considered, such as domain of application, degree of programmability and implementation features.

DOMAIN OF APPLICATION. Digital systems can be classified according to their principal domain of applications. A digital system can provide a service as a self-contained unit, or as a part of a embedded system. Examples of self-contained digital systems are information processing systems, ranging from laptop computers to super-computers. Applications of embedded systems are present in the manufacturing industry (plant and robot control), consumer products (intelligent home devices), in vehicles (control and maintenance of cars, planes ships), in telecommunication applications, and in territorial and environmental defense systems.

Digital systems can be geographically distributed (telephone network), locally distributed (aircraft control with different processing units on a local area network), or lumped (workstations).

DEGREE OF PROGRAMMABILITY. Most digital systems contain hardware which is programmed by some software to perform the desired functions. Hence the abstraction level used for programming models is the means of interaction between hardware and software. There are two key issues related to programmability level:

- **Levels of programming:** Digital systems can be programmed at different levels: application, instruction and hardware levels.

  The highest abstraction level is the **application level**, where the system runs dedicated software programs that allow the user to specify determined functionality options using a specialized language.

  Common components in digital systems are microprocessors, microcontrollers and programmable DSPs, which have an *instruction set architecture*. The instruction set defines the boundary between hardware and software by providing a programming model of the hardware. **Instruction-level programming** is achieved by executing on the hardware the instructions supported by the architecture. In some applications domains, such as data processing for telecommunications and image processing, it has been demonstrated that replacing standard processors by ASIPs is practical. ASIP are instruction-level programmable processors with an architecture tuned to a specific application [Goos97]. In an ASIP design, the instruction set and hardware structure are chosen to support efficiently the instruction mix of the embedded software for specific application. Unfortunately, this special instruction set implies developing application-specific compilers.

Hardware-level programming means configuring the hardware, after manufacturing, in a desired way. An example is microprogramming, where the behaviour of the control unit is determined by a micro-program, which is stored in a memory. Today microprogramming is common for DSPs, but not for general purpose microprocessors using RISC architectures, mainly due to performance reasons [Henn90]. Reconfigurable circuits, whose main representative are FPGAs, are the most significant example of hardware-level programming.

- **Access to programming**: This item defines who has access to what level of programmability. The end-user programming is often limited to application-level programming. An application developer relies on the programming language tools, operating system and the high-level programming environment for application development. A system integrator uses the system programmability in order to ensure compatibility of system components with market standards.
- **Implementation features**: System implementation deals with circuit design style, manufacturing technology and integration level. The circuit design style relates to the selection of circuit primitives, clocking strategy and circuit operation mode. A system may have components with different scale of integration and different fabrication technologies. The choice of hardware technology for the system components affects the overall performance and cost.

  System-level field programmability can be achieved by storing programs in memories and exploiting programmable interconnection.

  The architectural and performance characteristics associated to the components comprising a system determined the strategy to develop in the Hardware/Software co-design.

2.4.4. DESIGN OF HARDWARE/SOFTWARE SYSTEMS

The design of Hardware/software systems involves modeling, validation and implementation. **Modeling** is the process of conceptualizing and refining the specifications, producing a hardware and software model. **Validation** is the process of achieving a reasonable level of confidence. And **implementation** is the physical realization of hardware through synthesis and of executable software through compilation.

The modeling style can be *homogeneous* or *heterogeneous*. In the first case, a modeling language (C programming language) or a graphical formalism is used to represent both the hardware and software portions. In the second case, the hardware/software partition is often outlined by the model itself, because hardware and software components can be described in the corresponding languages.

Hardware/software system design is composed by four main tasks [Wolf92]:

- **Partitioning** the function to be implemented into smaller, interacting pieces. The partition of a system into hardware and software is critical because it has a first order impact on the cost-performance characteristics of the final design. Given a specific architecture, partitioning of a system-level functional description results in a labelling of its tasks as hardware or software operations. Partitioning can be decided by the designer, with a successive refinement and annotation of the initial model, or determined by CAD tools. Several approaches to partitioning task have been developed [Barr94] [Agra95] [Bend96] [Vahi94] to improve system performance while executing specific algorithms

- **Allocating** those partitions to microprocessors or other hardware units, where the function may be implemented directly in hardware or in software running on a microprocessor.
- **Scheduling** the times at which functions are executed. Scheduling can be defined as assigning an execution start time to each task in a set, where tasks are linked by some relations (dependencies, priorities, etc.). The tasks can be elementary which are denoted by operations (hardware operations or computer instructions) or can be an ensemble of elementary operations which is denoted by process (software programs). Task execution may be subject to real time constraint or not. Tasks execution requires the use of resources, which can be limited in number, thus causing the serialization of some task execution. The scheduling tasks are present at different abstraction levels during the design phase: operation scheduling in hardware level [Knap96], instruction scheduling in compilers [Aho88] and process scheduling in operating systems [Rama94].
- **Mapping** a generic functional description into an implementation on a particular set of components, either as software suitable for a given microprocessor or logic which can be implemented from given hardware libraries.

The design goals in each task depend on the application (performance, manufacturing cost, testability, etc.). The solutions to these problems clearly interact, the available choices for scheduling are controlled by how the design was partitioned, so on. Furthermore, not only can each of these steps be applied to the software and hardware components separately, but also to the division into hardware and software components itself, and the design decisions made for the hardware and software components separately interact with the co-design problem.

2.4.5. VISION SYSTEMS ARCHITECTURES

The organization or architecture of a vision system is highly application-dependent. Some are *stand-alone* systems which solve a specific measurement or detection problem, while others are sub-system of a larger entity that include other sub-systems for control of mechanical actuators, planning, information databases, man-machine interfaces, etc. The architecture of a vision system also depends on whether its functionality is pre-defined or it can be modified during operation.

In general, a standard vision system includes a camera or several cameras, a data processor responsible for carrying out the image processing algorithms, a system controller responsible for the configuration and control of system, memories to store images and programs, communication ports, a display unit and libraries of image processing routines included in a software environment conceived for writing, compiling, executing and debugging Vision Applications. Figure 33 is a schematic representation of a vision system.

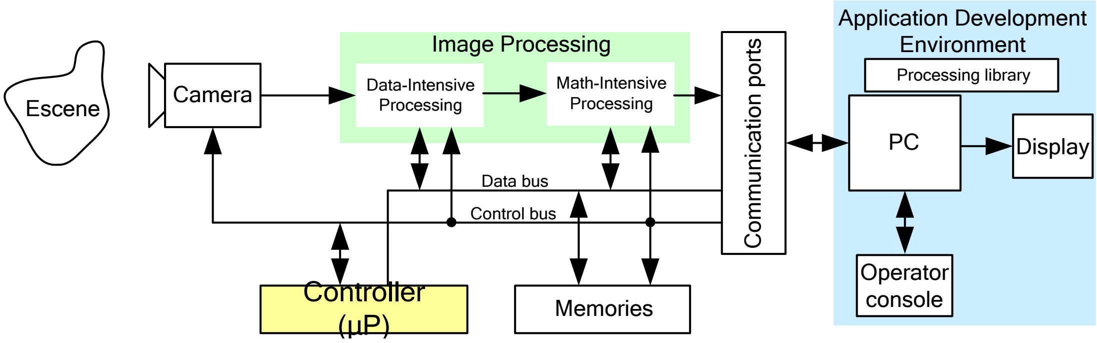


Figure 33. VISION SYSTEM DIAGRAM.

Physical vision system architectures can be roughly classified in the following groups:

- PC-based vision systems
- Embedded vision systems
- Network based vision systems

- Hybrid vision systems

PC-BASED VISION SYSTEMS are straightforward, flexible and robust, although bulky and slow unless high-performance processing are employed. This type of system relies on the PC's CPU to perform image analysis, feature extraction and pattern recognition tasks. The availability of various vision processing libraries for PC platforms makes this approach very popular and very well suited for fast application prototyping [MATROX]. PCs also provide a more flexible environment for building user interfaces.

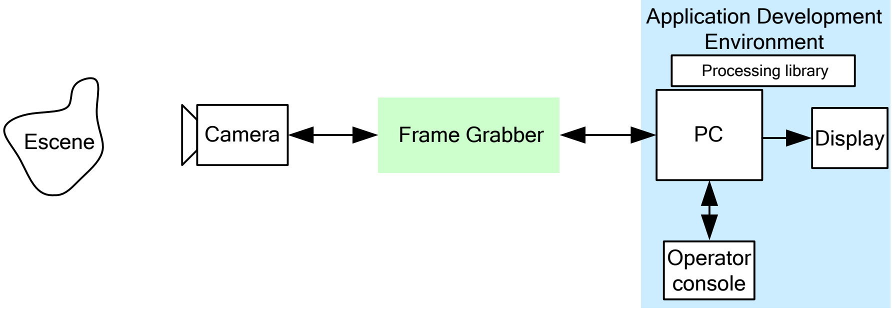


Figure 34. PC-BASED VISION SYSTEM.

PC-based vision systems include usually a camera connected to a PC through either a *frame grabber* or a communication port such as USB, Firewire, CameraLink or Ethernet, and usually an Application Development Environment (ADE), see Figure 34. Historically, frame grabbers have been the predominant interface. But, trends have changed in recent years; direct camera connections via USB, CameraLink or Ethernet interfaces have became practical and will be prevalent in the near future in order to reduce the system cost. Sometimes a camera is connected to a PCI (Peripheral Component Interconnect) processing board within a PC. In this case, the PCI board may perform most of the ASIP and output generation, while the PC provides a flexible operator interface or additional processing power. This kind of system is a special case of a compact vision system and a PC-based system. Compact vision systems are usually composed by a camera connected to an external embedded processing unit in a separate box to provide ASIP and communication/networking functionality.

An **EMBEDDED VISION SYSTEM** integrates image capture, all processing required by the task to perform by the vision system and application specific output generation into a single device casing – Figure 35 depicts the architecture of embedded vision system. While data-intensive tasks require high speed hardware to deal with high resolution and high frame rate, math-intensive tasks often require high performance processors to deal with issues such as pipelining and floating-point arithmetic. For demanding applications, hardware architecture of embedded vision systems can be based on a *multiple-processor* platform, with one or more processors capable of implementing parallel processing (FPGA) performing data-intensive tasks, and a DSP or RISC (Reduced Instruction Set Computer) performing math-intensive tasks. Figure 35 depicts the architecture of embedded vision system. Smart cameras are actually a particular type of embedded vision systems. Smart cameras offer the benefit of interacting directly with automation equipment via its discrete I/Os and built-in industrial protocols, like EtherNet. Unlike a PC-based vision system, which requires a separate discrete I/O card and the integration of third-party software to support these industrial protocols, a smart camera can support these devices. This smart camera concept as an embedded vision system has advantage over PC-based architecture, more compact volume and achieves lower cost and power consumption. This last parameter is essential for stand-alone applications. But, smart cameras have strengths and weaknesses like PC-based vision systems. Historically, a smart camera has been limited in terms of its processing capabilities. It does not offer the computing power of a traditional PC-based vision system because it cannot dissipate the amount of heat that is generated by high performance but power-hungry CPUs. Usually, a smart camera offers a software environment that does not require traditional programming or coding. Typically, the smart camera is designed to perform a specific task. Because the majority of smart cameras are configured, they are well-suited to users with little or no traditional programming experience. The disadvantage to this is that a user cannot work outside the mold; his application must match what the smart camera supports. However, applications that require complex logic in terms of numerous decision points, alternate path of execution and fallback mechanism, as well as very specialized algorithms, are better suited to traditional programming, which is the realm of PC-based systems but also available with some smart cameras.

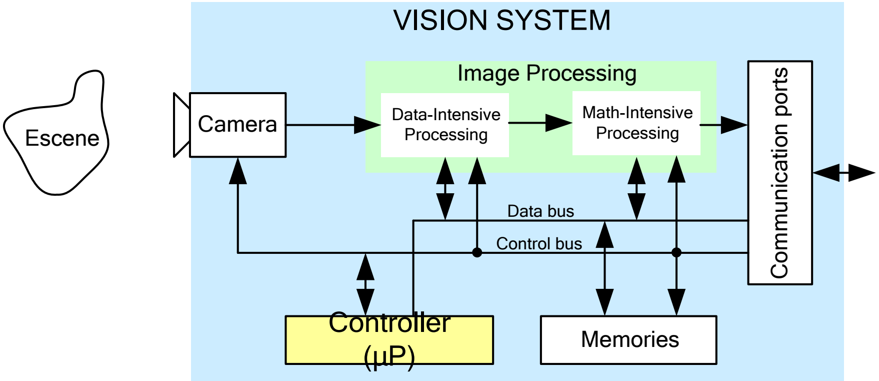


Figure 35. **EMBEDDED VISION SYSTEM.**

Compared with PC-based systems, embedded vision systems are subject to hard constrains from the point of view of design, implementation and production. Examples of typical restrictions are low power consumption, limited resources, real-time processing and low cost. These constrains proceed from the fact that embedded systems are usually considered for the implementation of products with a high level of compactness. Currently, embedded vision systems have progressed fueled by the availability of: powerful processors and memories with large compactness level and high operation speed, real-time operating systems, development of algorithms more suitable for computational implementation and the improvement of system development software and tools, among other factors.

Traditionally, embedded vision systems are designed for custom tasks. Hence, their hardware can be tailored to implement only those indispensable algorithms with minimum power and cost. The downside to this approach is that the user application must match what the embedded system support. However, by incorporating smart dose of programmability and configurability, for instance, by using a micro-processor for system control and including configurability in the operation of ASIP hardware, users can adapt the system for a variety of applications using an ***application development environment*** similar to PC-based system. But obviously, there is a trade-off between compactness-power consumption-cost and versatility.

In general, vision systems are usually embedded into larger systems executing monitoring-control functions and performing information-processing functions. In many practical applications embedded control systems regulate mechanical components based on image analysis. Hence, in the case of real-time applications, image analysis must be completed within prescribed time windows for proper operation [Shin94]. Since vision systems can be visualized as embedded systems working in real-time, their design methodology is similar to that followed in the development of embedded systems used in other application fields [Wolf92]. Indeed, many embedded systems are implemented as distributed systems, where hardware/software partition represents a physical partition of system functionality into application-specific hardware and software executed on one or more processors with inter-processor communication links between them. In the case of vision systems, the application-specific hardware performs the data-intensive tasks belonging to the early-processing stage. This specific hardware is implemented by custom circuits (ASICs), reconfigurable circuits (FPGA) or DSP's. The system control and high-level image processing will be performed by one or several processors (microcontrollers and microprocessors). For example, in the Eye-RIS system proposed in this Thesis, the application-specific hardware is composed by the Q-Eye and a digital image processor, both controlled by a dedicated controller. And the software is executed by the microprocessor NIOSII. While detailed explanations are provided in Chapter 5, at this point it is convenient to copy the Eye-RIS block diagram for easier reading – see Figure 36.

These embedded systems can be characterized by a co-processing architecture, where a micro-processor works in conjunction with dedicated hardware to deliver a specific application. The particular implementation of a co-processing architecture depends on the level of parallelism supported between hardware and software components. The co-processing hardware can operate under direct control of the microprocessor, which stalls while the dedicated hardware is operational. On the other hand, the co-processing hardware can operate concurrently with software execution.

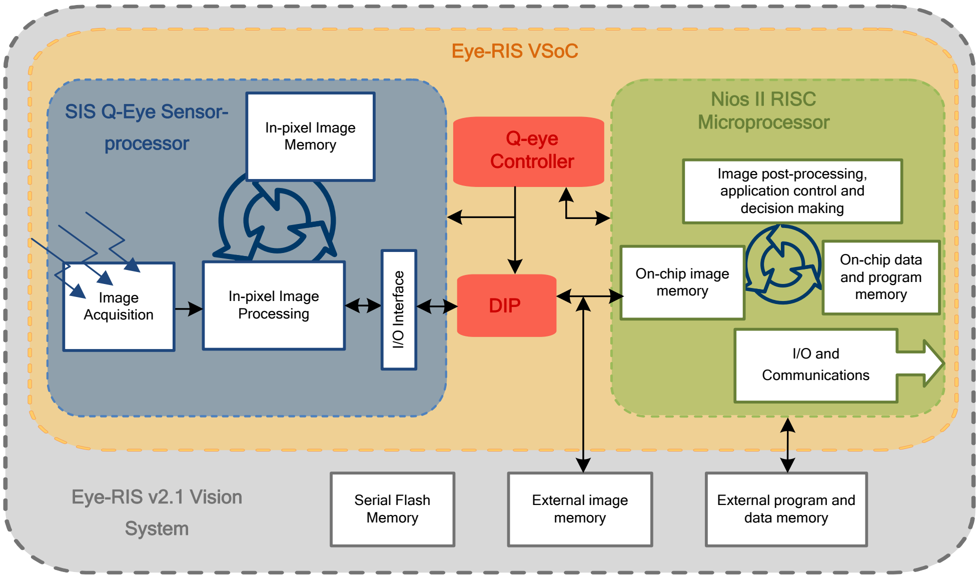

Eye-RIS VSoC
SIS Q-Eye Sensor-processor
In-pixel Image Memory
Image Acquisition
In-pixel Image Processing
I/O Interface
Q-eye Controller
DIP
Nios II RISC Microprocessor
Image post-processing, application control and decision making
On-chip image memory
On-chip data and program memory
I/O and Communications
Eye-RIS v2.1 Vision System
Serial Flash Memory
External image memory
External program and data memory

Figure 36. **BLOCK DIAGRAM OF EYE-RIS_v2.**

These embedded systems can be characterized by a co-processing architecture, where a micro-processor works in conjunction with dedicated hardware to deliver a specific application. The particular implementation of a co-processing architecture depends on the level of parallelism supported between hardware and software components. The co-processing hardware can operate under direct control of the microprocessor, which stalls while the dedicated hardware is operational. On the other hand, the co-processing hardware can operate concurrently with software execution.

Other factors which affects the partitioning formulation is the number of processors considered for the target architecture and the hardware/software interface.

According to scheduling tasks, timing constraints are common for hardware circuits and software applications in embedded control systems in which the vision system can be classified. During concurrent operation of hardware and software portions, there is an important issue related to scheduling, the synchronization of several operations and processes. Synchronization is needed when some delay is unknown in the model. Relative scheduling is an extended scheduling method to work with operations with unbounded delays [Ku92].

When considering compilation for general-purpose microprocessors, instruction selection and register allocation are often achieved by dynamic programming algorithms [Aho88]. But in embedded systems, the dedicated hardware is usually implemented by an ASIP, thus resulting a more complex compiler for this special processor because of irregular structures such as inhomogeneous registers sets and connections. For instance, the vision system Eye-RIS has required the development of a particular compiler associated to the microprocessor included in the Q-Eye controller in order to translate the specific programming language associated to the Focal Plane Processor (FPP) Q-Eye to the micro-instructions that Q-Eye controller will transmit to the Q-Eye system. The Q-Eye system together its controller and the digital image processor constitute the application-specific hardware included in the vision system Eye-RIS. In this case, the specific hardware thanks to the programming capability of Q-Eye has associated a set of instructions that give rise to the CFPP code, which is the language used for programming the CVIS Q-Eye.

Once introduced the embedded vision systems, another type of vision systems will be presented, which belongs to a higher hierarchy level from an architectural point of view, the **NETWORK BASED VISION SYSTEMS**. They are composed by a network of cameras (usually smart USB or Firewire cameras) that acquire and processing images locally to be subsequently sent to a central computer (server) or processing hardware in order to process the received information and take a decision. Currently, this type of vision systems is widely implemented in surveillance applications, see Figure 37. A complex video surveillance system with built-in local intelligent features extraction can be loosely considered as a network of smart cameras. Commonly, the video surveillance system is composed of several layers: camera layer, network layer, central computer layer and take-decision layer which consists of trained security personnel. Intelligent video surveillance systems carry out a re-distribution of ASIP tasks among the four layers, notably shifting processing load from security personnel to central computers and distributed smart cameras. The use of smart cameras reduces significantly the bandwidth problem caused by increment of cameras present in the system and their accuracy specs. Clearly, the transmission of raw pixels over the network is less efficient than sending the intermediate analysis results developed by the smart cameras.

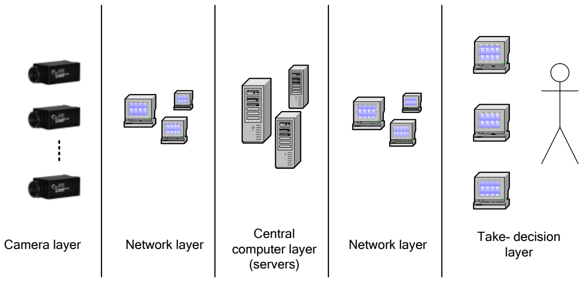


Figure 37. **VIDEO SURVEILLANCE SYSTEM**

Finally, it is very common that some of the previous architectures have to be implemented jointly during the development of a real application, giving rise to the **HYBRID VISION SYSTEM** concept.

2.5. TYPICAL SPECS FOR CAMERA AND VISION SYSTEMS

Cameras and vision systems constitute complex systems where several disciplines of engineering and science are involved in their development, as verified in previous sections. Regarding the development of products, real applications determine the specifications for the main parameters associated to the different sub-blocks which constitute the vision system or camera and the inputs for the different disciplines involved in the product design phase.

Conceptually, a camera is a sub-block in vision systems. Thus, the camera specifications are included and can be derived from the vision system specs. For this reason, this section starts introducing the system level parameters specified by the applications which will be performed by the vision system and finishes with the camera and image sensor parameters to be considered during the design phase.

2.5.1. APPLICATION REQUIREMENTS, SYSTEM SPECIFICATIONS

In order to define the product specifications, a detail study about the tasks which will be performed by the system and conditions under application runs identifies the key parameters to consider during the design phase and determines the requirements, fixing the product characteristics [Hombe]. These characteristics represent the specifications for the vision system sub-blocks.

In general, the vision system characteristics or parameters that play an important role in vision applications can be classified in different types depending on disciplines involved.

MECHANICAL characteristics: In embedded vision systems such as smart cameras, mechanical parameters that must be considered in relation to the case are size, weight, manufacturing materials, lens/image sensor alignment, etc. In general, the vision system is usually composed by several blocks (cameras and processors) depending on application. Each block has associated determined mechanical parameters indicated previously. But also not only the sub-blocks dimensions are important, there are vision systems where it is critical location of component such as cameras and illumination. As for the location of these components, the adjustment is important for installation, operation and maintenance.

<u>**SPEED OR TIMING PERFORMANCES**</u>: The tasks developed by the vision system have to be finished within a specified time. The timing requirements determine the data throughput associated to the data path of vision system (including the data interface of camera) and the maximum operation frequency for digital processors and memory blocks, defining the maximum frame rate of vision system which is expressed in frame per seconds. Therefore, these requirements regarding processing time will influence the choice of the hardware platform and will eventually limit the possibility of using certain algorithms.

Others important timing parameters associated to the image sensor operation are the minimum and maximum exposure times. These specifications are related with the lighting conditions.

<u>**ACCURACY CHARACTERISTICS**</u>: There are four parameters that must be considered in relation with this matter: effective spatial resolution, measurement accuracy, camera resolution and optical performance of camera. The first two parameters are related with the accuracy required in real scenes and the other two corresponds to specs of imager.

The effective spatial resolution or object resolution results from a direct mapping of real-world objects to the image sensor. It can be measured in millimeters per pixel. This spatial resolution depends on the camera resolution and field of view; the mapping is done by the lens.

The measurement accuracy can be considered as the overall accuracy of the system. It is the smallest feature in a captured image that can be measured. Depending on the processing algorithm, the required measurement accuracy can determine different camera specs (spatial resolution and optical performances). This means that both contrast of features as processing algorithms decide when features are measurable.

For instance, if the contrast of small defects is poor, the algorithm might not be able to detect a single pixel defect; then four or five pixels might be necessary. Certainly, the contrast of the feature is determined by the image quality (temporal and spatial noise associated to the image sensor operation). The required image quality defines the optical performances associated to the camera.

Thus, the effective spatial resolution that is necessary in order to achieve determined measurement accuracy depends on the feature contrast (image quality) and the processing algorithms.

<u>**SOFTWARE CHARACTERISTICS**</u>: The vision system has always several parameters and functionalities (exposure time, gain, processing tasks, control tasks, etc.) associated to the different sub-blocks of a vision system, which can be configured by the user through a PC in a general case. This means that a User or Programming Interface must be defined in order to configure and control the vision system. This Programming Interface depends on:

- System architecture type (PC-based system, embedded system, network based systems, etc).
- Parameters to be configured.
- Communication interfaces considered for the vision system, which include a control port (UART, RS232, SPI, etc.) and an image interface (CameraLink, USB, GigE, CoaXpress, etc.).
- Tasks to be performed by different blocks that comprise the system.
- A generic Programming Interface will be used or a custom one will be designed.

When vision systems allow users to define operations and processing algorithms which consist in a set of tasks to be performed by the hardware, the definition of libraries including basic operations and functions is very useful for the applications engineers. This fact provides a significant added value to the product. The user of vision systems can design new algorithms based on these libraries in a easy manner and short time periods. In order to facilitate the creation of new algorithms and applications an Application Development Environment must be designed over the Programming Interface. This ADE software has sense in general purpose vision system which has not designed to perform a specific application or task, where the user cannot modify the operations and processing algorithms implemented by the system.

<u>**FUNCTIONAL CHARACTERISTICS**</u>: The applications to be performed by the vision system define the functions and operations required in each processing level (acquisition, pre-processing, feature extraction, segmentation, description, classification and semantic) in order to implement the corresponding algorithms. Functions related with the control (operation flow) and configuration

of sub-blocks are denoted by control tasks. And operations related with the data processing are denoted by processing tasks.

Given a set of tasks, there are tasks which are implemented optimally at camera level, other tasks which are carried out by the ASIP hardware and tasks developed at system level by the standard digital core, read section 2.4.5.

For instance, there are functionalities which are optimally developed at camera level, such as exposure time control, Region of Interest (RoI) definition, control of conversion gain at analogue domain during image capture, sub-sampling, binning operation, shutter modes (global or rolling), triggering of sensing operation, etc. However, the camera operation requires usually several calibration processes in order to optimize the image quality associated to the captured images. Depending on camera smart level, these calibration processes can be carried out at camera level or system level, working jointly camera, control micro-processor and processing hardware.

ARCHITECTURAL CHARACTERISTICS: Once defined the set of tasks, their physical implementation is closely related to the architecture to develop, that depends on timing and accuracy characteristics particularized to the application belonging to areas such as machine vision, surveillance, autonomous vehicle, intelligent transportation systems, industrial robots, computer-human interaction, etc. For example, a stand-alone embedded system may be very useful in unmanaged vehicles or industrial robots.

From an architectural point of view, the communication ports of system are a key specification. These communication ports comprise the data interfaces and control interfaces. The data throughput associated to the data interface is defined by the image spatial resolution and pixel word length which determine the amount of data to download and the time that the interface has to transmit these data. The control interfaces are responsible for transmitting the configuration and control instructions. In this case, the transmission frequency must be taken into account as design parameter. It is defined by the time that system needs to perform the sequence of operation tasks.

The type of architecture together with frame data, timing and tasks to develop define the maximum data throughput required by each communication interface among the sub-blocks of vision system.

In a vision system, two types of communication interfaces can be defined:

- Internal interfaces which correspond to the communication among the system sub-blocks (cameras, micro-processors and PCs).
- External interfaces which connect the vision system to others systems in a more complex structure.

Considering the internal interfaces, the data buses corresponding to the communication between cameras and processing units which develop the data-intensive tasks have associated the highest throughput, being a critical point during the design phase.

Regarding the digital processors which perform the processing algorithms, there are two types of processing tasks which are described in Figure 38, read section 2.4.1 and section 2.4.5:

- Data-intensive tasks: They deal with a large amount of data (defined by the image resolution). The operating frequency for this processing hardware is determined by the image resolution, pixel word length and the frame rate required by the application. Therefore, for high-speed and high resolution applications the computing power is a critical parameter, these tasks require ASIP hardware operating at high frequencies. In this case, parallel architectures are optimal. Usually, a FPGA is considered in order to implement this processing circuitry in standard vision systems.
- Math-intensive tasks: The amount of data is smaller in this case, because the information to be processed is comprised by the features associated to the regions or points of interest from the application point of view. For this type of tasks, the computer power is not as critical as in the previous case, but the operations carried out in these algorithms require floating-point arithmetic, complex logic in terms of numerous decision points, alternate paths of execution, etc. Thus, high performance micro-processors (DSP or RISC) must be considered in order to implement the math-intensive tasks.

The computing power and the complexity of algorithms and operations determine the architecture to consider for ASIP hardware and type of multi-processor core to use. In the case of high-end applications parallel approaches and multi-core structures may be necessary in the implementation.

In addition to the computing power, there is another important parameter from an architectural point of view, the size of the memory required for the implementation of processing algorithms. There are two types of memories that must be specified to design the vision system:

- Data memory: The design of the data memory plays a crucial role for an efficient implementation of data-intensive tasks. The size associated to the data memories depends on the requirements by the processing algorithms, for example, the need to store one or more images by the system processing.
- Program memory: The math-intensive tasks comprise complex algorithms that imply program codes with a determined number of micro-instructions which define an important portion of size associated to the program memories.

In general, all processing tasks require data memories and program memories. But, depending on tasks type (data-intensive algorithms or math-intensive algorithms), the size specification for a particular type of memory will be more restrictive than other.

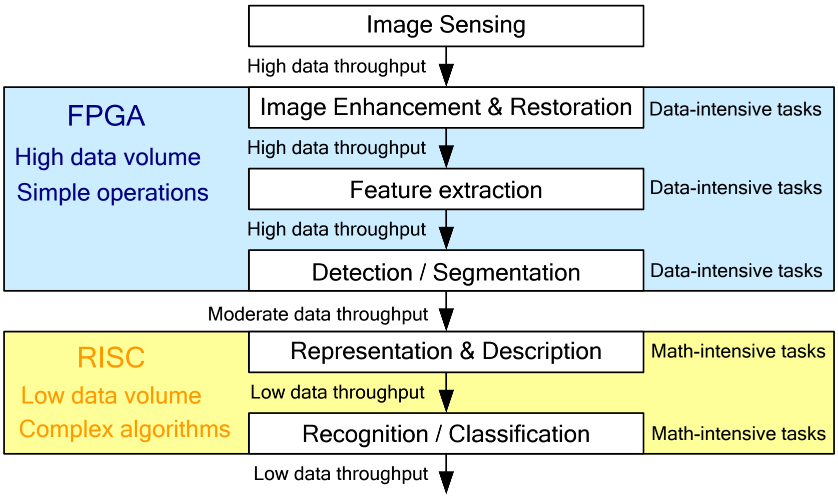


Figure 38. **MULTIPLE-PROCESSORS ARCHITECTURE.**

<u>**OPTICAL PERFORMANCES:**</u> The optical performances are determined by the accuracy characteristics, frame rate and lighting conditions specified by the application to be developed by the vision system. The optical performances are determined by the camera characteristics, which are described in section 2.5.2.

<u>**POWER CONSUMPTION:**</u> In general, the maximum allowable power consumption is an important specification of product. For stand-alone systems and embedded systems is especially critical. In embedded systems, the power consumption implies an increment of case temperature; this fact has a direct effect on the image quality captured by the image sensor. Also, for standalone products the optimization of power consumption budget is important to maximize battery time.

<u>**COST:**</u> From the marketing point of view, the total cost of vision system is essential. This total cost limits the costs assigned to each system sub-block and condition the design process.

In this section, the most common specifications determined by application characteristics have been introduced and the key parameters associated to the design phase of system sub-blocks have been described, indicating the relation between these parameters and the vision system specifications.

2.5.2. CAMERA SPECIFICATIONS

The vision system specifications define the characteristics of cameras to be used. The camera itself can be conceived as a vision system less complex, and therefore the same types of characteristic must be defined for the camera design.

The mechanical characteristics define the size of camera case and the dimensions of optics.

The configuration and control of camera is carried out by the user, usually through a PC. Therefore, a Programming Interface must be defined for this purpose as indicated in the previous section. Several functions and operation modes can be configured through the control interface. For this end, different commands can be sent to the camera by the Programming Interface through the control port. Table describes some of these instructions that must be implemented in the camera control and Programming Interface.

From an architectural point of view, the communication ports (image interface and control interface) must be defined considering the maximum frame rate and the communication with others blocks in a vision system. The camera architecture is determined by the control and processing tasks which will be implemented at level camera. The control camera develops tasks such as those required in the processes of automatic calibration carried out at camera level with the aim of improving the image quality. These tasks imply that the camera control will be usually a micro-processor. On the other hand, the processing circuit performs the data-intensive tasks for image enhancement and it is usually implemented in a FPGA due to the computing power required. The frame rate and image resolution define the computing power and the enhancement processing and control task specify the data and program memories to be implemented in the camera system.

In addition to the mechanical, software and architectural considerations, the image quality and functionality associated to the acquisition stage are fundamentals to guarantee the correct development of application. These aspects affect directly to the physical input of vision system. Therefore, next sections describe in more detail point to be considered during the camera design phase.

2.5.3. OPTICS

The camera lens is a vital part of the photo-capturing process because it molds and shapes the light that comes through it to the image sensor. Features associated to the camera lens are:

- Aperture: It determines the amount of light that camera lens are able to capture. The wider the aperture, the more light can enter. When more light enters, this allows for a faster shutter speed. Aperture diameter is usually specified as the F-number.
- Focal length: The focal length of an optical system represents how strongly the system converges (focus) light (or diverges). It is the distance over which initially collimated rays converge to a focus. A system with a shorter focal length has greater optical power than one with long focal length. Longer focal length or lower optical power is associated with larger magnification of distant objects and a narrower angle of view or field of view. A camera lens system can either have a fixed focal length or a range of focal lengths, which is referred to as a zoom lens. Focal length is measured in millimetres and the higher the number, the more a lens has zooming power. The focal length may also be defined as the minimum practicable distance between image sensor and the optical centre of the lens.
- Field Of View (FOV): The field of view defines the area of the inspection captured on the camera’s image sensor. The FOV parameter and the size of image sensor directly affect the effective image resolution which is a key factor in accuracy.
- Depth Of Field (DOF): Depth of field is the distance between the nearest and farthest objects in a scene that appear acceptably sharp in an image. Although a lens can precisely focus at only one distance at a time, the decrease in sharpness is gradual on each side of the focused distance, so that within the deep of field, the unsharpness is imperceptible. For a given object and camera position, the DOF is determined by the lens aperture diameter. Reducing the aperture diameter (increasing the F-number) increase the DOF; however, it also reduce the amount of light transmitted and increase diffraction, establishing a trade-off. The ability of a camera system to resolve detail is ultimately limited by diffraction. This is because a plane wave incident on a circular lens is diffracted; and light is not focused to a point but forms an Airy disk having a central spot in the focal plane with radius to first null given by the equation Eq.12.
- Modulation Transfer Function (MTF): The Optical Transfer Function (OTF) of an imaging system is a measurement of real resolution (image sharpness) that the system is capable

of. In the most common applications for cameras and vision systems, the relevant information is given by the Modulation Transfer Function (the magnitude of OTF). While image sensor resolution, commonly used as reference to camera systems, describes only the number of pixels in a image, and hence the maximum potential to show fine detail, the MTF parameter describes the ability of adjacent pixels to change from black to white in response to patterns of varying spatial frequency, and hence the real capability to show fine detail. Every component within a vision system has associated a MTF parameter and, as a result, contributes to the overall MTF of the vision system. This includes the imaging lens, image sensor, processing algorithms, etc. The resulting MTF is the product of all MTF curves of system components.

- Aberrations: Optical aberrations are deviations of images obtained by an optical system from the predictions of paraxial optics. There are two main classes: Monochromatic aberrations caused by the geometry of lens and chromatic aberrations caused by dispersion (variation of lens refractive index with wavelength).
- Size
- Cost

2.5.4. Camera Functionality

Generally, cameras that can be found in the market have additional functions to the image capturing. The camera functions can be classified in two groups: functions to improve the image quality (they are denoted by processing functions) and functions which configure the sensing operation (they are denoted by sensing functions).

Among the processing functions, the correction algorithms play an important role in order to improve the image quality compensating errors related with the image sensor operation. Errors which can be compensated are:

- Vertical Fixed Pattern Noise (VFPN)
- Horizontal Fixed Patter Noise (HFPN)
- Non-linearity associated to the responsivity curve of image sensor
- Shading
- Dark Signal Non-Uniformity (DSNU)
- Photo Response Non-Uniformity (PRNU)

The correction coefficients are calculated by a calibration processes which require special configurations for the image sensor, download images and calculate the correction coefficients following a determined algorithm or procedure. Usually, camera systems carry out automatically the multiple tasks which compose the calibration process. These control tasks are managed by the camera control block that consists in a micro-processor. The number of control tasks present in the calibration processes must be considered in the sizing of program memories.

The numeric operations carried out by the correction circuitry during the image readout process have associated a high operating frequency because of high data path throughput. This operating frequency is an important issue for high resolution and high frame rate applications, being necessary to implement the operations with specific digital circuitry (implementation based on FPGA or custom on-chip design). The data storage requirements by the correction algorithms and calibration procedure must be considered during the definition of data memory size.

In addition to the calibration/correction algorithms, there are others processing functions that implement enhancement algorithms such as color balance, Point-to-Point transformations (for example, to equalize the histograms) or some filters to reduce the noise contributions.

Also, the throughput defined by the spatial resolution, pixel word length and frame rate associated to the camera operation can be larger than that allowed by the bandwidth of transmission medium used in communication interface. Therefore, compression algorithms must be implemented in addition to the processing to improve the visual image quality. In image transmission, interest lies in techniques that achieve maximum reduction in the quantity of data to be transmitted, subject to the constraint that a reasonable amount of fidelity be preserved. This means that resulting images are acceptable for visual or machine analysis.

Associated to the sensing and reading image sensor operation, the camera has configurations and functionalities managed by control system. Among these, it stands out:

- Exposure time control: Useful functionalities for vision applications are, for example, changing the exposure time between frames without stopping system operation (this allows the implementation of High Dynamic Range algorithms) or applying different exposure times depending on the colour assign to the pixel.
- Regions of Interest: There are applications where a region with a determined feature must be downloaded for its processing. To define and readout regions smaller than full resolution images allows increasing the frame rate during the reading of these windows. The capacity to define these RoIs frame-to-frame allows vision systems identify regions of interest over a full frame by simple processing algorithms on later to make detailed analysis of RoIs.
- Sub-sampling operation: With this functionality, the spatial sampling frequency of capture operation can be configured. The reduction of effective resolution makes possible to increase the frame rate.
- Binning operation: This feature improves the Signal to Noise Ratio at the expense of a loss in the sensor resolution. This less resolution can be exploited by the system to increase the maximum frame rate.
- Offset and Gain in the analogue domain during the reading process. This function allows modifying the overall system gain (expressed in voltage per electrons). In colour applications, it is very interesting the possibility of applying different analogue gains depending on the colour channel.

2.5.5. CAMERA IMAGE QUALITY

Although the image quality can be improved significantly by digital processing, there are fundamental parameters that cannot be modified by processing. Also, the image processing has always pros and cons. When some characteristic is improved, there is some drawback and other feature is affected. Hence, an optimum initial state for the design of camera system is an image sensor with the best possible performances. This starting point could imply a simplification of digital processing circuit and control.

The camera image quality is given mainly by the image sensor performances. These performances depend on photo-diode characteristics and readout circuit features. In this section, the main image sensor specs will be described, [EMVA].

The image sensor characteristics can be classified into two parameter types:

- Parameters related with the image sensor responsivity.
- Parameters related with the noise present in a captured image, which define the image quality.

2.5.5.1. RESPONSIVITY PARAMETERS

In relation with the pixel, there are four parameters that contribute to the responsivity curves:

- The maximum number of electrons that can be collected by the photo-diode, which is denoted by Full Well Capacity (FWC).
- The product of pixel Fill Factor by the Quantum Efficiency (FF x QE), both parameters described in section 2.1.1 and section 2.2.2 respectively. This parameter defines the number of electrons (n_e) generated from photons incident on pixel area (n_{ph}):

$$FFxQE = \frac{n_e}{n_{ph}} \qquad \textbf{Eq. 28}$$

And the number of photons incident on pixel area is given by the equation:

$$n_{ph} = \frac{E \cdot A_{pixel}}{h \cdot \frac{\lambda}{c}} \cdot T_{\exp} \qquad \text{Eq. 29}$$

E is the irradiance on sensor surface [W/m^2], A_{pixel} is the pixel area, $T_{\exp}$ the exposure time, λ the wavelength, h the Planck constant and c the light speed.

- Pixel Conversion Gain (CG) which is the electron-voltage conversion factor. This parameter has two contributions; one due to the capacitance associated to the integration node of pixel (denoted by Floating Diffusion) and other given by the analogue gain associated to the output Source Follower (G_{SF}):

$$CG = \frac{q}{C_{FD}} \cdot G_{SF} \qquad \text{Eq. 30}$$

- Dark current associated to the photodiode operation introduces a global offset in sensed images, which depends on the exposure time. This offset implies an effective loss of sensor output, thereby reducing the intra-frame dynamic range, defined in section 2.5.5.2.

Related with the readout channel, there are two parameters that influence the responsivity characteristic:

- The analogue gain implemented in this analogue circuitry G_{read}, which together with pixel Conversion Gain defines the overall system gain K:

$$K = CG \cdot G_{read} = \frac{q}{C_{FD}} \cdot G_{SF} \cdot G_{read} \qquad \text{Eq. 31}$$

- Output range of complete readout channel. It can be expressed in electrons at the input through the overall system gain parameter. Thus, the effective FWC (saturation) is given by the most restrictive value between the photo-diode FWC and readout FWC. During the design phase of image sensor, these parameters must be compatible.

Mathematically, the responsivity (R=[V/(W/m^2]) is the slope associated to the curve resulting from the representation of output voltage or corresponding Digital Number versus irradiance (W/m^2) incident on sensor surface. Thereby, the responsivity is related with the FFxQE and overall system gain (K) parameters:

$$R(\lambda) = K \cdot FFxQE(\lambda) \cdot \frac{A_{pixel}}{h \cdot \frac{\lambda}{c}} \cdot T_{\exp} \qquad \text{Eq. 32}$$

The effective FWC and overall system gain are related for a given output range of readout channel. A larger system gain K means that the image sensor reaches the same output voltage level or digital number with less photo-generated electrons and hence less light intensity. The image sensor can see in lower light conditions. But, the increment of system gain implies that the output saturation level of readout channel is reached for a smaller number of photo-generated electrons, and thus, the effective FWC is less. As it will be explained in following point, Full Well Capacity is related with the maximum Signal to Noise Ratio. This parameter is a measurement of image quality under lighting conditions. Therefore, there is a trade-off between responsivity and full well capacity of image sensor.

Another important characteristic of responsivity curve is the non-linearity. There are several sources of non-linearity due to the pixel and the readout channel. Usually, the capacitance associated to the integration node C_{FD} changes with voltage level of this node, due to the diffusion areas of devices. Hence, the conversion factor is not constant for the whole signal range, performing a non-linear photon-voltage conversion. Another source of non-linearity is the Input-Output characteristic of pixel Source Follower (SF) used in general to read photo-generated signals. Also, a possible non-linear behavior of readout channel is mapped directly in the responsivity curve. Normally, during the design phase, the readout channel must be designed such that its non-linear characteristic will be negligible respect to the pixel SF non-linearity.

For monochrome images, the effects generated by a non-linear responsivity curve are not directly observable in capture images, when this non-linearity is an Integral Non Linearity (INL) type, as shown in Figure 39 where a quadratic compressive non-linearity error has been considered.

In the case that the non-linearity error is a Differential Non Linearity (DNL), artifacts can be observed in sensed images, mainly in regions of an image where a gradual gradient in gray levels is present, because some gray levels disappear due to the Differential Non Linearity, as described in Figure 40, where digital codes have been lost due to the AD conversion process. Presence of DNL errors in the ADC converters of readout channels may be detected by analyzing the histogram of images similar to image shown in Figure 41. For instance, the error described in Figure 40 provokes missing codes in the histogram.

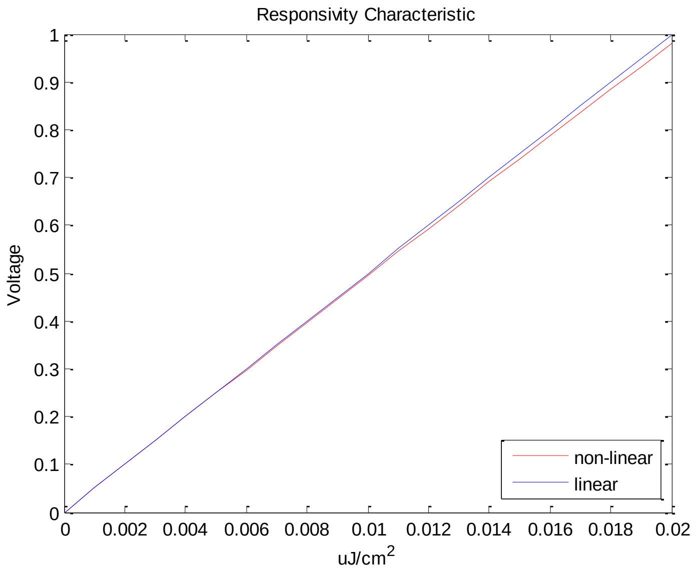


Figure 39. INTEGRAL NON LINEARITY IN RESPONSIVITY CURVES.

Figure 41 represents the artifact generated by the DNL error depicted in Figure 40. In a real scene consisting of a continuous degradation of light intensity, the DNL error generates an artificial edge. This issue is unacceptable for further processing levels in vision applications.

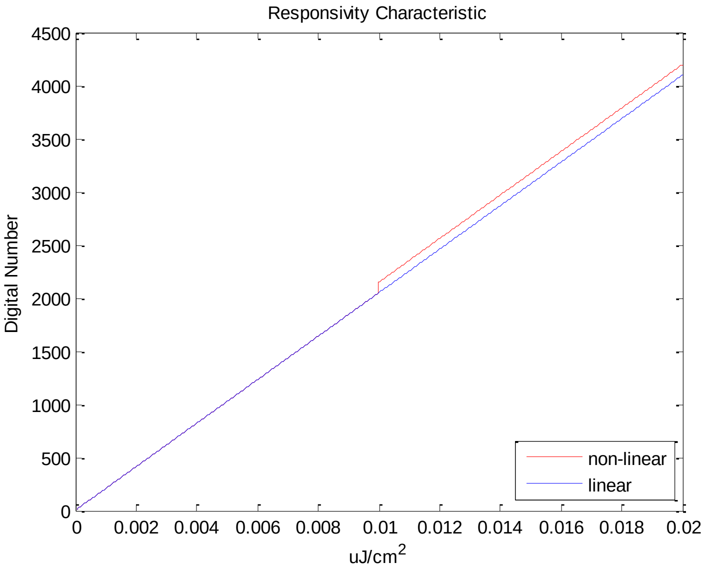


Figure 40. DIFFERENTIAL NON LINEARITY IN RESPONSIVITY CURVES.

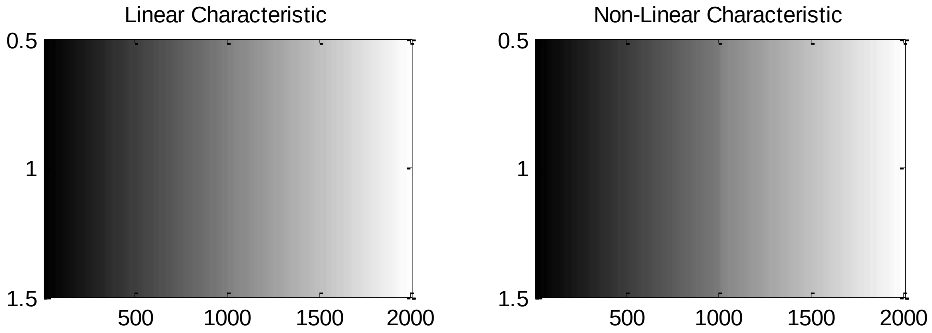


Figure 41. ARTIFACT DUE TO DNL ERROR.

For color images, both types of non-linearity generate artifacts. In this case, any type of non-linearity in the responsivity curve affects to the color balance results. The color filter array deposited over the pixel array to implement the color image sensor affects to the responsivity associated to each type of color pixel (Red, Green or Blue). Each color filter has a different transmittance in function of wave length. This fact implies that each responsivity curve associated to each color has a different overall system gain. In case of a linear responsivity, the color balance has to apply different factors to obtain the real color for a particular pixel. In the presence of non-linearity errors, the correct color value cannot be obtained by multiplicative color constant factors. In this case, the multiplicative color factors depend on intensity level due to the non-linearity errors, complicating the implementation of color balance algorithm.

As mentioned in previous paragraphs, the overall system gain has components associated to the pixel and components associated to the readout channel circuitry. Any deviation respect to the nominal value of these components due to the mismatch of physical devices provokes spatial noise. In this case, the spatial noise is proportional to the signal (Photo Response Non Uniformity). This spatial noise is a parameter related with the image quality, which will be explained in next section 2.5.5.2.

In a similar way, the mismatch of non-linearity error of responsivity curve generates spatial noise in sensed images. The spatial noise is not proportional to the photo-generated signal in this case. Thus, the calibration-correction algorithm will be more complex. Usually, a linearization process is required for this correction.

2.5.5.2. IMAGE QUALITY PARAMETERS

In section 3.3.4 of Chapter 3, a detailed description of non-idealities associated to the behavior of pixels is performed. In this case, the study is focused on the particular pixel of Q-eye system, but their conclusions can be extrapolated to the standard behavior of photosensors without loss of generality. The previous mentioned study described the sources of different errors or deviations respect to the ideal behavior, mainly the errors associated to the pixel (photo-diode, reset transistor and output Source Follower). In this section, the subject of study will be focused on their effects in captured image at camera system level and will present the parameters which characterize these noises.

From the image quality point of view, the noise of a captured image has two components:

TEMPORAL NOISE, which is composed by the contributions of pixel and readout channel. It can be lumped into three additive components:

- Shot noise due to the photo-current i_{ph} and dark-current i_{dc} generated by the photodiode, whose Power Spectral Density (PSD) ,expressed in e^2, is given by the equation:

$$\sigma_{shot}^2 = \frac{1}{q} \cdot \left(i_{ph} + i_{dc}\right) \cdot T_{\text{int}} = n_{ph} + n_{dc} \qquad \textbf{Eq. 33}$$

Where T_{int} is the integration or exposure time and n_{ph} , n_{dc} are the electrons photo-generated and electrons generated in dark conditions respectively.

- Reset noise associated to the thermal noise of reset transistor included in the pixel. Its PSD is denoted by σ^2_{reset} (expressed in e^2).
- The readout noise which is composed by the electronic noise of pixel source follower and readout channel. The corresponding PSD is denoted by σ^2_{read} (expressed in e^2).

The total temporal noise power results from the addition of powers associated to each contribution:

$$\sigma^2_{temp} = \sigma^2_{shot} + \sigma^2_{reset} + \sigma^2_{read} \qquad \textbf{Eq. 34}$$

The temporal noise in dark conditions represents the smallest detectable input signal:

$$\sigma^2_{temp-dark} \approx \sigma^2_{reset} + \sigma^2_{read} \qquad \textbf{Eq. 35}$$

Where the shot noise contribution of dark-current has been considered negligible.

This temporal noise in dark conditions together with the effective full well capacity (n$_{sat}$), which defines the largest non-saturating input signal, determines the Dynamic Range (DR) parameter.

The dynamic range quantifies the ability of an image sensor to adequately represent both high lights and dark shadows in a scene. Mathematically it is defined by the equation:

$$DR = 20 \cdot \log_{10} \frac{n_{sat}}{\sigma_{temp-dark}} \qquad \textbf{Eq. 36}$$

Another important parameter to determine the image sensor quality is the Signal to Noise Ratio (SNR), which is defined as the ratio of the input signal power to the average input referred noise power:

$$SNR = 10 \cdot \log_{10} \frac{n^2_{ph}}{\left(\sigma^2_{shot} + \sigma^2_{reset} + \sigma^2_{read}\right)} \qquad \textbf{Eq. 37}$$

The SNR parameter is a measurement of quality. As explained in section 2.5.1, the measurement accuracy required by applications defines the maximum noise level which allows developing correctly the processing algorithm without errors. This system specification determines the SNR necessary in order to implement a particular application.

The SNR increases with the input signal (n_{ph}), first at 20dB per decade for small input since read noise dominates, then at 10dB per decade when shot noise due to the photo-detector current dominates.

The maximum SNR achievable by the image sensor is determined by the effective full well capacity (n_{sat}):

$$SNR_{\max} = 10 \cdot \log_{10} \frac{n^2_{sat}}{(n_{sat} + n_{dc}) + \sigma^2_{reset} + \sigma^2_{read}} \approx 10 \cdot \log_{10} n_{sat} \qquad \textbf{Eq. 38}$$

Therefore, the effective full well capacity (n_{sat}) is involved in the definition of dynamic range and the maximum quality achievable by the sensor. For a given output range of readout channel, there is a tradeoff between responsivity and effective full well capacity, what means that there is a tradeoff between responsivity and achievable image quality (DR and $SNR_{\max}$).

Regarding this trade-off between responsivity and image quality, the pixel conversion gain is other parameter that contributes to this trade-off. The PSD of reset noise (read section 3.3.4.2 of Chapter 3) is given by:

$$\sigma^2_{reset} = \frac{k \cdot T}{C_{FD}} \qquad (V^2)$$

$$\sigma_{reset}^{2} = \frac{k \cdot T}{q^{2}} \cdot C_{FD} \qquad \left(e^{2}\right)$$ **Eq. 39**

Then, considering the power spectral density of reset noise expressed in electrons in the floating diffusion (FD) node, the reset noise is proportional to the capacitance associated with this FD node. In order to reduce the noise contribution associated to the reset transistor, the FD capacitance should be incremented. But, this increment means a larger conversion gain (Eq.30). An increment in the conversion gain implies a reduction in the full well capacity for a given output range of readout channel.

Depending on the readout channel architecture implemented in the camera system, the temporal noise and settling errors associated to the analogue references used in the readout process could generate artifacts in capture images such as line noise or flickering between frames in a video sequence. For instance, in a column ADC approach for Digital Image Sensor, the noise in a global reference common to all converters can generate a line noise as shown in Figure 42.

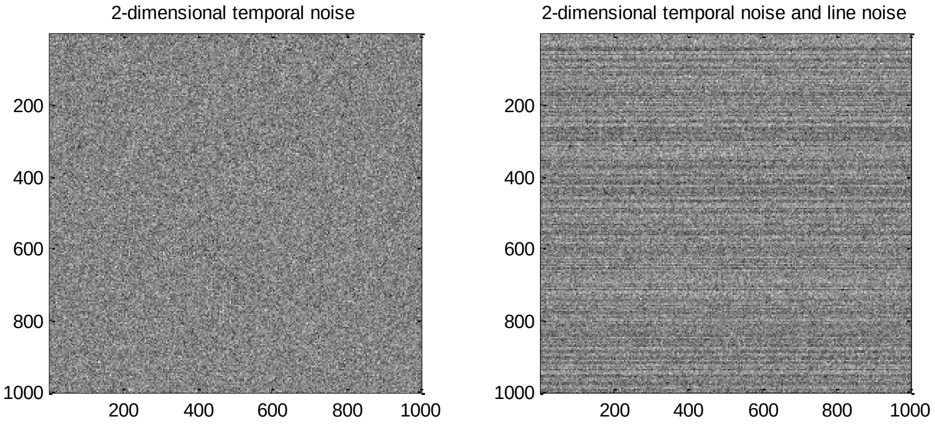


Figure 42. **TEMPORAL LINE NOISE EFFECT.**

If this temporal line noise is an offset, then it can be compensated when Optical Black columns are implemented in the pixel array. The corresponding offset error is calculated averaging the Optical Black columns in order to subtract the mean value from the corresponding active pixels.

During the design phase the PSD of this line noise must be considered and compared with the expected two-dimensional temporal noise.

The DR and SNR features can be improved implementing a true Correlated Double Sampling. During this operation, the output of pixel is sampled twice: once after reset and the second time with the photo-generated signal.

The sample resulting from the first sampling phase is given by:

$$n_1 = n'_{reset} + n'_{read}$$ **Eq. 40**

The sample obtained in the second sampling phase is:

$$n_2 = n_{ph} + n_{dc} + n_{shot} + n_{reset} + n_{read}$$ **Eq. 41**

Thus, the difference of samples results:

$$n_2 - n_1 = n_{ph} + n_{dc} + n_{shot} + \left(n_{reset} - n'_{reset}\right) + \left(n_{read} - n'_{read}\right)$$ **Eq. 42**

In the case of true CDS ($n_{reset} = n'_{reset}$), the PSD associated to the operation is given by the equation:

$$\sigma_{n_2 - n_1}^{2} = \sigma_{shot}^{2} + 2 \cdot \sigma_{read}^{2}$$ **Eq. 43**

When true CDS operation cannot be carried out, for example, a 4T pixel topology operating in global shutter, the resulting temporal noise is:

$$\sigma^2_{n_2-n_1} = \sigma^2_{shot} + 2\cdot\sigma^2_{reset} + 2\cdot\sigma^2_{read} \qquad \textbf{Eq. 44}$$

The spatial noise has not been included in the definition of DR and SNR because the spatial noise is unlikely that it is corrected. So, the fundamental limits for DR and SNR are given by the temporal noise.

In the case that the spatial noise or Fixed Pattern Noise is considered, the DR and SNR will be given by the equations:

$$DR = 20\cdot\log_{10}\frac{n_{sat}}{\sigma_{temp-dark} + \sigma_{FPN}} \qquad \textbf{Eq. 45}$$

$$SNR = 10\cdot\log_{10}\frac{n^2_{ph}}{\left(\sigma^2_{shot} + \sigma^2_{reset} + \sigma^2_{read} + \sigma_{FPN}\right)} \qquad \textbf{Eq. 46}$$

<u>FIXED PATTERN NOISE</u> (FPN) characterizes the spatial variation in pixel outputs under uniform illumination conditions due to device and interconnect mismatches, which provokes different transfer functions or responsivities for each pixel. The errors associated to the static characteristic (responsivity) have a global component equal for all pixels and readout channels and a random component due to this mismatch of device behavior. The last one generates FPN on the sensed image, because under uniform illumination the resulting image presents spatial variations. This spatial noise remains superimposed on the image in a video sequence.

Mainly there are two FPN components:

- Offset (Dark Signal Non Uniformity, DSNU): Generated by the random component of photodiode dark current, the different offset contributions associated to the readout channels when more than one channel is implemented in the image sensor for download process, mismatch of settling errors during the pixel reading, etc.

  Under dark conditions, the captured image presents spatial noise due to this mismatch offset associated to the pixel operation and readout channel. The offset error resulting from pixel operation is distributed randomly over the image, and this is the component which is denoted by DSNU. Usually, the offset error associated to the readout channels generates structures or artefacts in the image, which are unacceptable for further processing. These structures depend on the readout architecture. For instance, for only one readout channel, the error offset of circuitry is a global contribution to the image. But, when two or more channels are used to download an image Vertical Fixed Patter Noise (VFPN) appears due to the error offset associated to each channel. This is a typical parameter for column ADC approach in Digital Image Sensors. There are readout architectures which comprises several channels per column to download several rows in parallel. This is a usual topology for high resolution and high frame rate image sensors. In this case, a Horizontal Fixed Pattern Noise (HFPN) could be observed in capture images.

  Therefore, the spatial noise in dark conditions is given by the equation:

$$\sigma^2_{spatial\_offset} = \sigma^2_{DSNU} + \sigma^2_{VFPN\_offset} + \sigma^2_{HFPN\_offset} \qquad \textbf{Eq. 47}$$

  The VFPN and HFPN are unacceptable in vision applications, in case that the image sensor has this type of spatial noise calibration/correction procedures must be implemented at camera level. The correction coefficients will be additive values to compensate channels offsets.

  Under dark conditions, if a unique readout channel is considered, only the random offset contribution associated to the pixel array operation can be observed in captured images, Figure 43 shows an example.

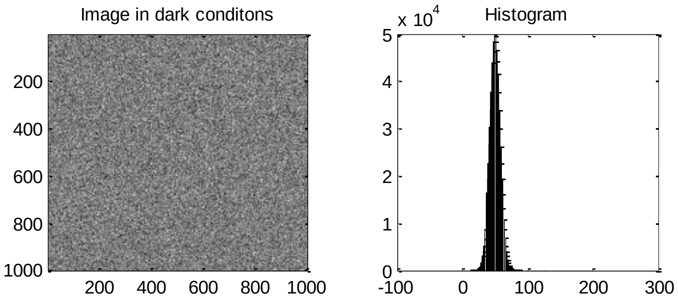


Figure 43. **DSNU EFFECT.**

Calculating the histogram, the captured image has a global offset due to the systematic component of dark current and readout circuitry offset. And the random distribution is provoked by the mismatch of dark current among pixels.

Assuming a column ADC approach for readout architecture, the offset contribution of column circuitry introduced VFPN in addition to the random offset associated to the pixel array, see Figure 44.

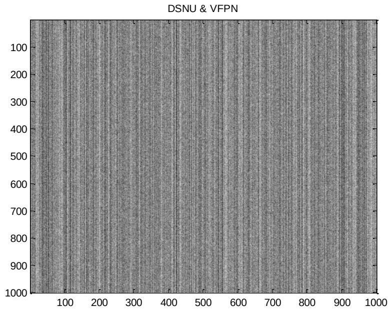


Figure 44. **VERTICAL FPN EFFECT.**

Depending on the VFPN magnitude, the vertical line can be detected or not. In Figure 45, several PSDs for VFPN are shown considering always the same 2-dimensional random spatial noise (DSNU).

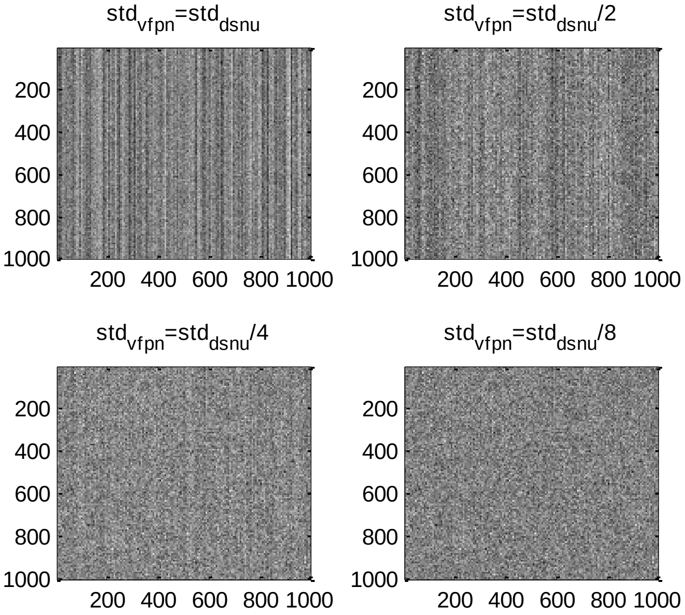


Figure 45. **PSD OF VFPN VERSUS PSD OF DSNU.**

- Gain (Photo Response Non Uniformity, PRNU): Generated by the mismatch of pixel conversion gain, which generates a 2-dimensional random spatial noise, the mismatch of analogue gain applied by the analogue circuits of readout channels, when two o more channels are implemented, the settling errors during the reading process, etc.

  Usually, the random noise introduced by the pixel array is denoted by PRNU. The gain error introduced by the readout channels has a systematic component which introduces a global effect and a random component due to the device mismatch which generates VFPN and HFPN in a similar way than offset component. But in this case, the magnitude of gain error depends on input signal, being null under dark conditions and maximum when maximum signal is reached.

  For a particular signal level the spatial noise due to the gain errors is given by the expression:

  $$\sigma^2_{spatial\_gain} = \sigma^2_{DSNU} + \sigma^2_{VFPN\_gain} + \sigma^2_{HFPN\_gain} \quad \textbf{Eq. 48}$$

  Therefore, the total spatial noise power is composed by the two error contributions:

  $$\sigma^2_{FPN} = \sigma^2_{spatial\_offset} + \sigma^2_{spatial\_gain} \quad \textbf{Eq. 49}$$

  The gain VFPN and HFPN are unacceptable for vision applications too. Then, calibration/correction algorithms must be developed when the image sensor presents this type of spatial errors. Considering a linear transfer function of image sensor, the correction coefficients consist in multiplicative factors.

In addition to the temporal and spatial noises, there are other important parameters which determined the image quality required by a particular application. One of these parameters is the Modulation Transfer Function. The MTF parameter depends on several factors related with the pixel, such as pixel size, metallization, presence or absence of micro-lens, etc. The meaning of this parameter has been introduced in section 2.5.3 [Naka08].

Other important characteristic associated to the image sensor is the parameter denoted by **SHUTTER EFFICIENCY**, mainly when the image sensor operates in Global Shutter. This parameter is related with the fact that the diffusions associated to the integration node (Floating Diffusion, FD) constitute a parasitic photo-detector. In Global Shutter, the photo-generated carriers are stored in the FD capacitance during the frame reading process. In a row by row download scheme, this means that the charge stored in the last read rows is more exposed to the operation of parasitic photodiode than the first downloaded rows. Figure 46 depicts the degradation suffered by a uniform image due to this phenomenon.

The degradation is caused both by the dark current associated with this parasitic photodiode as its sensing operation collecting photo-generated charge into the FD capacitance. This latter component is predominant. Thus, the Shutter Efficiency parameter is defined as the ratio between the photodiode responsivity and the Floating Diffusion responsivity. With the aim of increasing the shutter efficiency, several strategies are considered at architectural level (introduction of analogue memories in pixel) and physical level (doping profiles, metallization, micro-lens, etc.).

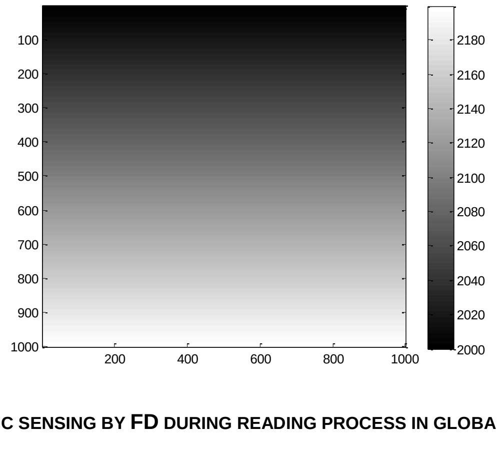


Figure 46. **PARASITIC SENSING BY FD DURING READING PROCESS IN GLOBAL SHUTTER.**

Finally, there is a phenomenon which degrades image quality of pictures acquired in consecutive captures. It is called image lag. Lag occurs when residual charges remains in the pixel during the next consecutive exposed frame and may corrupt the image (especially images of bright objects in low-light scenes). Residual charges may come due an incomplete transfer in the photodiode or dynamic errors during the reset phase of integration node [Naka08].

To conclude this section, Table.1 summarizes the main parameters to consider when the quality of a camera or image sensor must be determined.

Parameters			Units
Array resolution			pixel x pixel
Pixel size			µm
Conversion bits (word length)			
Responsivity		Full Well Capacity	e
		Pixel Conversion Gain	µV/e
		Readout Analogue Gain	
		Fill Factor x Quantum Efficiency	%
		Responsivity	V/(µJ/cm^2)
		Non-Linearity	% Full Scale
Noise	Dark conditions	DR	dB
		Temporal line noise	%FS
		DSNU	%FS
		VFPN$_{offset}$	%FS
		HFPN$_{offset}$	%FS
	With illumination	SNR	dB
		PRNU	%FS
		VFPN$_{gain}$	%FS
		HFPN$_{gain}$	%FS
Dark Current			e/sec
Modulation Transfer Function			
Shutter Efficiency			
Image Lag			

Table.1. Camera image quality.

2.6. PREDECESSORS OF THE Q-EYE

In this section, several Analog Parallel Array Processors (APAP) will be introduced which have been designed in order to carry out image processing efficiently with outstanding computing power figures when compared to conventional serial digital signal processors.

The neural networks (NN) are parallel processing architectures. These neural networks are composed by a set of processing units interconnected, called neuron. Each unit processes the weighted inputs which come from the other neurons belong to the network. Also, the output unit is transmitted to the rest of network being the input to others neurons. Different types of neural networks can be defined depending on type of processing units implemented in the neural network and the interconnection among units to generate the processing network: feed-forward neural network, recurrent neural network, fully-connected neural network, partially connected neural network, etc.

An interesting recurrent neural network (RNN) from an image processing point of view is the Cellular Neural Network (CNN) defined by professors L.O. Chua and L. Yang in 1988 [Chua88]. The CNN is a new class of analogue dynamic array processors, where the elementary processing units are locally connected. This means that they interact directly only with those processing units located within a finite local neighborhood. The interest of CNNs is based on the fact that many functions related to the processing of multi-dimensional analogue signals, such as images, can be formulated as operations between each individual (one-dimensional) signal value (pixel), and the values placed within a local receptive field (neighborhood), directly mappable onto CNNs.

2.6.1. THE CNN PARADIGM

The Cellular Neural Network [Chua93] is an array of non-linear dynamic processing units, called cells, which occupy the nodes of a n-dimensional grid. These cells satisfy two properties:

interactions between them are restricted to a local scope and state variables are continuous-valued signals.

Due to the images are two-dimensional variables, from now, a two-dimensional array composed by MxN processing cells will be considered, without loss of generality because definitions and results are easily extended to higher dimensional networks.

A cell of processing array will be denoted by $C(i,j)$, where $i \in [1,M]$ represents the row and $j \in [1,N]$ the column.

The $C(i,j)$ neighborhood of r order is defined by the following equation:

$$N_r(i,j)=\{C(k,l)/\max(|k-i|,|l-j|)\le r, 1\le k\le M \quad 1\le l\le N\}$$ **Eq. 50**

where r is a positive integer.

Figure 47 represents a first order neighborhood, where each cell of the array interacts directly only to those cells of the grid located within a finite radius $r=1$.

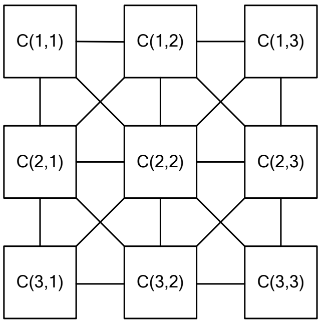


Figure 47. FIRST ORDER NEIGHBORHOOD.

The topology of CNN becomes univocally defined by the grid shape (rectangular, hexagonal, etc.), the neighborhood order and a set of boundary conditions which are defined by the set of cells situated less than r cells away from the border of the network. Without loss of generality, the shape of grid considered in the study has been rectangular.

Each cell has associated three variables:

- The state variable, denoted by $x_{ij}(t)$, represents the cell state and has information about the cell energy.
- The output variable, denoted by $y_{ij}(t)$, is the cell output. It is obtained from the state variable through a non-linear transformation f :

$$y_{ij}(t)=f\left[x_{ij}(t)\right]$$ **Eq. 51**

Figure 48 describes several output functions.

- The input variable, denoted by $u_{ij}(t)$, represents the external excitation.

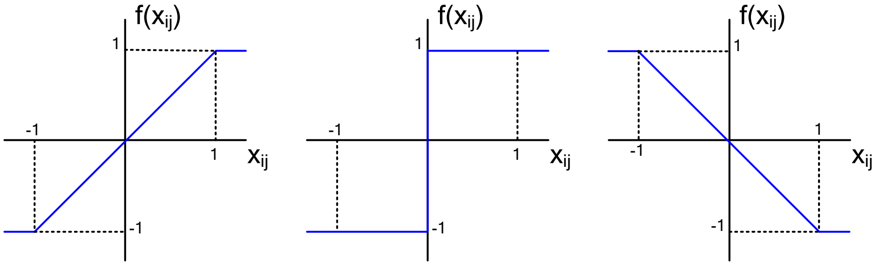


Figure 48. OUTPUT FUNCTION.

For a given initial state of cells represented by a vector of initial state $\overrightarrow{x_0} = \{x_{ij}(0)\}$, an external excitation described by an input vector $\vec{u} = \{u_{ij}\}$ and particular boundary conditions, the processing array evolves following a determined dynamic and the outcome is given by an output vector $\vec{y} = \{y_{ij}\}$. These vectors for a two-dimensional processing array adopt the form of matrix.

The dynamic of CNN cell is described mathematically by the ordinary differential equation in continuous time:

$$\begin{aligned}\tau \cdot \frac{d}{dt} x_{ij}(t) = -g\left[x_{ij}(t)\right] + z_{ij} + \sum_{k=-r}^{r} \sum_{l=-r}^{r} \Big\{ & A_{ij,kl}\left[y_{((i+k),(j+l))}\right] + \\ & + B_{ij,kl}\left[u_{((i+k),(j+l))}\right] + C_{ij,kl}\left[x_{((i+k),(j+l))}\right] + \\ & + D_{ij,kl}\left[u_{((i+k),(j+l))}, x_{((i+k),(j+l))}, y_{((i+k),(j+l))}\right] \Big\}\end{aligned} \qquad \textbf{Eq. 52}$$

τ is the time constant associated to the processing array, g is the dissipation function (Figure 49) represents several examples of dissipation functions), z_{ij} is an offset term for each processing cell and $A_{ij,kl}$, $B_{ij,kl}$, $C_{ij,kl}$ and $D_{ij,kl}$ are the interaction operators. These interaction operators are non-linear in the more general case, but many processing tasks require only linear interaction among cells, thus receiving the name of interconnection weights. Moreover, $C_{ij,kl}$ and $D_{ij,kl}$ operators are null for many applications. In most practical cases, the nonlinearity appears only at the scaling factor of an otherwise linear transformation.

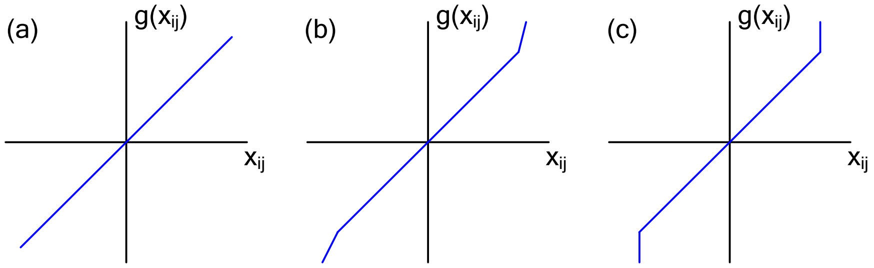


Figure 49. DISIPATION FUNCTIONS.

The general model of CNN has been presented. Several particular cases based on the general model have been developed in order to implement many image processing algorithms.

2.6.2. LEON O. CHUA AND LIN YANG MODEL

The original model of CNN presented by Chua and Yang has a constant neighborhood order (neighborhood radius r is constant) and the interconnection weights do not depend on cell position in the array. These characteristics provide the property of translational invariance. The local interconnection, regularity and translational invariance are features very convenient for a VLSI implementation.

The dissipation function considered in the Chua-Yang model is linear, option (a) in Figure 49.

The interconnection operators $A_{ij,kl}$ and $B_{ij,kl}$ are linear, while the operators $C_{ij,kl}$ and $D_{ij,kl}$ are nulls.

Under these considerations the differential equation which describes the behavior of processing cell is given by:

$$\tau \cdot \frac{d}{dt} x_{ij}(t) = -x_{ij}(t) + z_{ij} + \sum_{k=-r}^{r} \sum_{l=-r}^{r} \left[a_{kl} \cdot y_{((i+k),(l+l))} + b_{kl} \cdot u_{((i+k),(l+l))} \right] \qquad \textbf{Eq. 53}$$

This equation can be modeled by an equivalent circuit which is represented in Figure 50.

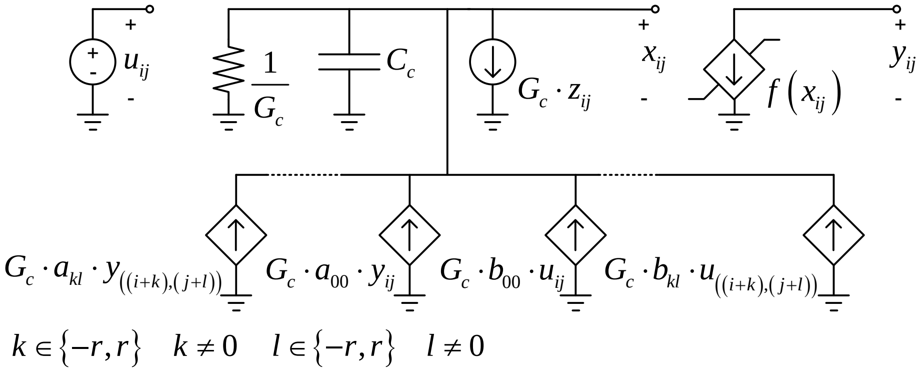

+
-
u_{ij}
$\frac{1}{G_c}$
C_c
$G_c \cdot z_{ij}$
+
x_{ij}
-
$f(x_{ij})$
+
y_{ij}
-
$G_c \cdot a_{kl} \cdot y_{((i+k),(j+l))}$
$G_c \cdot a_{00} \cdot y_{ij}$
$G_c \cdot b_{00} \cdot u_{ij}$
$G_c \cdot b_{kl} \cdot u_{((i+k),(j+l))}$
$k \in \{-r, r\} \quad k \neq 0 \quad l \in \{-r, r\} \quad l \neq 0$

Figure 50. **CHUA-YANG CNN CIRCUITAL MODEL.**

The output nonlinear function is described by the equation:

$$y_{ij} = f(x_{ij}) = \frac{1}{2} \cdot \left(|x_{ij} + 1| - |x_{ij} - 1| \right) \qquad \textbf{Eq. 54}$$

This function corresponds to the option (a) in Figure 48.

As operators are linear and cell neighborhood has a constant radius, the interconnection operators a_{kl} and b_{kl} can be represented by matrixes. In the case of first order neighborhood these matrixes will be the 3x3 order:

$$A = \begin{pmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & a_{0,0} & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{pmatrix} \quad B = \begin{pmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{pmatrix} \qquad \textbf{Eq. 55}$$

The matrix A is called “feedback template” and it represents the weights associated to the output contributions of cell neighborhood. The matrix B is called “control template” or “feedforward template”, it represents the weights associated with the external excitation of cells belong to the processing vicinity.

2.6.3. FULL SIGNAL RANGE (FSR) MODEL

Another CNN model interesting from a point of view of VLSI implementation is the Full Signal Range model [Espe94b]. This model is characterized by the fact that the state variable x_{ij} is bounded by a nonlinear dissipation term $g(x_{ij})$ given by the equation:

$$g(x_{ij}) = \lim_{m \to \infty} \begin{cases} m(x_{ij} - 1) & if \quad x_{ij} > 1 \\ x_{ij} & if \quad |x_{ij}| \leq 1 \\ -m(x_{ij} + 1) & if \quad x_{ij} < -1 \end{cases} \qquad \textbf{Eq. 56}$$

This dissipation function corresponds to the case (c) in Figure 49.

In this model, as Chua-Yang model, the neighborhood radius is constant, the interconnection coefficients are independent from the cell position in the array, the operators C and D are null and the operators A and B are linear. Therefore, the differential equation which describes the dynamic behavior of processing cell is:

$$\tau \frac{d}{dt} x_{ij}(t) = -g(x_{ij}) + z_{ij} + \sum_{k=-r}^{r} \sum_{l=-r}^{r} \left[a_{kl} \cdot y_{((i+k),(j+l))} + b_{kl} \cdot u_{((i+k),(j+l))} \right] \qquad \textbf{Eq. 57}$$

The FSR model is reduced to the Chua-Yang model for $m = 1$.

In both models, the Chua-Yang model and FSR model, the output variable is bounded by the interval [-1,1]. And although the state variable is bounded in both models, the maximum value for state variable in Chua-Yang model [Chua88] is determined by the expression:

$$x_{\max} = 1 + z_{ij} + \sum_{\substack{(k,l)\in N_r(i,j) \\ (k,l)\neq(i,j)}} \left\{ \left| A_{ij,kl} \right| + \left| B_{ij,kl} \right| \right\} \qquad \textbf{Eq. 58}$$

This equation implies that the state variable has a range dependent on coefficients and different from output variable. While in FSR model, the state variable and output variable has the same range due to the non-linear dissipation function considered.

From a VLSI implementation point of view, the CNN has appropriate features like the local interconnection, regularity and translational invariance, present in Chua-Yang and FSR models. Also, the matching of ranges associated to the output and state variables for FSR model allows the simplification of design process and an optimization of area and power consumption required by the integrated circuit [Domi97][Rodr93]. Because the state variable x_{ij} and output variable y_{ij} have the same value in FSR model, they can be identified, thus eliminating the nonlinear VCVS at the output of cell in the Chua-Yang model, see Figure 50. Also, the maximum values of the state variable x_{ij} , during transient and at steady-state, do not depend on the template coefficients. The voltage excursions of the state variable node will be the same no matter which set of templates we are applying. This is a very useful result from the VLSI implementation point of view, because, on one side, states variables are bounded avoiding to be clipped by the non-linear characteristics of the basic circuit blocks, on the other side, because programmability is restricted if the signal range varies depending on the template elements [Espe94a].

2.6.4. IMAGE PROCESSING BASED ON CNN TEMPLATE

The image processing can be carried out using the CNN concept. An initial image can be represented by the state variables (x_{ij}) of cell array and an input image can be assigned to the cell input (u_{ij}). When the coefficients of interaction operators (a_{kl}, b_{kl}) and bias term (z_{ij}) are specified, the dynamic evolution law is determined. And under specific boundary conditions, after temporal evolution of processing array an output image is obtained. This process represents the basic function of image processing. Parameters A, B and z determine the operation to be performed in order to map an input image **u** into an output image **y**, assuming that the network has an initial state **x**(0). The CNN can be seen as a processing system.

The previous operation requires that the CNN converges to a stable state during the temporal evolution. Therefore, the convergence conditions must be determined in order to implement image processing algorithms.

The convergence properties of this CNN model are analyzed in [Chua88] and [Carm01] by using the Lyapunov’s method [Guck83]. Considering that every cell in the CNN but cell $C(i, j)$ has evolved to a steady state. Eq.53 can be re-written:

$$\tau \cdot \frac{dx_{ij}(t)}{dt} = -x_{ij} + a_{00} \cdot y_{ij} + k_{ij} = F(x_{ij}) \qquad \textbf{Eq. 59}$$

Where k_{ij} groups the contributions of the neighborhood and bias term:

$$k_{ij} = \sum_{\substack{k=-r \\ k\neq 0}}^{r} \sum_{\substack{l=-r \\ l\neq 0}}^{r} a_{kl} \cdot y_{(i+k)(j+l)} + \sum_{k=-r}^{r} \sum_{l=-r}^{r} b_{kl} \cdot u_{(i+k)(j+l)} + z_{ij} \qquad \textbf{Eq. 60}$$

Studying the dynamic routes for cell $C(i,j)$, which constitutes a non-linear system, different equilibrium points are found, $F(x_{ij}) = 0$, associated to different values of k_{ij} and a_{00}. A detailed study of these dynamic routes is developed by Ricardo Carmona Galán in [Carm01].

For $a_{00} > 1$, none of the cells in a CNN has stable equilibrium points in the linear region of the sigmoidal output function (Figure 48). This means that the output voltage at steady state has a saturated value for every cell in the CNN. Therefore, the CNN output matrix y is binary when the network reaches steady-state. Depending on k_{ij} value the cell can have one or two stable equilibrium points, each one in a determined saturated region, see Figure 51.

For some applications, this result allows the implementation of processing algorithms with binary or digital outputs.

On the other hand, for $a_{00} < 1$, the dynamic routs are completely different. A single stable equilibrium point can exist in the linear region of the sigmoidal output function (the central region of F(x$_{ij}$)). The location of the equilibrium point depends on the magnitude of k_{ij}. It can be found in the linear region or in the saturated regions of the output non-linear function, see Figure 52.

This dynamic behavior allows the implementation of processing algorithms for gray level images using the CNN (gray level image processing).

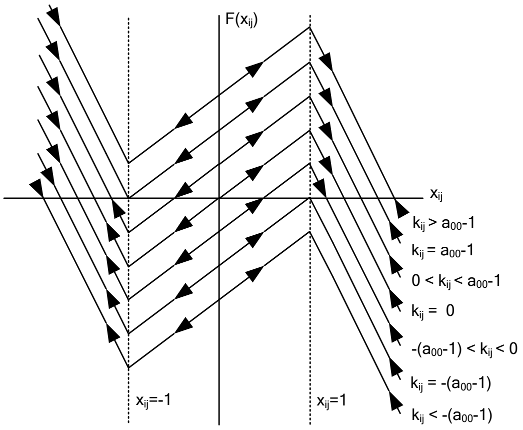


Figure 51. DYNAMIC ROUTE FOR A$_{00}$>1.

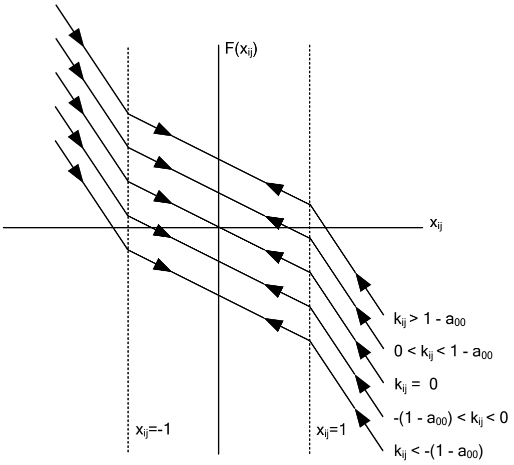

F(x$_{ij}$)
x$_{ij}$
k$_{ij}$ > 1 - a$_{00}$
0 < k$_{ij}$ < 1 - a$_{00}$
k$_{ij}$ = 0
-(1 - a$_{00}$) < k$_{ij}$ < 0
k$_{ij}$ < -(1 - a$_{00}$)
x$_{ij}$=-1
x$_{ij}$=1

Figure 52. DYNAMIC ROUTE FOR A$_{00}$<1.

2.6.5. CNN TEMPLATES DESIGN

The templates **A** and **B**, and the bias term **z_{ij}** together with a particular boundary conditions determine the dynamic evolution of processing array and thus define the processing operation to be applied on an input image.

Designing the CNN templates to realize a particular operation can be accomplished by different approaches. There are algorithms based on analytical methods where the operation to be performed is mapping into a set of local dynamic rules so as the coefficients can be derive analytically [Chua91] [Mats90].

There are applications where it is difficult to derive a functional form for the set of dynamic rules or conditions associated to the template coefficients. In this case, a different approach to compute the necessary template coefficients for a particular operation is learning [Noss96].

The learning algorithms are based on a set of **Q** input-output pairs which receives the name of training set. Each pair in this training set is composed generally by an initial state vector $\vec{x}^q(0)$, an input image vector $\vec{u}^q$ and the desired output image vector $\vec{d}^q(\infty)$. Given a parameter vector $\vec{p} = \left[a_{-1-1}, a_{-1,0}, ..., b_{1,0}, b_{1,1}, z\right]^T$ composed by the template coefficients and bias term, the objective is to find a parameter vector $\vec{p}$ for the corresponding initial state and input, such that the network output $\vec{y}^q(\infty)$ matches the desired output vector $\vec{d}^q(\infty)$ for all pairs belonging to the training set, $q = 1, ..., Q$.

This problem consists in the minimization of an error function defined on the parameters space. The error in the input-output mapping for cell $C(i, j)$ associated to the parameter vector $\vec{p}$ is defined as a distance metric of norm ν:

$$e_{ij}^q\left(\vec{p}\right) = \left|\vec{d}_{ij}^q(\infty) - \vec{y}_{ij}^q(\infty)\right|^\nu \qquad \textbf{Eq. 61}$$

Then, the error surface $\varepsilon\left(\vec{p}\right)$ is defined as the sum of all the individual cell errors over the complete training set:

$$\varepsilon\left(\vec{p}\right) = \sum_{q=1}^{Q} \varepsilon^q\left(\vec{p}\right) = \sum_{q=1}^{Q}\sum_{i=1}^{M}\sum_{j=1}^{N} e_{ij}^q\left(\vec{p}\right) \qquad \textbf{Eq. 62}$$

Taking into account that the non-linear characteristic of CNN generates an irregular error surface in which deterministic optimization methods based on gradients can easily fail to converge to a global minimum, the use of stochastic methods like genetic algorithms [Koze93] and perturbative approximation methods [Cauw96] is justified.

Therefore, templates can be found to develop a wide variety of operations onto a binary or gray scale input image [Rosk95]. Some of them represent a non-linear relation between the state variable and the neighborhood inputs and outputs [Rosk90]. Interesting operations can be implemented with non-linear operators. These non-linear templates can be decomposed and implemented by linear operators if complex control configurations and local memories are provided at the cell level.

2.6.6. THE ACE AND CACE CHIPS

ACE and CACE chips are CNN-based array processors. Each processing cell is based on the Full Signal Range CNN model (FSR), described in section 2.6.3.

Both system architectures are based on CNN Universal Machine concept (CNNUM) [Rosk93]. The CNN Universal Machine is a hybrid computing architecture that combines programmable spatio-temporal dynamics, supported by an analog processing circuit core (CNN), with programmable logic and control driven by digital signals. The CNNUM architecture is depicted in Figure 53.

There are two main blocks: the analogic processing cell array which contains the CNN-core and the Global Analogic Program Unit (GAPU) which store the analogic program and controls its execution. GAPU is composed by two main functional blocks. On one side the storage unit, containing the machine code instructions for the analog and logic operators and switches configuration. On the other side, a control unit which decodes these instructions into a microcode, that is transmitted to the processing cells. Therefore, the CNNUM architecture is a massively parallel SIMD processor.

Conceptually ACE and CACE have similar architecture described in Figure 54.

There are two important differences between both systems. One is related with the I/O data interface, since the data interface of CACE chip is analog and ACE system includes a digital I/O interface with a column ADC/DAC approach. The other difference is the processing cell which will be described in the following sections. ACE cell contains a single-CNN layer together with an optical sensor. Thus, the ACE can process the images directly acquired by the system or work as co-processor. On the contrary, the CACE cell includes two interconnected CNN layers which perform complex-dynamics analogue computing, but does not include sensing function and only works as co-processor.

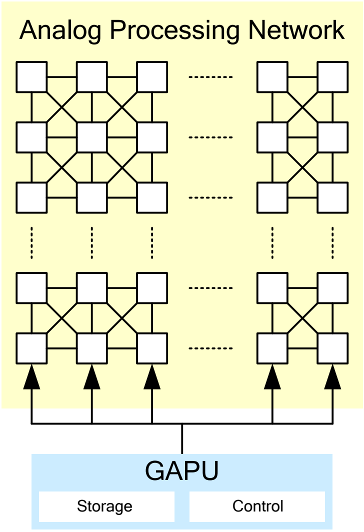


Figure 53. CNN UNIVERSAL MACHINE ARCHITECTURE.

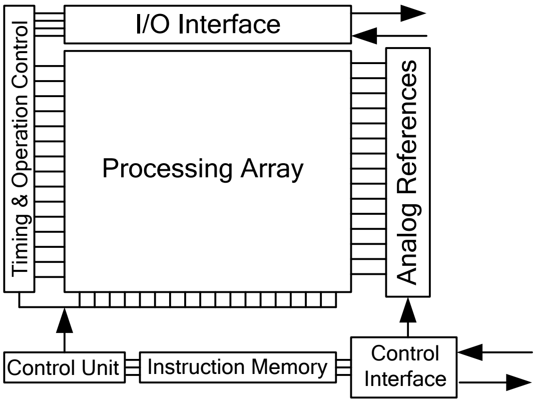


Figure 54. ACE & CACE SYSTEM ARCHITECTURE.

2.6.6.1. ACE PROCESSING CELL

This section describes the functionality performed by the processing circuitry included at pixel level. The elementary processing unit can be defined as an association of several functional blocks sharing a data bus (denoted by ACE bus), which is actually a single electrical node. Data transferences between blocks are carried out through this internal bus. Figure 55 depicts a block diagram of ACE cell.

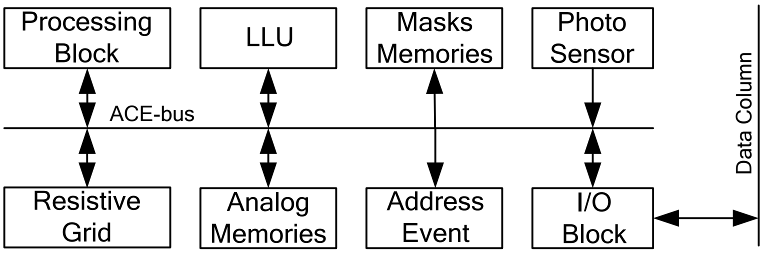


Figure 55. BLOCK DIAGRAM OF ACE PROCESSING CELL.

The pixel includes the following functionalities:

- Local Analogue Memory (LAM) module: It is an internal memory which can store several pixel values (image storage). The I/O access to the LAM is done via the internal data bus. Therefore, this module can receive and send data to the rest of functional blocks through the ACE bus.
- Local Logic Unit (LLU): It is a fully programmable two-input logic device. It performs logic operations defined by a programmable truth table.
- Resistive Grid: This module performs spatial low pass filters.
- Address Event Module: The address event capability provides a fast form of downloading sparse binary images. This module belong to the processing cell takes its input from the internal data bus and provide its output directly to the address event detection circuitry at the peripheral of the processing core.
- Mask Memories: Two dynamic binary RAM memories are used to store binary patterns used as masks in order to develop local enabling/disabling functions. Both mask memories read from internal data bus. One mask denoted by Dynamic Evolution Mask allows selective enabling/disabling of the transient evolution of CNN layer. The other mask called LAM Writing Mask allows selective updating/preserving the contents of a LAM module depending on whether the cell is enabled or not to be overwritten.
- Optical module: Each cell includes an optical module that permits the chip to acquire images projected over its surface. The optical module is a reconfigurable block allowing the use of different acquisition modes and different sensing devices.
- Processing block: It corresponds to the CNN processing layer composed by the synapses (four quadrant analog multiplier) and processing circuitry described in section 2.6.6.3.

2.6.6.2. CACE PROCESSING CELL

In this case, the processing cell includes Local Analog Memories (LAM) and Local Logic Memories (LLM) to store respectively gray scale images and binary images, a Local Logic Unit (LLU) and two different processing blocks. A block diagram of elementary processor is depicted in Figure 56.

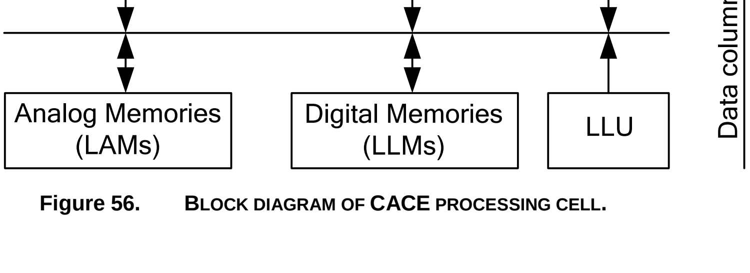


Figure 56. BLOCK DIAGRAM OF CACE PROCESSING CELL.

Each processing block of cell belongs to one of the two different CNN layers of the network. These CNN cores are coupled by some inter-layer weights and their outputs are combined by a third additional layer embodying analog arithmetic. This scheme constitutes a 2nd-order 3-layer CNN, whose conceptual diagram is represented in Figure 57.

The evolution law of each cell, $C(i,j)$, following this 2nd-order 3-layer CNN model is given by these two coupled differential equation:

$$\tau_1 \cdot \frac{dx_{1,ij}(t)}{dt} = -g\left[x_{1,ij}(t)\right] + \sum_{k=-r_1}^{r_1} \sum_{l=-r_1}^{r_1} a_{11,kl} \cdot y_{1,(i+k)(j+l)} + b_{11,00} \cdot u_{1,ij} + a_{12} \cdot y_{2,ij} + z_{1,ij}$$

$$\tau_2 \cdot \frac{dx_{2,ij}(t)}{dt} = -g\left[x_{2,ij}(t)\right] + \sum_{k=-r_1}^{r_1} \sum_{l=-r_1}^{r_1} a_{22,kl} \cdot y_{2,(i+k)(j+l)} + b_{22,00} \cdot u_{2,ij} + a_{21} \cdot y_{1,ij} + z_{2,ij}$$

Eq. 63

Where the non-linear losses term and the output function in each layer are those of the FSR CNN model, described in section 2.6.3 .

Figure 58 depicts the circuital diagram of 2nd order CNN model described by Eq.63. Programming different dynamics in this CNN is possible by adjusting the template elements and the time constants associated to each layer.

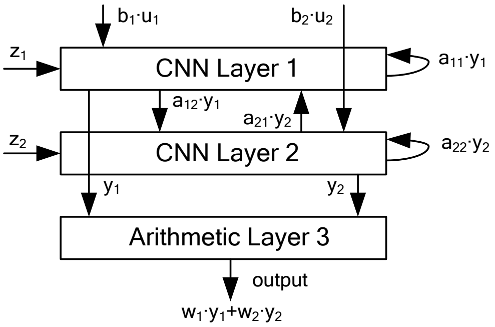


Figure 57. CONCEPTUAL DIAGRAM OF THE 2ND ORDER CNN.

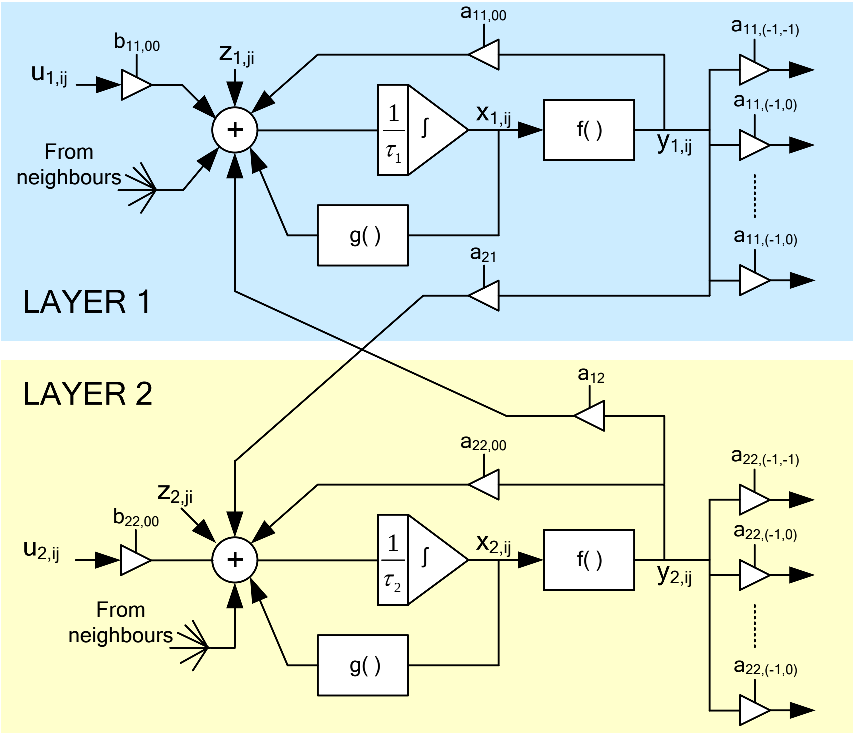


Figure 58. CIRCUITAL DIAGRAM OF CACE PROCESSING BLOCKS.

2.6.6.3. PROCESSING CIRCUITRY: CNN LAYER AND SYNAPSES

In this section, the physical implementation of CNN processing layer used in ACE and CACE chips is presented. This processing circuitry is composed of four main building blocks: an input block based on a class II current conveyor which collects the current contributions from neighboring cells, a current memory used to store and subtract an offset term generated by the chosen synapse structure (based on S^{3}I regulated cascade current memory), a non-linear block which processes the input current (based on voltage comparators and switches) and finally the synapse which is a four quadrant analog multiplier [Espe94b][Domi98]. Figure 59 depicts a block diagram of this processing core.

One of most important blocks in CNN cell is the synaptic block, because the majority of the circuitry structures employed in the cell are strongly dependent on its implementation [Carm01]. The ideal characteristic associated to the synapse is described by the following equation:

$$I_o = k \cdot V_w \cdot V_x \qquad \textbf{Eq. 64}$$

Where V_w and V_x represent the weight and cell state signals, I_o is the output current and k is a constant with dimensions [A/V^2]. If the cell state and the weight signals are represented by voltages, they can be easily conveyed to any high-impedance node by a simple wire. On the other side, the output of the multiplier should be represented by a current. In this way, the contributions of all neighbors can be summed at the input of the processing core wiring all these contributions concurrently to a low-impedance node.

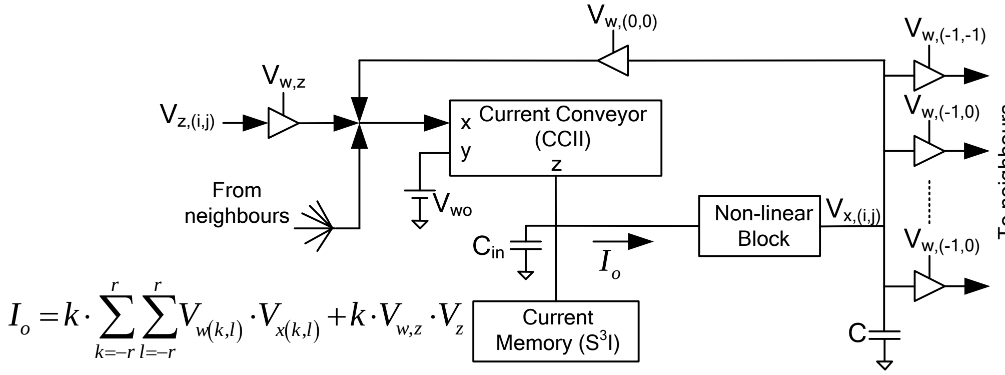


Figure 59. ANALOGUE PROCESSING CORE.

There are several CMOS circuits for implementing a multiplier [Espe94b] [Gilb68] [Socl85]. Among them, there is one approach which is based on a single MOS transistor operating in the ohmic region. Using one transistor reduces the area requirements and a better relation between bias power and signal power is obtained (in the saturation region the information is carried by a small fraction of the actual currents flowing through the devices).

Figure 60 describes a PMOS transistor operating in ohmic region. A p-type channel has been considered because its higher channel resistance allows smaller current, and so power consumption, for the same transistor length.

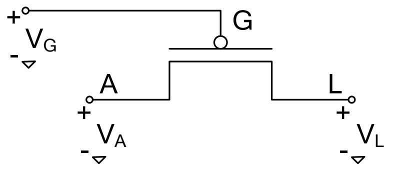


Figure 60. SINGLE MOS MULTIPLIER.

Considering:

$$\begin{aligned} V_G &= V_{xo} + V_x \\ V_A &= V_{wo} + V_w \\ V_L &= V_{wo} \end{aligned} \qquad \textbf{Eq. 65}$$

Being V_{x0} and V_{w0} bias references to guarantee that the transistor operates in ohmic region, the output current is given by the equations:

$$I_o = -\beta \cdot V_w \cdot V_x - \beta \cdot V_w \cdot \left(V_{xo} + \left| V_T^{'} \right| - V_{wo} - \frac{V_w}{2} \right) = -\beta \cdot V_w \cdot V_x + I_{offset}\left(V_w\right) \qquad \textbf{Eq. 66}$$

$$\beta = \mu_o \cdot C_{ox}^{'} \cdot \frac{W}{L} \qquad \textbf{Eq. 67}$$

$$V_T^{'} = \begin{cases} -\left|V_{T0p}\right| - \gamma\left(\sqrt{\phi_B + V_{DD} - \left(V_{wo} + V_w\right)} - \sqrt{\phi_B}\right) & V_w \geq V_{wo} \\ -\left|V_{T0p}\right| - \gamma\left(\sqrt{\phi_B + V_{DD} - V_{wo}} - \sqrt{\phi_B}\right) & V_w < V_{wo} \end{cases} \qquad \textbf{Eq. 68}$$

The correct operation of single-MOS multiplier requires a virtual reference at node L (V_L), this low-impedance node is implemented by the current conveyor, whose circuital realization is depicted in Figure 61.

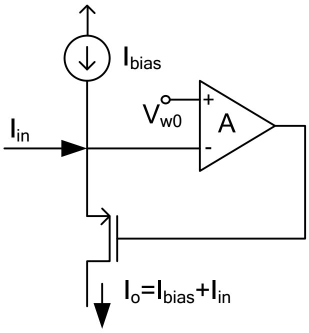


Figure 61. CURRENT CONVEYOR CIRCUIT.

The offset term $I_{offset}\left(V_w\right)$ is stored in the current memory to be subtracted later. In order to achieve the necessary accuracy levels, a S^{3}I current memory is employed [Toum93]. Its corresponding circuit is shown in Figure 62.

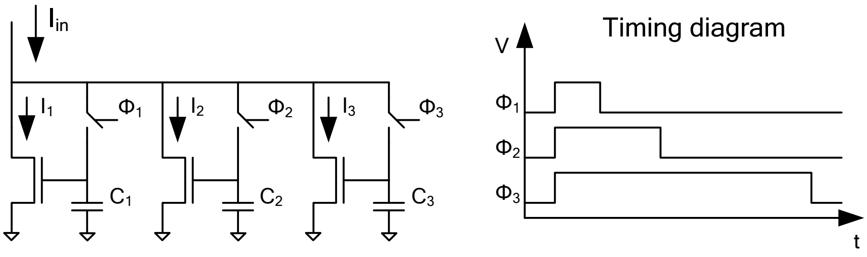


Figure 62. S^{3}I CURRENT MEMORY CIRCUIT.

Therefore, the CNN operation requires an offset calibration step. For this purpose, before CNN operation, but after new weights has been uploaded, all synapses are reset to $V_x = 0$ ($V_G = V_{x0}$). Then the resulting current, which is the sum of the offset currents of all synapses concurrently connected to the same node, is memorized. This value will be subtracted on-line from the input current when the CNN loop is closed to run CNN dynamic.

Associated to the non-linear block, there are two operation modes: soft limiting mode and hard limiting mode.

In soft limiting mode, the total input current, after flowing through the current conveyor, is directed towards an electrical node in which a non-linear resistor is in charge of keeping the voltage value within some limits. Except for this resistor, the node impedance is purely capacitive. The ideal characteristic of this non-linear resistor is given by:

$$g\left(V_x\right)=\begin{cases} -\infty & V_x \leq V_{xL} \\ 0 & V_{xL} < V_x < V_{xH} \\ \infty & V_x \geq V_{xH} \end{cases} \qquad \textbf{Eq. 69}$$

The total output current flowing into the capacitive node with the non-linear resistor produces the CNN dynamic process when the loop is closed, which is described by the equation:

$$C\cdot\frac{dV_x}{dt}=-g\left(V_x\right)+k\cdot\sum_{k=-r}^{r}\sum_{l=-r}^{r}V_{w(k,l)}\cdot V_{x(k,l)}+k\cdot V_{w,z}\cdot V_z \qquad \textbf{Eq. 70}$$

The soft limiting mode is intended to be used in applications with gray-scale images. On the other hand, the hard limiting mode is intended for those applications providing a binary result. In this case, the total output current is sent to the capacitive input of a voltage comparator. The comparator operation is described by:

$$V_x=g(V_{in}-V_{x0})=\begin{cases} V_{x\min} & V_{in}-V_{x0}<0 \\ V_{x\max} & V_{in}-V_{x0}>0 \end{cases} \qquad \textbf{Eq. 71}$$

The dynamic evolution is modeled by the equation:

$$C_{in}\cdot\frac{d\left(V_{in}-V_{x0}\right)}{dt}=k\cdot\sum_{k=-r}^{r}\sum_{l=-r}^{r}V_{w(k,l)}\cdot V_{x(k,l)}+k\cdot V_{w,z}\cdot V_z \qquad \textbf{Eq. 72}$$

In conclusion, running spatial-temporal dynamics in these networks requires following several initialization and calibration steps. First of all, acquisition of the input image and auxiliary masks and patterns. After that, the analog instruction constituted by the set of synaptic weights needed for a specific operation, is selected and transmitted to all cells in the array. Then, the offsets of critical Operational amplifiers are calibrated. After this, the time-invariant offsets of the synaptic blocks are computed and stored in the current memories. At this state, network is almost ready to operate. Then, the state capacitors and the feedforward synapses are initialized by means of the appropriate switch configuration, and the network evolution is run by closing the feedback

Loop in each processing element (according to the current freezing mask). Finally, before stopping the network, the final state is stored in a LAM for further operations.

2.6.7. LIMITATIONS OF THE ACE AND CACE CHIPS

As mentioned in previous sections, in-pixel processing approach there is a tradeoff between accuracy and pixel area, which affects directly to the achievable fill factor. For a given pixel pitch, the area devoted to processing circuitry is removed from sensing area.

On the other hand, the integrated circuit fabrication process is exposed to a number of factors that can provoke the presence of defects on the chip. This phenomenon reduces the production yield and is characterized by the density of defects. This defect density increases with the die area, increasing the risk of suffering from fatal defects in fabrication, and thus, obtaining a smaller production yield. Therefore, yield limits the spatial resolution of processing array that can be integrated in a chip.

From an implementation point of view, another factor that limits the resolution of processing array is the power consumption per cell. For instance, in the case of ACE and CACE chips, at least one synapse is associated to each contribution of neighboring pixels, plus the contribution of the pixel itself and the bias term. Therefore, each processing cell contains ten synapses at least. Each synapse generates a current contribution which is summed and integrated by the processing circuitry. This means that additional circuitry is required for the correct operation of multipliers and processing of signal currents. In this case, this additional circuitry is comprised by a current conveyor, current memory and limiter, whose operation implies determined power consumption. In CACE chip, the sum of all contributions ranges from 18µA to 46µ [Carm01].

This total current implies a power consumption of 0.15mW in the worst case, considering a power supply of 3.3V. Then, the power consumption associated to the synapse operation, without considering the other processing circuitry, for an array of 32x32 resolution is around 0.5W. In the case of quarter CIF resolution (176x144), the power consumption increases to 3.8W. This power consumption produces an increment of chip temperature, being very important to consider the thermal characteristics of package in the calculation of this increment. An usual thermal coefficient for packages is around 10º-20ºC/W, this means that the die temperature increases between 10º-20ºC per watt. Thus, the temperature increment for a processing array with resolution 176x144 ranges from 65ºC to 103ºC, where a room temperature of 27ºC has been considered. This range of temperature is high for the photodiode operation, the dark current increase exponentially with the temperature degrading the quality image significantly.

Another problem or design issue related with the power consumption and multiplier implementation is the distribution of weight voltages (V_w), which are associated to a diffusion terminal of MOS transistors (Figure 60). The metal lines that distribute the weight references drain current, and thus, the voltage drop provoked by line resistance must be considered. This drop in the weight voltages mean that processing cells in the array have different weights generating gradients in the processed image. For example, the mean filter is a low pass filter characterized by a template whose coefficients are equals:

$$A=\begin{pmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{pmatrix}$$

All pixels contribute with the same weight, being the output image:

$$y_{ij}=\frac{1}{9}\sum_{k=-r}^{r}\sum_{l=-r}^{r}x_{kl} \qquad \textbf{Eq. 73}$$

For a uniform input image (x_{ij}), the output image (y_{ij}) should be the same. But in the case that weight contributions change along the array, the resulting image presents a gradient. For instance, a linear horizontal drop in the weight distribution means that the convolution template depends on column coordinate:

$$A(j)=\begin{pmatrix} \frac{1}{9}\cdot\left(1-\varepsilon_{1,-1}\cdot j\right) & \frac{1}{9}\cdot\left(1-\varepsilon_{1,0}\cdot j\right) & \frac{1}{9}\cdot\left(1-\varepsilon_{1,1}\cdot j\right) \\ \frac{1}{9}\cdot\left(1-\varepsilon_{0,-1}\cdot j\right) & \frac{1}{9}\cdot\left(1-\varepsilon_{0,0}\cdot j\right) & \frac{1}{9}\cdot\left(1-\varepsilon_{0,1}\cdot j\right) \\ \frac{1}{9}\cdot\left(1-\varepsilon_{-1,-1}\cdot j\right) & \frac{1}{9}\cdot\left(1-\varepsilon_{-1,0}\cdot j\right) & \frac{1}{9}\cdot\left(1-\varepsilon_{-1,1}\cdot j\right) \end{pmatrix}$$

Considering negligible the mismatch of drop error among contributions, the result of mean convolution depends on column coordinate (j), generating a horizontal gradient:

$$y_{ij}\left(j\right)=\frac{1}{9}\cdot\left(1-\varepsilon\cdot j\right)\sum_{k=-r}^{r}\sum_{l=-r}^{r}x_{ij}$$

The drop associated to the power supplies distribution, which is provoked by the current demand of processing cells, has a similar effect on the CNN operation.

Therefore, depending on error drop allowed by the processing circuitry in order to achieve the required accuracy in the algorithm implementations, there is a tradeoff between current consumption per cell and width of distribution lines (for weight references and power supplies). This width is determined by the current that flows through the line and the length of this one. Then, the resolution is determined by current consumption per cell and processing array area.

There is an additional issue related with the errors introduced by the references and power supplies distribution, the dependence of these errors with the technological parameter variations and temperature. Although this errors due to gradients in the weight references and power supplies could be compensated by an optimization of coefficients and inputs associated with the CNN operation or by any other calibration procedure, these correction factors may change with temperature or may not be valid for different chips. This is a critical problem for the production and introduction into the market of a particular vision system.

2.6.8. OTHER TYPES OF COMPUTATIONAL SENSORS

As read in sections 2.3.3 and 2.6, the CNN approach is one of the different architectures that can be considered for the implementation of an Analog Parallel Array Processor (APAP).

The CNN approach processes in parallel all contributions, which represents the maximum degree of parallelism to implement. But in this case, parameters such as power consumption and processing array resolution become limiting. That is, there is a tradeoff between the level of parallelism in processing and power consumption-resolution figures. In order to optimize the cell area and reduce the power consumption per cell, which would allow increasing the resolution of processing array at the expense of a reduction in computing power, the different contributions could be processed serially. This means that instead of nine multipliers working in parallel, the cell operation requires only one multiplier developing the operations serially, with the corresponding reduction in area and power consumption. This strategy increases the cell processing time, which implies reducing the computational power, but also allows a sensing-processing array with higher resolutions. Note that this computational level achieved by having one operator (one multiplier) per pixel does not compromise the speed requirements demanded by the vast majority of applications.

Most of the implemented APAPs include the minimum circuitry necessary to perform a specific image processing task at pixel level to optimize power and area; several examples have been reported in section 2.3.3. They are application specific information processors at pixel level. Therefore, this approach implies to develop vision systems for a specific processing algorithm with a low capability of programmability. There is a tradeoff between the flexibility or programmability level and implementation efficiency (power consumption and area).

Ideally, from a functional point of view, a high level of programmability to carry out many processing algorithms is a desirable characteristic of a vision system with the objective of engage the maximum number of applications.

In order to include the greatest possible number of pixel functions ensuring a level of efficiency that makes viable the system, a set of basic operations required by different image processing tasks belonging to the early-processing stage has to be defined. And these basic operations can be implemented efficiently by different type of circuits at pixel level. For instance, the spatial low pass filters can be implemented by a resistive or diffusion grid (section 3.2.8 of Chapter 3), the arithmetic operations can be developed by switched- capacitors circuits (section 3.2.5 of Chapter 3). Spatial high pass filters can be obtained based on these arithmetic operations and low-pass filtering.

Generally, in case that a generic spatial filter has to be implemented, the spatial-domain functions performed in the early-stages of digital image processing in image enhancement algorithms can be expressed as:

$$y(i,j) = T\left(x(i,j)\right) \qquad \textbf{Eq. 74}$$

Where $x(i,j)$ is the input image, $y(i,j)$ the processed image and T the operator. One of the principal approaches in this formulation is based on the use of so-called templates or convolution masks. Basically, a template is a two-dimensional matrix whose coefficients are chosen to detect a given property in the image. Usual dimensions of this template is 3x3, Figure 63 shows the coefficients and corresponding image pixel locations.

The convolution operation for a 3x3 template is described by the following equation:

$$
\begin{aligned}
y(i,j) = T\left(x(i,j)\right) = w_1 \cdot x(i-1,j-1) + w_2 \cdot x(i-1,j) + w_3 \cdot x(i-1,j+1) + \\
+w_4 \cdot x(i,j-1) + w_5 \cdot x(i,j) + w_6 \cdot x(i,j+1) + \\
+w_7 \cdot x(i+1,j-1) + w_8 \cdot x(i+1,j) + w_9 \cdot x(i+1,j+1)
\end{aligned}
\qquad \textbf{Eq. 75}
$$

w_1 (i-1,j-1)	w_2 (i-1,j)	w_3 (i-1,j+1)
w_4 (i,j-1)	w_5 (i,j)	w_6 (i,j+1)
w_7 (i+1,j-1)	w_8 (i+1,j)	w_9 (i+1,j+1)

Figure 63. 3X3 CONVOLUTION MASK.

In this generic case, the basic operator for image processing can be implemented in the analog domain at pixel level by a switched-capacitor circuit (multiplier-accumulator, section 3.2.5 and section 3.2.7 of Chapter 3), where the different coefficients of template are defined by relations between capacitors. Each contribution associated to a determined coefficient is calculated and added or subtracted serially to obtain the processed pixel value. The main characteristic required by pixels to perform convolutions is the interconnectivity among neighbors.

In this physical implementation, the distribution of analogue references which are necessary to perform the circuit operation and the signal voltages are distributed by purely capacitance nodes. Therefore, the drop associated to the current flow is not present in these references as in CNN approaches. Only the power supplies drop must be considered during the design phase. But, in this case, this drop does not affect directly to the operation, being a second order error given by the non-ideal behavior of analogue circuit, which can be compensated by design.

So far, only an APAP (Analogue Parallel Array Processor) architecture at the focal plane (processing-in-pixel) has been considered for the implementation of smart sensors. But, there is other possible interesting architecture which reduces the power consumption respect to the previous one and solves the pixel fill-factor problem, obtaining higher spatial resolution and accuracy, since the processing circuitry has available a greater area budget. This new architecture is based on a column processing approach, similar to the column ADC approach implemented in the Digital Image Sensors. This new approach for the smart sensor consists of a conventional pixel array and associated with each column there is an elementary processor, which has the ability to connect with the neighboring columns. Figure 64 depicts the proposed architecture.

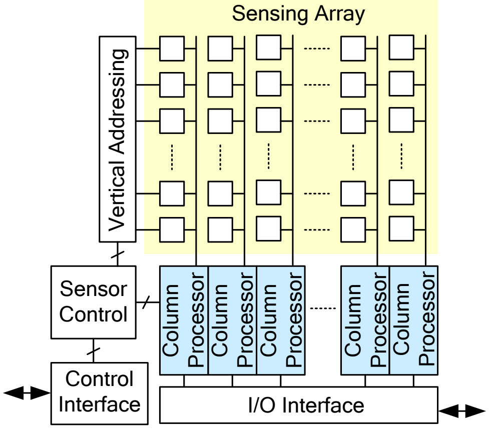


Figure 64. COLUMN PROCESSING APPROACH.

In general, each elementary column processor can be composed by analogue processing circuitry, analogue memories, A/D conversion and threshold operation as interface with digital domain, digital memories and elementary digital processors. Figure 65 presents a general block diagram of an elementary digital processor. Usually, the elementary digital processor is a custom digital circuit designed and optimized to performed specific tasks.

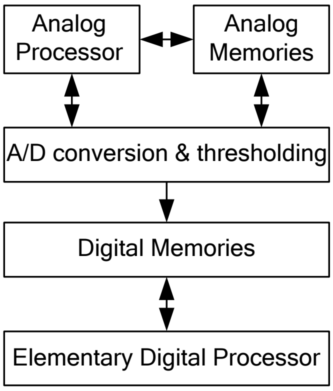


Figure 65. ELEMENTARY COLUMN PROCESSOR BLOCK DIAGRAM.

A particular case is the column processor composed by an A/D converter plus digital memories and elementary digital processor. This case can be interpreted as the first step towards parallel processing architecture on-chip in order to perform the tasks of early-processing algorithms. It is an approach similar to that followed by conventional digital cameras. But in this case, due to the implementation of low-level processing circuitry on chip, the system gets a higher integration level and avoids the speed limitations associated with the data interface. This last feature together with a higher parallelism in the processing implies an increment in the computing power of system.

The introduction of a microprocessor into the smart sensor with access to digital memories of elementary column processors allows the implementation of high-level processing algorithms, given rise to the concept of vision system on chip.

2.7. CHALLENGES FOR CAMERAS AND VISION SYSTEMS. OVERVIEW OF THE STATE-OF-THE-ART

Challenges are determined by market demands and society requirements. For instance, in surveillance applications, current Closed Circuit TV systems (CCTV) are mostly not automated systems and rely strongly on trained security personnel to perform image analysis, object tracking and identification. The increasing number of cameras makes difficult the analysis in real-time by security personnel. Network bandwidth is another important issue affecting real-time processing needed for crime prevention. The intelligent video surveillance systems (IVSS) will try to provide solutions to these problems. Smart cameras will be one of the fundamental building blocks of the IVSS, making it possible to build and deploy automated, distributed and intelligent multi-sensory surveillance systems capable of tracking humans and suspected objects, analyzing human behaviors, etc. (Read section 2.4.5).

Another important application area for vision systems and cameras is the industry Machine Vision. The demand for compact vision systems has been increasing over the years and is primarily driven by the increasing demand for better production efficiency and quality control in industries such as robotics, semiconductor, electronics, pharmaceutical, manufacturing, food, plastics, printing, etc. The tasks of these compact systems perform bar-code reading, part inspection, flaw detection, surface inspection, dimensional measurement, assembly verification, print verification, object sorting, optical character recognition, etc. Other important application areas for embedded vision systems, where their demand is increasing too, are Intelligent Transport Systems (ITS), Human Computer Interface (HCI), games, toys, automobiles (collision avoidance, passenger monitoring), unmanaged vehicles, etc.

All previously submitted applications require either high-spatial-resolution or high-frame rate or both. This fact means a high throughput for the data interface of image sensor and camera. Current digital cameras on the market that contain a digital CMOS image sensor can achieve very high transmission frequencies above 78 Gbits per second, which corresponds to a sensor with a resolution of 1280x1024, including 12 bits AD conversion and running at 5000 frames per second. These throughputs imply currently that a custom interface must be designed to interconnect these high frame rate cameras with others blocks in a vision system, which are responsible for image processing. This custom interface increases the production cost of vision system. Standard interfaces on the market such us GigE, USB, Firewire, Camera Link or CoaXpress do not achieve these throughputs. The fastest interface is CoaXpress whose maximum throughput is 6.25 Gbits per second.

Therefore, the standard communication interfaces used currently in vision systems are a bottle neck and limit the operation speed of vision systems. In this context, embedded vision systems (as smart cameras) incorporating parallel processing in the focal plane may be an alternative; this approach is able to avoid this bottle neck in the image transmission, because only a compress version of real images with useful information is transmitted.

Technically, high frame rates are achieved in the capture operation by CMOS image sensors. Thus, CIS technology allows designing high speed cameras which continue to shrink in size, reduce in cost and gain in performance (Moore's law) thanks to its ability to integrate into a single chip; all the building blocks that comprise the image sensor: pixel array, readout circuitry (including the AD converters), sensor control block, the digital data interface, etc. (see section2.4.1). For instance, the column ADC approach is commonly used in high speed digital image sensors. But this high speed operation in digital CMOS image sensors is attained losing image quality. Therefore, there is a trade-off between speed-parallel processing level and quality.

One of the main lines of research and development in CIS technology is improving image quality in order to reach levels comparable to those obtained in CCD technology, which is a mature technology with excellent image quality; high sensitivity and long exposure time, thanks to extremely low noise, high quantum efficiency and very high fill factor.

Regarding to the temporal read noise, current readout circuitry implemented in image sensors chip has noise levels similar to that obtained in CCD image sensors (1-8 e), operating both sensors at same frame rates. The trend in digital image sensors is the design of new readout circuitry for high-speed applications extending this low noise level to the maximum possible operating frequency.

Other important parameter to obtain a good image quality is the number of photo-generated electrons for a given irradiance, take into account that the SNR parameter is defined by the maximum number of photo-generated electrons and the noise level, read section 2.5.5.2. This parameter is determined by the quantum efficiency and fill factor together with the pixel area. Improvements related to QE require optimization at the process level: optimization of photo-detector doping profile to improve photo-charge collection and top layers to enhance optical transmission. Thanks to new high volume markets, silicon founders are investigating to develop photodiodes with higher QE and lower Dark Current.

To compensate the loss of fill factor, micro-lenses are widely used in CMOS image sensors and backside thinning is currently being developing. In addition, several attempts are on the way to get near 100% fill factor and efficient charge collection. They are based on 3D integrated circuits where photodiodes are made in an optimized detection layer on top of a CMOS readout/processing circuitry layer, see Figure 66.

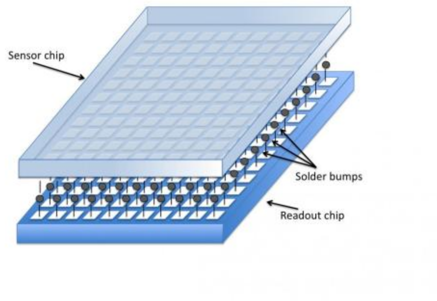


Figure 66. 3D IC TECHNOLOGY BASED ON THROUGH SILICON VIAS (TSV) INTERCONNECTION

Figure obtained from Wikipedia

In summary, current applications demand vision systems with more operation speed and accuracy (high spatial resolution and low noise).

High speed implies the following challenges:

- Higher responsivity in sensing operation in order to reduce the exposure time. This requirement demands increments in quantum efficiency and fill factor.
- Higher frame rate and computational power: This means to develop shorter conversion times during readout process and parallel architectures for data processing. Processing at pixel level limits the fill factor and spatial resolution.
- Limited bandwidth of data transmission channels: This limitation is overcome by the introduction of processing at sensor and camera levels.

And higher accuracy involves the challenges:

- Higher spatial resolution: This requirement implies smaller pixels, which decrease the responsivity per pixel. Also, an increment in the number of pixels carries a higher data throughput.
- Higher precision in the readout and processing circuitry: This objective represents an increase of area. For the case of processing at pixel level, this increment of area affects to the fill factor and spatial resolution.

3D integrated circuit technology is a promising approach to address the previous challenges. This technology allows implementing a massively parallel analog processing without jeopardizing resolution and responsivity, because the readout-processing circuitry is located in a different layer, see Figure 66. In addition, the accuracy achievable by the processing circuitry will be higher since the increment of area required for that may not affect to the pixel area and fill factor. Regarding the accuracy and signal integrity, locating the digital circuitry (memories and micro-processor) in a third different layer reduces the impact digital activity on the effective precision reached by the mixed-signal processors, since sensing devices, mixed-signal circuitry and digital circuitry are implemented in independent substrates.

From a practical point of view, when a camera or vision system achieved the specifications for speed and accuracy to develop a specific task successfully, come into play parameters such as cost and dimensions. Both directly related with the integration level that can be achieved during the system implementation,

2.7.1. CAMERAS-ON-CHIP

As deduced from examples presented at the beginning of section 2.7, the market is demanding cameras more compact. There are applications where the number of cameras required is high (for example, surveillance applications), or the space available for camera location is very limited (for example, machine vision or automobile applications), etc. At this juncture, characteristics such as size and cost must be minimized.

Currently, a digital camera whose architecture is described in Figure 67 can be implemented into a single chip in CMOS technologies. This approach has several advantages demanded by the market: reduction cost, power consumption and size of the camera.

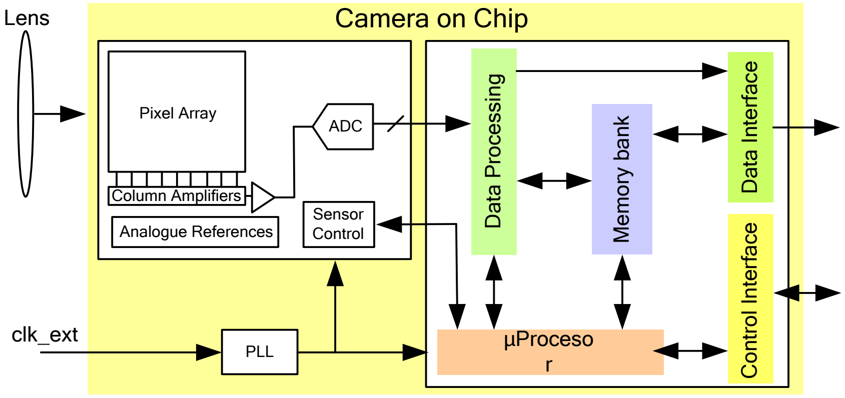


Figure 67. CAMERA-ON-CHIP ARCHITECTURE.

But there are drawbacks or problems in these embedded systems that must be considered during the design phase. One of them is the coexistence on a common substrate of sensor array, mixed-signal circuitry associated to the readout process and digital circuitry which is comprised by a micro-processor, memories and processing custom circuitry. The abovementioned digital blocks will introduce a significant level of noise in power supplies, especially when the frequency operation is very high due to high-spatial-resolution and high speed applications. Therefore, the temporal noise floor associated to the mixed-signal circuits will increase, what implies a degradation of image quality.

Another issue to consider is the power consumption associated to the large digital circuitry, mainly when the frequency operation is high. This power consumption means an increase in chip temperature, which will increase significantly the spatial noise due to higher dark current. Also, the power consumption is critical for stand-alone cameras.

The limitation of resources due to limited area and power consumption, such as the amount of memory that can be implemented on the chip, is a major constraint in the design of the functionalities to be developed by the camera.

In camera-on-chip design, it is important that the communication ports are standard, both the control interface and the data interface, such that the chip can be connected directly without any additional external component to a PC or frame grabber.

The micro-processor is not strictly necessary, since a PC can control and configure the camera directly via a standard control interface (UART, for example). But the introduction of a microprocessor provides it an important added value, operation autonomy, developing all tasks related with the operation of camera (configuration, sensing control, calibration process, etc.). This feature is having a growing demand in the market for stand-alone applications.

2.7.2. VISION SYSTEMS-ON-CHIP

As mentioned previously, there are more and more applications requiring embedded vision systems, such as smart cameras. And one reason for this increment is that standard video cameras have large output bandwidth requirements (directly proportional to the resolution of image sensor and frame rate), while embedded vision systems can have very low data bandwidth requirements at the output, take in mind that the primary function of a vision system is to produce a high-level understanding of the imaged scene in order to extract information at a higher level of abstraction, what allows system to solve some task, take a decision or understand the scene.

This low data bandwidth of embedded vision systems solves the problem of communication among cameras and processing sub-systems that exist in current vision systems for high resolution and high frame rate. The cost and size of these embedded vision systems are minimized when they can be implemented on a single chip.

The problems associated with vision system implementation on a single chip is similar to the camera on chip design, common substrate contamination due to the operation of the digital circuitry, power consumption and limited resources. But in this case, the processing algorithms and control, which have to be implemented on silicon, are more complex; accentuating these problems, see section 2.4.

To extract information and understand the scene, the vision system performs two types of processing task: data-intensive tasks and math-intensive tasks. The latter can be performed by a microprocessor or DSP, but the data-intensive tasks require high speed hardware to deal with the high pixel volume and high frame rate. This hardware with a very high computing power is the main issue in the design of vision-systems-on-chip. This issue opens two lines of development: Processing algorithms and circuital architecture, in order to optimize resources (silicon area) and power consumption in the VSoC implementation.

Data-intensive processing is carried out by a custom hardware to achieve the required computing power. The limited resources available during the design phase and power consumption budget mean that these digital circuits are designed to perform a specific task, there is a tradeoff between silicon area/power consumption and the level of programmability with that is provided the digital circuit processing. Therefore, current VSoCs in the market cannot be usually reprogrammed to perform several tasks or algorithms. In the research and development of new architectures to address efficiently the data-intensive processing on-chip, it is important achieve the maximum level of configurability and programmability. Since this feature allows that VSoC is used in a larger number of applications, reducing development costs.

Apart from the conventional digital implementations, new architectural approaches to perform data-intensive tasks in vision systems on-chip are being developed. These approaches are more efficient from a point of view of area and power consumption, such as massively parallel analog array processors described in section 2.6.6 and CVIS develop in the context of this thesis. The latter appears as a promising option to develop compact and stand-alone vision systems operating at high speed.

CHAPTER 3 - THE Q-EYE PIXEL

Imagers and vision sensors consist of an array of interconnected *pixels*. As described in Chapter 2, typical pixels include a photo-sensor device, and besides, some circuitry for initialization, control of the exposure time, addressing and readout [Ohta08]. Some pixels employed in advanced machine *vision* sensors include also analog memories for either error compensation [Wan11] [Akah06] or A/D conversion at pixel level [Fowl98]. However, conventional image sensor architectures do not embed processing functions in-pixel [Gama05]. On the contrary, the pixel employed in the Q-Eye belongs to the category of the so-called *multi-functional pixels* [Zara11] [Rosk01] [Rodr04]; i.e., pixels that embed circuitry for image acquisition along with circuitry for *concurrent*, *parallel* processing of the incoming image. Concurrent means that images are processes as they are acquired, right at the pixels. Parallel means that all pixels are processed at the same time. In the particular case of the Q-Eye, each pixel is a kind of programmable mixed-signal processor with local memory [Chua02] which can be software-configured [Rosk01] to realize different early vision tasks [Toma06]. Thus, the Q-Eye does not only deliver raw electronic images, but also processed images with reduced data, such as image features, scaled images, etc [Delb06]. Due to the large variety of functions included within the Q-Eye pixel, from now on we will refer to it by the term *cell* instead of pixel.

3.1. Q-EYE CELL ARCHITECTURE

3.1.1. BLOCK DIAGRAM AND EMBEDDED OPERATORS

Figure 1 depicts the Q-Eye cell architecture. Each cell processes data captured point-wise together with data coming from its 8 neighboring cells. Thus, the computations needed for early vision [Toma06] are collectively implemented by the interacting cells.

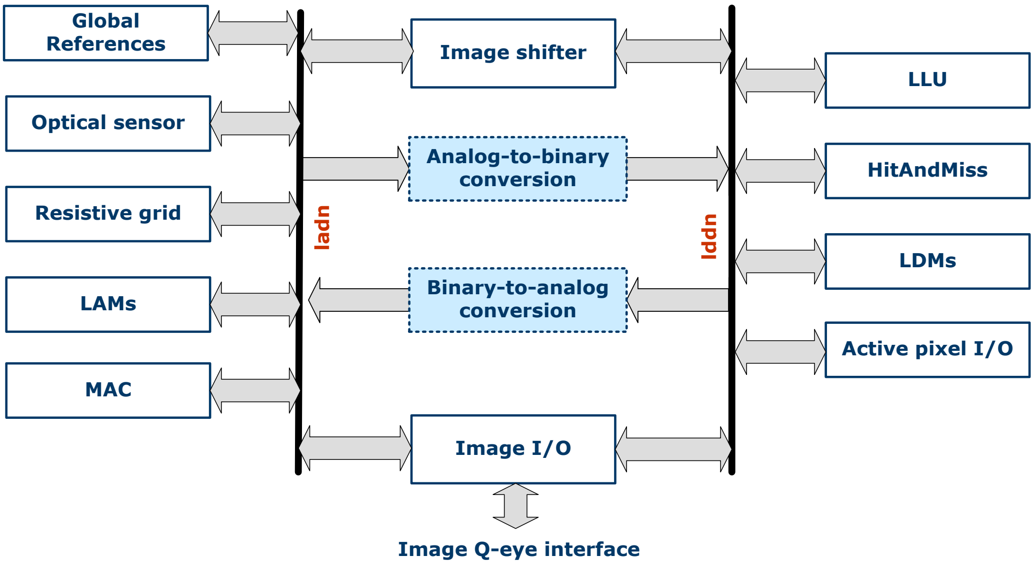


Figure 1. ARCHITECTURE OF A Q-EYE CELL.

These blocks are selected to enhance the flexibility and versatility of the Q-Eye. The goal is making it a *general-purpose* early vision processor, suitable to implement the largest possible variety of tasks [Rodr10a, b]. The rationale for choosing the specific blocks in the figure above is based on previous experience [Espe94a, b] [Rodr93] [Carm01] [Liña02a, b] [Carm03] [Rodr04] [Liña04] and will be explained in the next section. Before doing that, the next paragraphs of this section include preliminary descriptions of each of the blocks in the Q-Eye cell.

SENSING BLOCK (OPTICAL SENSOR). It employs a 4T pixel architecture [Gama05] (see Chapter 2), and a voltage buffer. It is conceived to provide both *linearly* encoded images and *compressively* encoded images high-dynamic range [Tu98], [Deck98]. The sensing device is a P-well photodiode with an active area of 9µm^2 rendering 1.0/V(lux·sec) sensitivity for green light (λ=555nm). The voltage buffer connects the pixel output with any block in the local analogue bus labelled **ladn**. in Figure 1. For example, it can either be stored in an analogue memory (**LAM** block in Figure 1), or filtered by using the **Resistive Grid** module.

GLOBAL ANALOGUE REFERENCES. Several analogue reference voltages, generated by on-chip programmable Digital to Analogue Converters (DACs), are distributed throughout the processing array so that any of these references can be connected in parallel through this block to the bus **ladn**. Thus, this functional block enables parallel loading of a uniform gray level image into the sensing-processing array.

LOCAL ANALOGUE MEMORIES (LAMS) are included inside every cell to temporarily store gray-level (analogue) images. Each cell includes two banks of **LAMs**:

- Bank 1 contains seven memories for general purpose, while
- Bank 2 contains two memories which are used for internal operation.

LAMs are intended for gray-level, analogue image storage with an equivalent resolution of 8 bits. They can store gray-level images coming from any block connected to the bus **ladn**, such as the **Optical Sensor**, the **MAC** or the **resistive grid** module. They can also serve to store input operands for the same blocks.

LOCAL DIGITAL MEMORIES (LDMS). They are intended for storing binary images at cell level. As with the **LAMs**, each cell of the Q-Eye chip contains two banks of **LDMs**, namely:

- Bank 1 contains eight **LDMs**, whereas
- Bank 2 contains one memory.

Only four memories in the Bank 1 are available to the user; the rest of them are employed for internal operations. **LDMs** can be used to either store or output data to the different blocks connected to the bus **lddn**. Unlike **LAMs**, **LDMs** are non-volatile and can store binary images as long as the chip is biased without degradation.

RESISTIVE GRID. It provides direct conductive connections among cells as to perform low-pass Gaussian filters with programmable spatial bandwidth over the input images. The Resistive Grid can be combined with the **Multiplier-Accumulator Circuit** (**MAC**) to implement any type of filter: high-pass, band-pass. Similar to what happens with all blocks inside the Q-Eye cell, the operation of the resistive grid is global. This means that all resistive grids inside all the Q-Eye cells perform the same operation at the same time. There is, however, the possibility of selectively deactivating the operation of this block in certain cells so that the image filtering only takes place in certain region(s) of the image.

MULTIPLIER-ACCUMULATOR (MAC). It supports image addition and scaling in accordance with:

$$I_{out} = I_0 + k \cdot (I_1 - I_2) \qquad \textbf{Eq. 76}$$

Where I_0 is a programmable constant image, k is a constant whose value can be ±1, ±½ and ±2, and I_1 and I_2 are input images. The result of this operation can be accumulated through several iterations, thus giving:

$$I_{out} = I_0 + k\sum_{i=0}^{N} I_i \qquad \textbf{Eq. 77}$$

This block can be programmed to realize a *comparison* between two analogue images and provide an output binary image. Its operation is represented by the following equation:

$$I_{out} = sign(I_1 - I_2) = \begin{cases} 1 & if\ I_1 - I_2 > 0 \\ 0 & i.o.c \end{cases} \qquad \textbf{Eq. 78}$$

Where I_1 and I_2 are the analogue input images and I_{out} is the resulting binary image. Depending on the nature of I_2, the threshold process is classified as:

- *Global threshold*, if I_2 is a spatially-invariant image, and
- *Local threshold*, if I_2 is not constant.

It is important noticing that while inputs to this operator are analogue images, which must be stored in **LAMs**, the result is a binary image, and hence must be stored in one of the **LDMs**. Consequently, this block is the natural "bridge" between the analogue and digital blocks in the Q-Eye cell. It implements the analogue to binary conversion function depicted in Figure 1.

IMAGE SHIFTER. It is a multiplexor which shifts an analogue image to any of the eight neighbors of every cell. The source image has to be inside one **LAM** and the resulting shifted image will be placed in another **LAM**. The same multiplexor can shift a binary image to any of the eight neighbours of every cell. The source must be an **LDM** and the destination another **LDM**.

IMAGE I/O BLOCK. This module performs the input-output processes of the analogue values associated to gray-level image. It sends the analogue value stored inside a **LAM** to the corresponding column **Sample-and-Hold** (**S&H**) which is connected to the 8-bit A/D converters contained in the Image I/O interface of Q-Eye during a download image process.

During image loading, this module receives an analogue value from the corresponding column **S&H** which is connected to the 8-bit D/A converters of the Image I/O interface and sends it to a **LAM.**

This module is also used to read-out a binary image stored in a **LDM**. When a binary image is downloaded via a column binary input/output block belonging to the Image Interface of the Q-Eye system, the value of every cell is codified with one bit. Therefore a binary image is converted into a digital image of 3,168 *bytes.* The opposite is also possible: digital images stored in memory outside the Q-Eye are converted into binary and loaded in one of the internal **LDMs**.

LOCAL LOGIC UNIT (LLU). It is a two-input logic block that performs a logic operation between both input binary images. Such operation is defined with a programmable truth table. Both the input and the output operands of this block are data stored in **LDMs**.

HITANDMISS OPERATOR. It is used to perform binary *morphological* operations for a given binary image. *HitAndMiss* is a well-known image-processing operation that is often used to look for particular binary patterns within an image. It is also the basic operation of *binary morphology*, and most of the binary morphological operators can be derived from it [Russ92].

The HitAndMiss operation checks whether the 3x3 neighborhood of a cell matches a specified pattern or not. This is performed in parallel for all the cells of the Q-Eye. The inputs to this block are respectively:

- the image to be analysed (that must be stored in a **LDM**) and
- the pattern to match.

The result is an image with white pixels at points whose 3x3 neighbourhood matches the input pattern. Three states can be defined as the value of a pixel: 1 (white), 0 (black) and *DNC* (*Do Not Care),* meaning that the value of that pixel is irrelevant to the matching. As a very simple example, the pattern below looks for pixels that are south-western corners inside an image.

$$\begin{bmatrix} 0 & 1 & DNC \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

ACTIVE PIXEL I/O BLOCK. This block provides a fast way for loading and downloading sparse [7] binary images to/from the Q-Eye.

The Active Pixel I/O block takes as input the binary image inside a given **LDM** and outputs the (x,y) coordinates of the active pixels. The adopted criterion is that a cell is "active" if it is white

[7] Sparse images are binary images with a reduced number of black (or white) pixels.

(logic '1'). Therefore, active pixel downloading becomes especially effective when the readout images have few white (or black) pixels.

The same operation can be used in the opposite way; isolated pixel in a binary image can be activated by providing its (x,y) coordinates from outside.

BINARY TO ANALOGUE CONVERSION (BAC). This function is used to generate an analogue image from a binary one. The input must be the binary image stored in one **LDM** and the output the **LAM** where the resulting analogue image will be stored. The generated image has the maximum analogue value for pixels that are 1 in the binary input image, and has the minimum analogue value for pixels that are 0 in the input image.

The implementation of this conversion function requires using the basic operations of some others functional blocks described above. In particular, the global references block and analogue memory block. In this last one, the writing mask function is used taking into account the binary image to be converted.

ANALOGUE TO BINARY CONVERSION (ABC). This function is carried out by the MAC block operating as a comparator in order to develop a threshold operation. The result of this operation will be stored in a binary Local Digital Memory (**LDM**).

IMAGE I/O. This block is the interface for image uploading/downloading in the complete system and will be described in Chapter 3.

3.1.2. DATAFLOW, SIGNAL FORMATS AND CONTROL OF THE Q-EYE CELL

Optical images are the primary Q-Eye inputs. Each spatial image sample is generated by a number of photons impinging into a corresponding *active* area of a photodiode. Photons are first transformed into electrical charges and then into voltages; these voltages are stored in a local analogue memory for ulterior processing – see Figure 2.

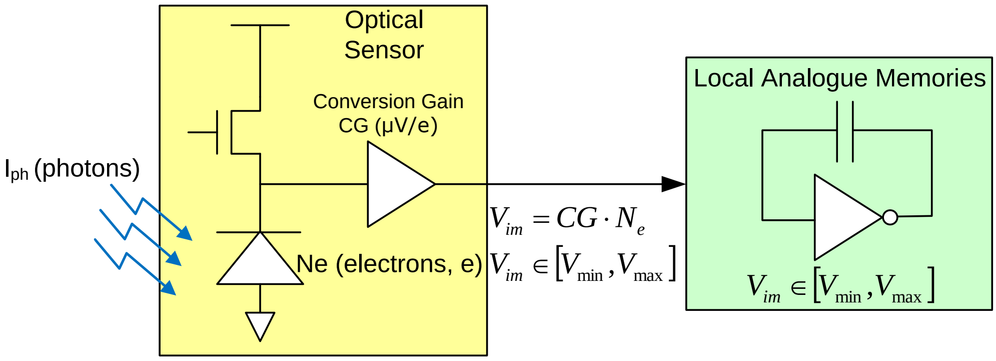


Figure 2. PROCESS OF LIGHT SENSING IN THE Q-EYE CELL.

The intensity value sensed by the cell photodiode is mapped onto an unsigned output voltage $V_{im}(x,y)$ comprised within the range $[V_{\min}, V_{\max}]$, namely $V_{im}(x,y) \in [V_{\max}, V_{\min}]$. During processing the unsigned signals stored in the **LAMs** are handled and interpreted as signed signals by making the transformation:

$$I(x,y) = V_{im}(x,y) - V_{zero}$$

where $V_{zero} = \frac{V_{\max} + V_{\min}}{2}$ is at the middle of the dynamic range $[-V_R, V_R]$ of the signals being handled during processing.

Figure 3 illustrates the signal modalities, corresponding ranges and signal transformations within the Q-Eye cell. It is worth mentioning that the sign of the signals employed for analogue processing carries itself information which may be relevant for processing, such as the direction of the movement of an object present in the scene, the direction of zero crossings, etc.

When an analog image is downloaded from the Q-Eye, it is converted to digital such that the sensing range $[V_{\min}, V_{\max}]$ and the processing range $[-V_R, V_R]$ are both linearly mapped into the positive interval $[1,255]$ by means of Analogue to Digital Converters (ADC) embedded in the external image interface. Likewise, when a digital gray-level image is loaded to the Q-Eye, the interval $[1,255]$ is linearly mapped into the sensing range $[V_{\min}, V_{\max}]$ by means of Digital to Analogue Converters (DAC) embedded in the I/O Image Interface of system, see section 4.1 and section 4.2 of Chapter 4.

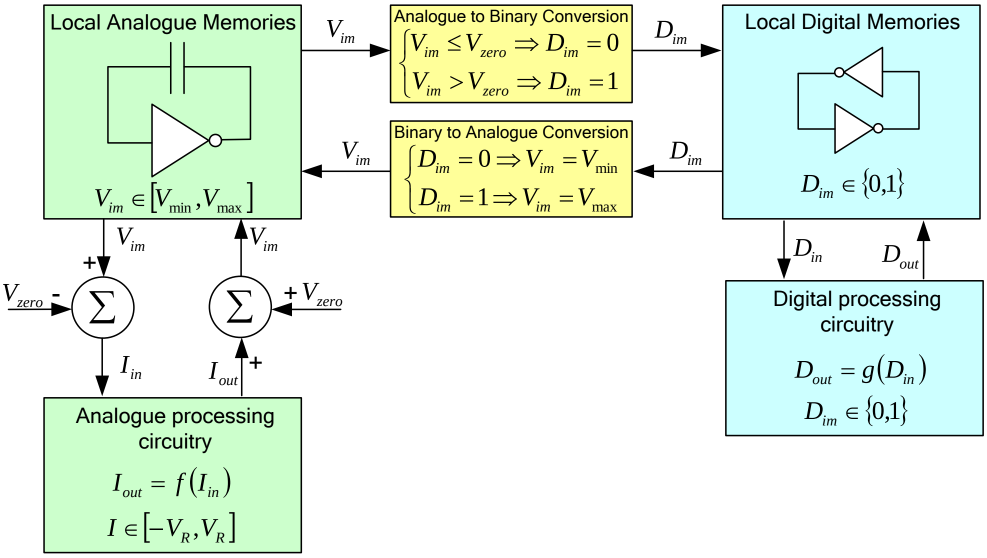

Local Analogue Memories
$V_{im} \in [V_{\min}, V_{\max}]$
V_{im}
Analogue to Binary Conversion
$V_{im} \le V_{zero} \Rightarrow D_{im} = 0$
$V_{im} > V_{zero} \Rightarrow D_{im} = 1$
D_{im}
Local Digital Memories
$D_{im} \in \{0,1\}$
Binary to Analogue Conversion
$D_{im} = 0 \Rightarrow V_{im} = V_{\min}$
$D_{im} = 1 \Rightarrow V_{im} = V_{\max}$
V_{zero} -
+
Σ
+ V_{zero}
I_{in}
I_{out}
D_{in}
D_{out}
Digital processing circuitry
$D_{out} = g(D_{in})$
$D_{im} \in \{0,1\}$
Analogue processing circuitry
$I_{out} = f(I_{in})$
$I \in [-V_R, V_R]$

Figure 3. SIGNAL MODALITIES AND RANGES IN THE Q-EYE CELL.

As already mentioned, the Q-Eye handles both analog and binary images. Note in Figure 3 that the analogue-to-binary and binary-to-analogue conversions implement a correspondence between the "0" binary level and the reference voltages $-V_R$ or $V_{\min}$ and the "1" binary level and analogue reference V_R or $V_{\max}$. During loading and downloading of binary images, the external image interface of Q-Eye writes directly the binary values "0" and "1" in **LDMs**.

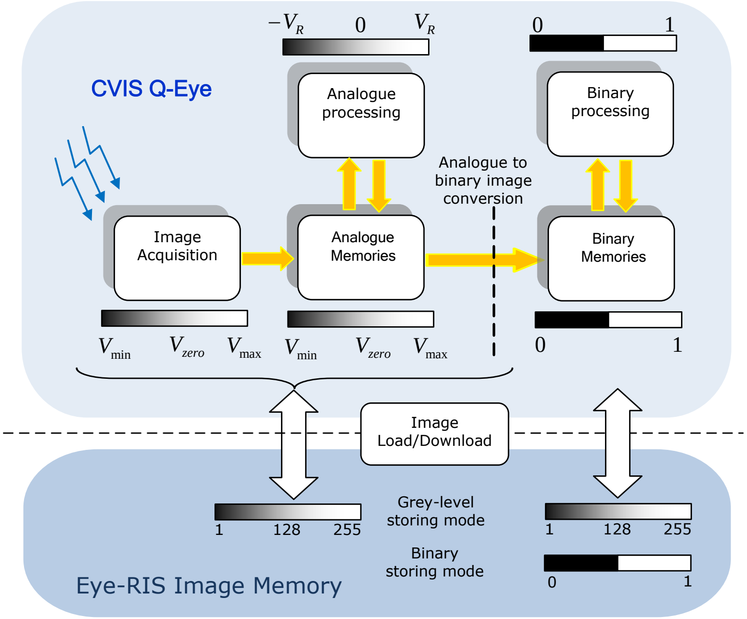


Figure 4. DATA FLOW AND IMAGE CODIFICATION IN THE Q-EYE.

Figure 4 summarizes the data flow in the Q-Eye cell and the data codification used in the main processes involved in the operation of the cell, namely: image acquisition, analogue storing, analogue processing, binary storing and binary processing.

Proper control is needed to exploit the parallel processing capabilities of the Q-Eye. Control of cell array has been designed in the Q-Eye system following the principles of Single Instruction Multiple Data architecture (SIMD) [Zara11], [Dunc90]. The cell array processes in parallel one image applying the same operation to each pixel. The operation is determined by the control unit of system.

Several control signals arrive to each functional block in the Q-Eye cell for configuring and controlling the operation of the corresponding mixed-signal circuit, see Figure 5 and Figure 6.

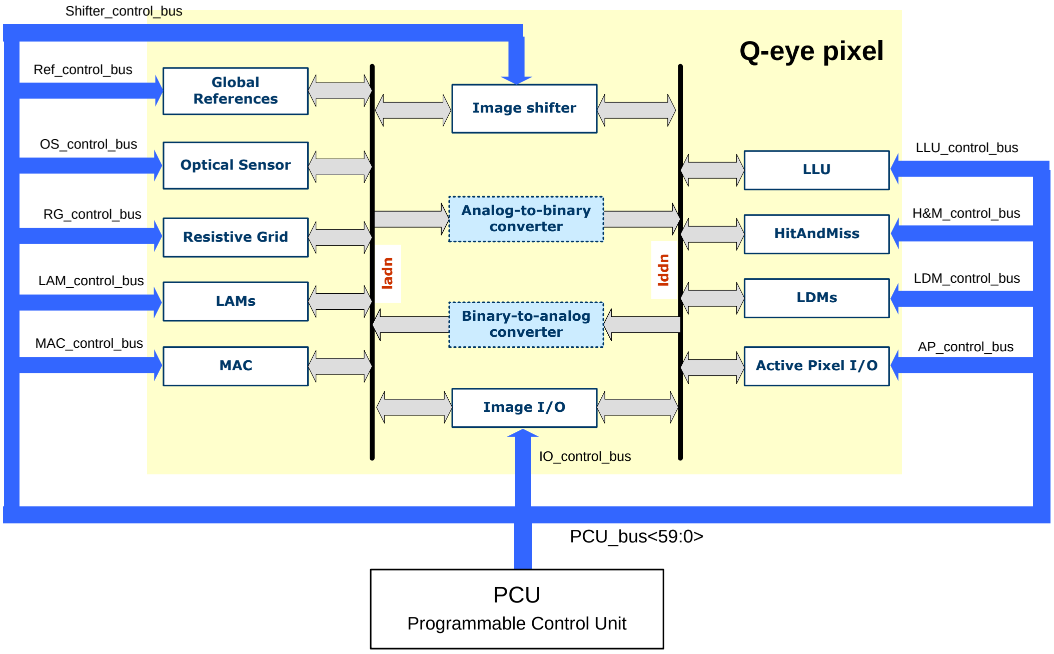


Figure 5. Q-EYE CELL CONTROL.

The control of system is managed by the Programmable Control Unit digital block named **PCU**. This block controls the processing array applying a control signal vector or instruction which is stored in a register called **PBCTRL** included in **PCU** block. The **PBCTRL** register has a width of 60 bits. Each bit is associated to a specific control signal of the Q-Eye cell.

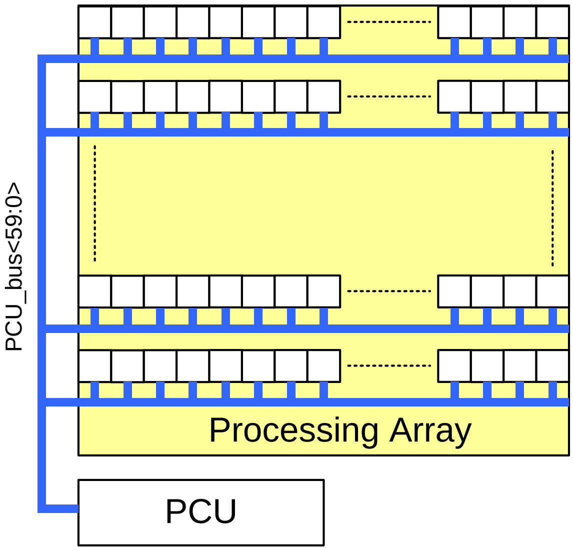


Figure 6. SIMD ARCHITECTURE.

Instructions are applied simultaneously to all cells in the array. The implementation of one basic operation in the processing array (for instance, read one analog memory, see Table 2) implies the application of a sequence of these instructions. Further details are given in Chapter 4, section 4.1.2.

3.1.3. RATIONALE FOR SELECTION OF THE Q-EYE OPERATORS

Basic *early vision* tasks include image convolutions, spatial-temporal filtering with programmable bandwidth, extraction of image features such as centroids, mean values, pixel coordinates… [Gonz92]. Early vision is aimed to reduce the amount of data associated to images before sending these data for further digital processing [Toma06] [Shi05]. The rationale is that images do contain much more data than information. In other words the same information can be retrieved with a reduced set of the data captured by the optical sensors and defining image frames.

Reducing the data is far from trivial because it must be done without degrading the underlying information. In the more general case, linear, non-linear, static, dynamics and statistical operations are required to cover "all" possible early vision task. However, in practice a tradeoff between the in-pixel circuitry and the pixel pitch must be made [Rodr03]. In the case of the Q-Eye this tradeoff is addressed on the basis on the experience accumulated with previous generations of ACE chips [Espe94a, b] [Rodr93] [Carm01] [Liña02a, b] [Carm03] [Rodr04] [Liña04] and the request posed by practical applications [Ohta08] [Deva08].

The available operations can be classified within three categories, namely:

- Point-wise operations:
  - Storing (gray-level and binary images)
  - Addition
  - Subtraction
  - Multiplication by a constant
  - Threshold operation
  - Combinational operations (binary images)
- Local operators:
  - Low-pass Gaussian filters, implemented by diffusion using a resistive grid.
  - Hit-And-Miss binary operations considering 3x3 neighbourhood.
  - Spatial convolutions, combination of arithmetic operations (addition, subtraction and multiplication by a constant) and shifting operation.
- Shifting operation (gray-level and binary images)
- Inter-frame processing:
  - Addition
  - Subtraction
  - Combinational operations

Point-wise operators do only depend on the value of the pixels to which these operators are applied. Regarding local operators, they require as well the values of the pixels located within a certain neighborhood of the processed pixel. Point-wise and local operators are used for *intra-frame* processing. Memory elements are used to store several consecutive frames and thus implement inter-frame operations. For instance, in motion detection applications it is common to compute a difference image by subtracting the pixel value of a previous frame from the value in the current frame.

The early-processing stage comprises image enhancement and restoration followed by processing to extract image features such as object locations, shapes, edges, etc. During image enhancement and restoration, the input images are transformed in some predefine sense to simplify the extraction of features to be used during the post-processing stage. Typical functions used in the enhancement and restoration process include:

- the gray-level transformations to improve the image contrast,
- smoothing or averaging *spatial filters* to reduce the spatial noise,

- sharpening spatial filters to highlight fine image details,
- temporal averaging to reduce the temporal noise,
- non-linear median filters to eliminate the impulse noise, etc.

The gray-level transformations can be implemented by using point-wise operations. Linear filters can be implemented through the resistive grid and arithmetic operations or spatial convolutions. The Q-Eye cell includes a non-linear operation, the *threshold operation*, which permits to implement non-linear filters in the processing array, for example *Rank-values* filters, like the median filter, or adaptive filters.

The extraction of image features comprises many complementary tasks. *Image segmentation* is a typical processing which subdivides an image into its constituent regions or objects. The level to which the subdivision is carried depends on the problem being solved. The segmentation should stop when the objects of interest in an application have been isolated. Segmentation algorithms generally are based on one of two basic properties of intensity values:

- discontinuity and
- similarity.

In the first category, the approach is to partition an image based on abrupt changes in intensity. The detection of intensity discontinuities (points, lines, edges) is developed applying high-pass filters. The principal approaches in the second category are based on partitioning an image in regions that are similar according to a set of predefined criteria. Thresholding, region growing, and region splitting and merging are examples of tasks in this category. Several morphological operators can be implemented with Hit-And-Miss operations and combinational logic.

The discontinuities detection functions and threshold function together with morphological operators constitute the early-processing level in the segmentation algorithms and extract the features interesting for a determined application. These features are described in a binary image which contains a reduced set of data compared with the amount of data contained in the original gray-level image. This compression of data implies a faster communication between the Q-Eye and the microprocessor which develop the high-level functions, for example, the post-processing segmentation and classification algorithms. This is the reason why the Q-Eye includes a fast interface for loading and downloading sparse binary images.

Therefore, characteristic algorithms of early-processing level such as filtering, threshold, edge detection, morphological transformations, binary operations, arithmetic operations, etc. are implemented by applying a group of simple operators on every pixel of the image. The early-processing algorithms are characterized by being simples and applied to a vast amount of data. In this case, the computation requirements are very high being optimal a parallel processing architecture consists of a high number of elementary and simple processors, similar to the approach implemented in the Q-Eye.

The architecture of elementary processor must be the simpler simple as possible in order to optimize power consumption and area. There is a trade-off between the number of operators implemented in the elementary processor and their power consumption and area. For instance, the Q-Eye cell pitch limits the maximum number of control signals which can be driven to each processor.

3.1.4. ILLUSTRATING DATA REDUCTION THROUGH THE Q-EYE

The Q-Eye plays the fundamental role of reducing the amount of data at the early stages of the vision processing chain, illustrated in Figure 7. This figure describes the overall architectural target with reference to a conceptual representation of a vision processing chain [Russ92]. It includes several processing steps and shows that the amount of data decreases as information travels along the chain, namely:

- At initial steps the number of data is huge and many of the data are redundant and hence useless to the purposes of reaction prompting.
- As information flows across the processing chain and abstract features are extracted from the incoming images, the number of data decreases.

In conventional vision architectures the border between sensors and processors is placed at a point where the amount of data is large. However, in the Eye-RIS architecture this border is located at a point where the amount of data is small. Assume for illustration purposes that we target to tracking objects moving at 40m/sec into a scene. It requires capturing and analyzing images at 2000F/s rate. At the outcome of the capture/analyze process the only pertinent data is the predicted position of the objects. This is actually the only information driven to the digital processor. The Q-Eye plays the fundamental role of extracting this information by executing the following tasks within the sensor:

i) Image acquisition;
ii) low-pass filtering;
iii) activity detection;
iv) motion estimation;
v) object tracking;
vi) loop control, and
vii) position prediction.

and thus shifting the border between sensors and processors to a point where only a few data are left – see Figure 7 and Figure 8.

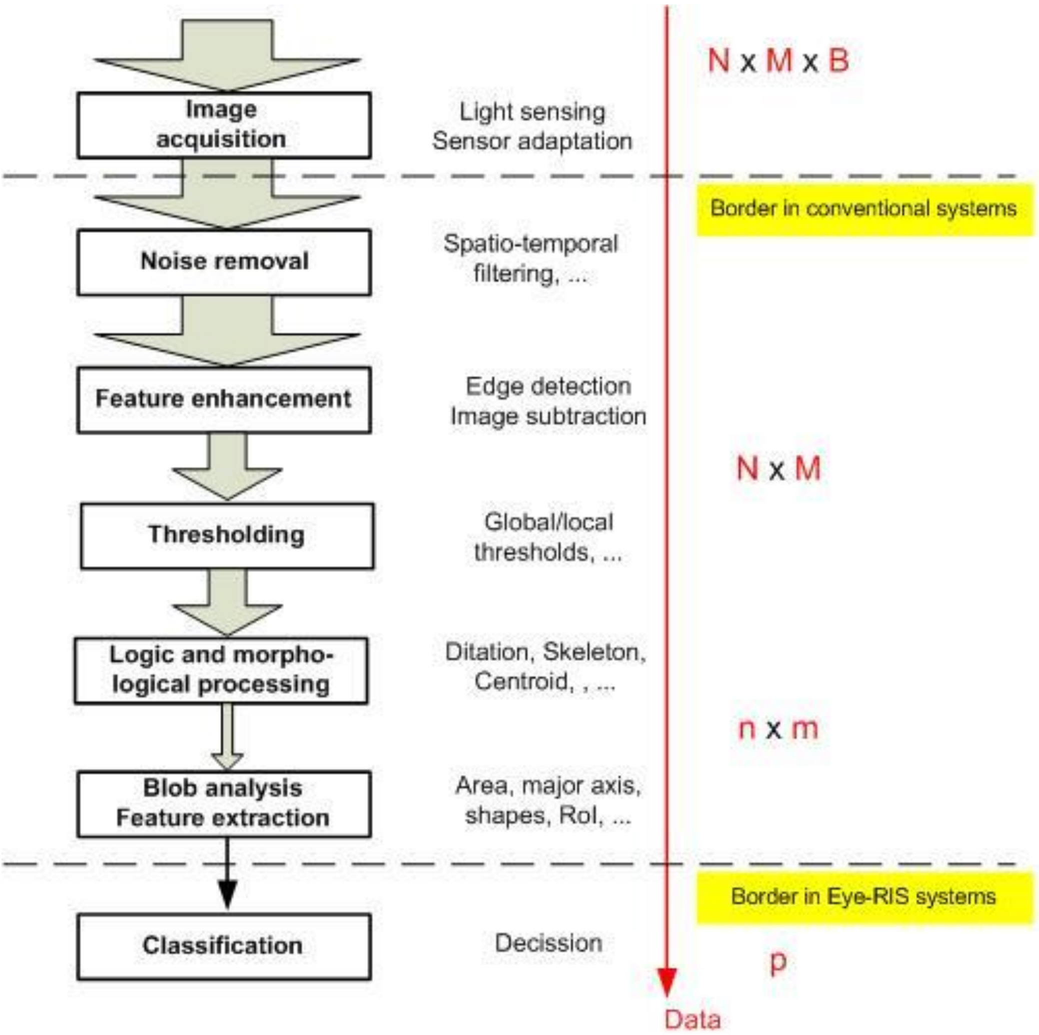

N x M x B
Image acquisition
Light sensing
Sensor adaptation
Border in conventional systems
Noise removal
Spatio-temporal filtering, ...
Feature enhancement
Edge detection
Image subtraction
N x M
Thresholding
Global/local thresholds, ...
Logic and morpho-logical processing
Ditation, Skeleton, Centroid, , ...
n x m
Blob analysis
Feature extraction
Area, major axis, shapes, RoI, ...
Border in Eye-RIS systems
Classification
Decission
p
Data

Figure 7. PROCESSING CHAIN OF VISION. AS THE DATA EVOLVE FROM THE SENSOR (RAW DATA), THE AMOUNT OF DATA DECREASES AND THE ABSTRACTION LEVEL INCREASES. *M* REPRESENTS THE NUMBER OF COLUMNS, *N* THE NUMBER OF ROWS AND *B* THE NUMBER OF BITS USED PER PIXEL DATA: $n < N$; $m < M$ AND $p < (n, m)$.

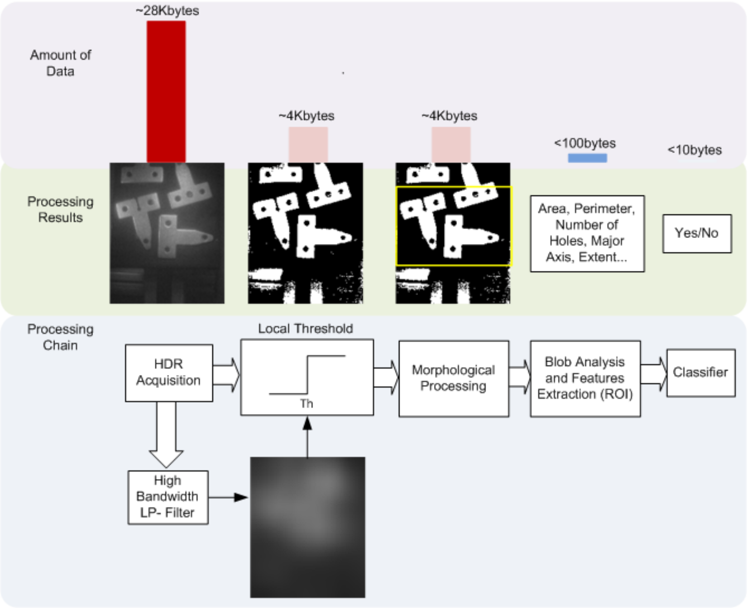


Figure 8. ILLUSTRATION OF THE PROGRESSIVE DATA REDUCTION OF DATA ALONG THE VISION PROCESSING CHAIN AS IT ACTUALLY HAPPENS IN THE EYE-RIS. ALL STEPS OF THE PROCESSING CHAIN ABOVE BUT THE LAST ONE ARE COMPLETED IN THE Q-EYE SENSORY-PROCESSING FRONT-END. THUS, THE DATA SET DELIVERED FOR PROCESSING BY THE HOST DIGITAL PROCESSOR IS QUITE SMALL.

Figure 8 shows major steps of the processing chain implemented within the Eye-RIS_v2.1 to find defective parts in a production line. Eye-RIS_v2.1 identifies defective parts based on feature analysis instead of brute force pattern matching. It enables speed improvements (number of pieces per second) of several orders of magnitude as compared to conventional systems [COGNEX]. The figure illustrates the progressive reduction of data along the processing chain. Out of the some 26kbyte raw data acquired by the sensors at the Q-Eye front-end, only some 100byte remain after pre-processing and are actually coded in digital domain and sent to the post-processing stage. Such reduction, together with the intensive parallel processing performed at the front-end is instrumental to achieve this data reduction and hence the overall efficiency enhancement in the completion of the part finding task.

3.2. CIRCUIT IMPLEMENTATION OF THE Q-EYE CELL OPERATORS

This section describes the circuitry employed for the different blocks in the Q-Eye cell and presents the equations that support the ideal behavior of such circuitry. Electrical analyses included in this section are first-order for all the cases. The impact of circuit errors will be discussed in Section 3.3.

3.2.1. GLOBAL ANALOGUE REFERENCES

Correct operation of the Q-Eye mixed-signal blocks require proper biasing. Also re-configurability considerations call for a large variety of biasing conditions. Table 1 describes the main reference voltages indicating the functional block which uses the references.

REFERENCE	DESCRIPTION	FUNCTIONAL BLOCK
V_{pch}	Precharge voltage of pixel	Optical sensor
V_{hpch}	Control signal of reset transistor in pixel	Optical sensor
V_{hold}	Reference used during hold configuration in analogue memories	LAM block, Global Reference block
V_{offset}	Reference to define the offset in MAC operation.	MAC block, Global Reference block
V_{misc}	General purpose reference	Global Reference block

Table 1. **Analogue references of processing cell.**

The Global Reference block consists of an analogue multiplexer that connects the references V_{hold}, V_{offset} and V_{misc} to the local analogue node **Iadn** in function to the control signals. Figure 9 depicts the architecture. Global references have been represented in red and the control signals in blue. This notation will be maintained throughout the chapter.

The general purpose reference V_{misc} is a useful programmable reference which allows loading in parallel way a gray level uniform image to the processing array. Each analogue reference described in Table 1 is generated on-chip in the *Q-Eye* system by a programmable *DAC* through a configuration register accessible through the external programming interface.

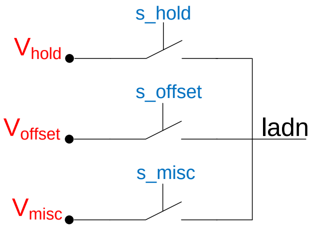


Figure 9. **GLOBAL ANALOGUE REFERENCE BLOCK.**

3.2.2. DATA STORAGE AND TRANSFER

As already mentioned, rach Q-Eye pixel embeds 9 **LAMs** and 9 **LDMs**. The topology of these local memories will be described in following sections.

3.2.2.1. ANALOGUE MEMORIES

Figure 10 depicts the architecture of the **LAM** block. This block consists of two banks of analogue memories. One bank has seven (7) memories and it is called LAMs bank. The other bank has two (2) analogue memories and it is named Auxiliary LAMs bank. Each memory bank consists of a **S&H** (Sample and Hold) circuit containing *N* (either seven or two) capacitors associated to a common, shared, sensing amplifier. The S&H topology has been chosen to yield:

- non-destructive recovery of the stored signal;
- reduced harmonic distortion caused by signal-dependent switching errors; and
- insensitiveness to the sensing amplifier offset [Greg94].

The basic operations associated to a memory bank are *writing*, *reading* and *holding* respectively. During the writing phase, the capacitor C_m is charged by the voltage $\left(V_{data} - V_{Nq}\right)$:

$$Q_{C_m}(n) = C_m\left(V_{data} - V_{Nq}(n)\right) \qquad \textbf{Eq. 79}$$

In the ***reading phase***, the charge of capacitor C_m is:

$$Q_{C_m}(n+1) = C_m\left(V_{Nout}(n+1) - V_{Nq}(n+1)\right) \qquad \textbf{Eq. 80}$$

Because the bottom plate of C_m is isolated during the reading phase, the charge in the capacitor does not change, $Q_{C_m}(n) = Q_{C_m}(n+1)$, and hence the output voltage in the *Nout* node is given by:

$$V_{Nout}(n+1) = V_{data} + \left[V_{Nq}(n+1) - V_{Nq}(n)\right] \qquad \textbf{Eq. 81}$$

Considering a sensing amplifier with infinite gain obtains $V_{Nq}(n+1) = V_{Nq}(n)$, meaning that the data can be recovered during reading:

$$V_{Nout}(n+1) = V_{data} \qquad \textbf{Eq. 82}$$

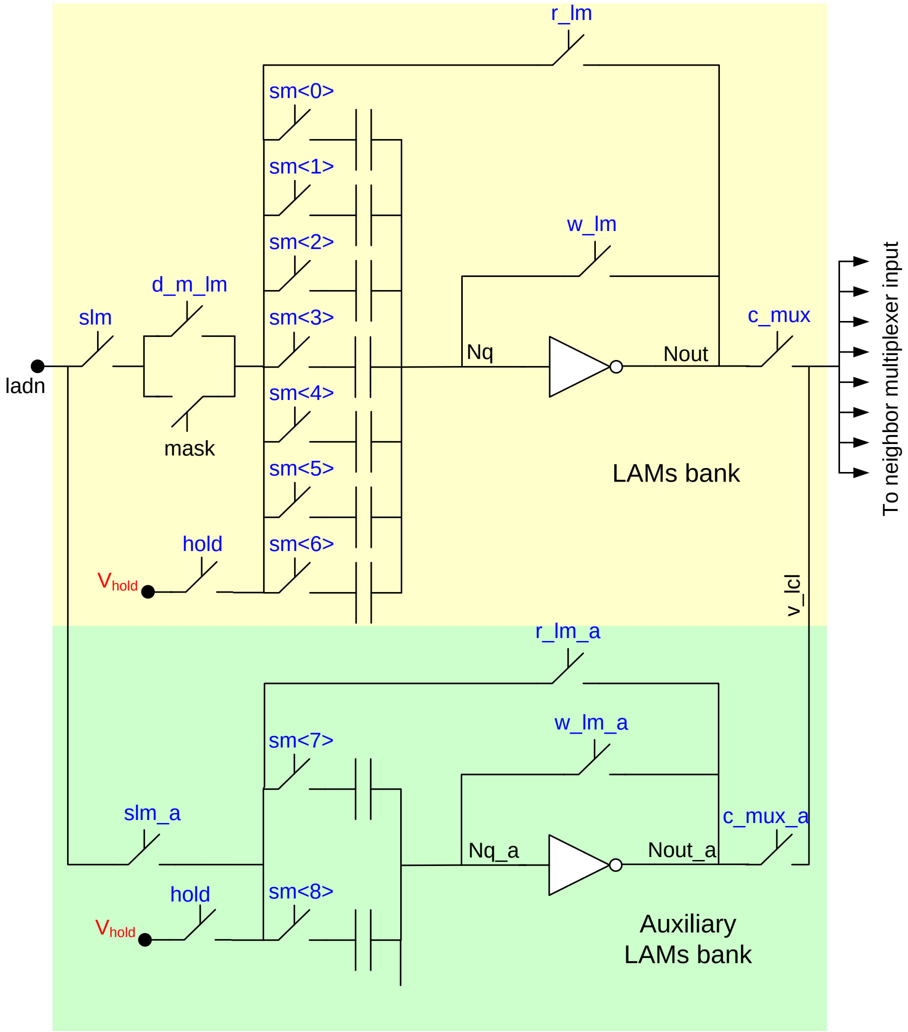


Figure 10. LAM BLOCK ARCHITECTURE.

In order to optimize power consumption and area, the sensing amplifier has been implemented by a telescope cascode inverter [Malo01] and the capacitors have been developed with MOS transistors (MOS capacitors) – see Figure 11.

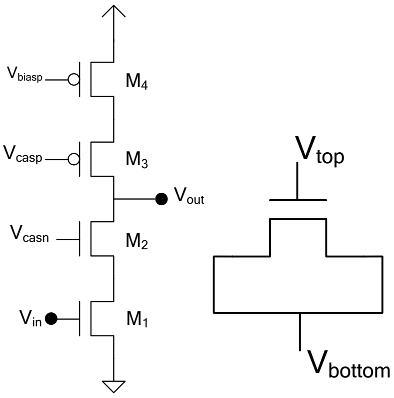


Figure 11. TELESCOPE CASCODE INVERTER.

Note from Figure 10 that writing and reading operation can be made conditional through the usage of the *writing-reading* mask. The configuration signal ***d_m_lm*** enables the mask option. In this case, the writing or reading operation is masked depending of a binary image stored in a particular local digital memory. To optimize the Q-Eye pixel area, the binary mask used in the

LAMs bank is defined by the same local digital memory which stores the mask for the resistive grid.

The instruction sequence to develop the operations related with the memory bank is presented in Figure 12 and Table 2. The control of the Auxiliary LAMs bank is similar to the LAMs bank control.

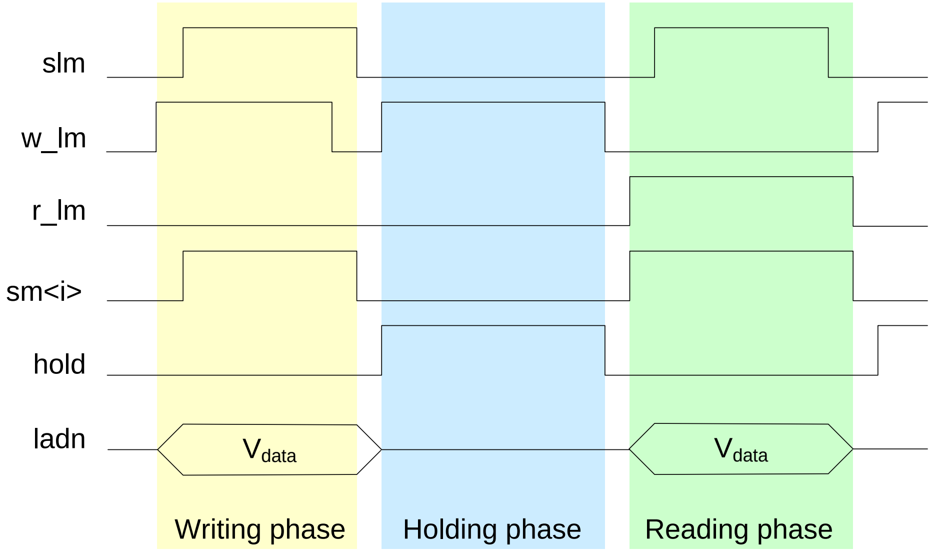


Figure 12. **INSTRUCTIONS FOR ANALOGUE MEMORY OPERATION.**

Nº Instruction	**slm**	**w_lm**	**r_lm**	**sm<i>**	**hold**	**ladn**	**Operation**
1	0	0	0	0	0	-	-
2	0	1	0	0	0	Vdata	Writing
3	1	1	0	1	0	Vdata	Writing
4	1	0	0	1	0	Vdata	Writing
5	0	0	0	0	0	-	-
6	0	1	0	0	1	-	Holding
7	0	0	0	0	0	-	-
8	0	0	1	1	0	-	Reading
9	1	0	1	1	0	Vdata	Reading
10	0	0	1	1	0		
11	0	0	0	0	0		

Table 2. Analogue memory instructions.

During the reading phase, the memory banks can drive the stored value in the local data node **ladn** or send the data to the analogue multiplexer, activating the slm, slm_a control signals or c_mux, c_mux_a signals respectively - see Figure 10. In the holding phase, the analogue memory bank is configured to reduce the effect of leakage currents in the MOS capacitors controlling the voltage of critical nodes.

3.2.2.2. DIGITAL MEMORIES

Figure 13 depicts the architecture of the **LDM** block. It is composed by 9 latches with the schematic of Figure 14.

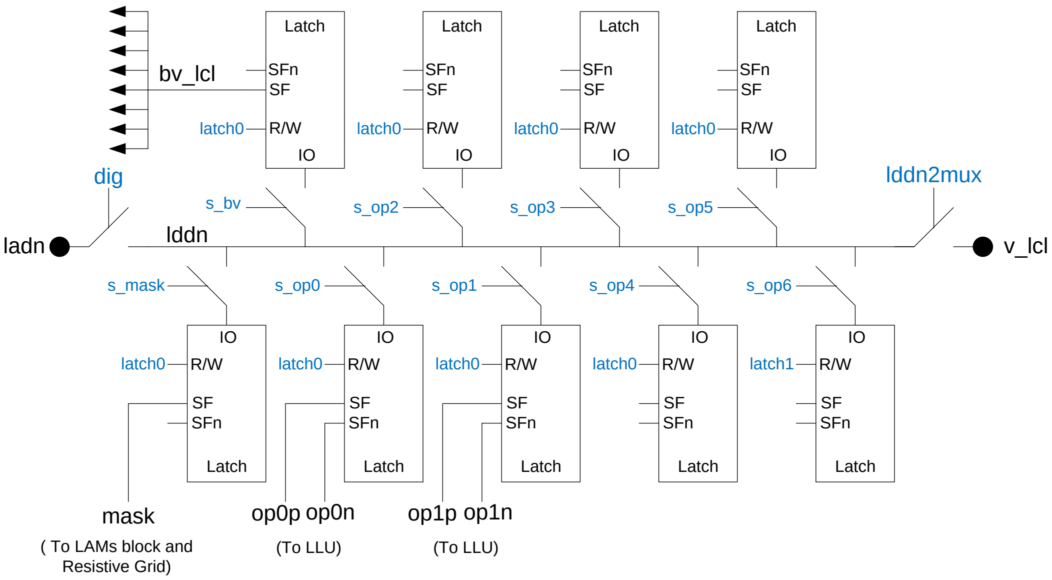


Figure 13. **LDM ARCHITECTURE.**

The stored data can be fixed in the digital data node **Iddn** through the selection control signals (s_bv, s_mask, s_op0, s_op1, s_op2, s_op3, s_op4, s_op5, s_op6). There are 3 memories whose data has a specific function in the Q-Eye processing cell. The mask memory, with selection control signal s_mask, defines the mask value in the LAMs block and Resistive Grid block. The LLU memories, with selection control signals s_op0 and s_op1, are the operands in the LLU block. Finally, the HitAndMiss memory, with selection control signal s_bv, sends the data to the neighbors through the interconnection bus during the Hit And Miss operation. The rest of memories have a general purpose.

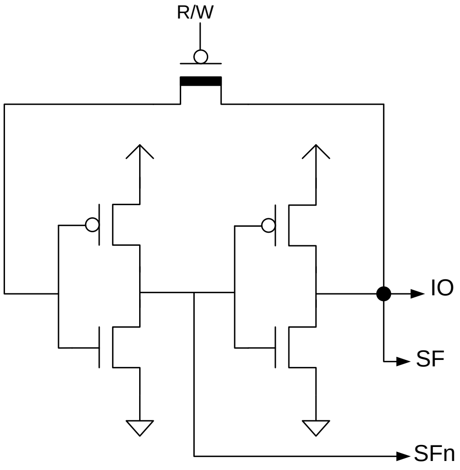


Figure 14. **LATCH TOPOLOGY.**

Figure 15 and Table 3 show the instruction sequence to develop writing and reading process for the digital memories. A binary data can be transferred between two digital memories, through the digital data node **Iddn**, configuring one in reading operation and the other in writing phase. All digital memories are controlled by the same feedback control signal R/W (latch0 in Figure 13) except one memory which is controlled by a different feedback control signal latch1. This memory has the selection control signal s_op6 and it is named auxiliary memory. Therefore, there are two banks of digital memories, each one controlled by the signals latch0 and latch1 respectively. The data transfer between two digital memories controlled by the same feedback signal latch0 must be carried out through the auxiliary memory.

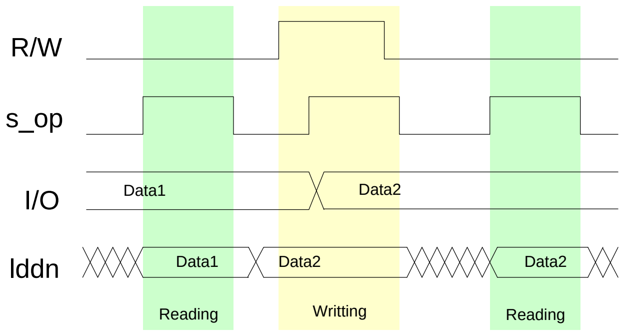


Figure 15. CONTROL SIGNALS FOR A DIGITAL MEMORY.

Nº Instruction	**R/W**	**s_op**	**Iddn**	**Operation**
1	0	0	-	-
2	0	1	Data1	Reading
3	0	0	-	-
4	0	0	Data2	Writing
5	1	0	Data2	Writing
6	0	1	Data2	Writing
7	0	0	Data2	Writing
8	0	0	-	-
9	0	1	Data2	Reading
10	0	0	-	-

Table 3. Digital memory instruction.

The analogue data node **Iadn** and the digital data node **Iddn** can be connected by the control signal **dig** in Figure 13.

A digital data fixed in the digital node **Iddn** can be sent to the analogue multiplexer with the control signal Iddn2mux for a shifting operation.

The digital to analogue conversion is used to generate an analogue image from a binary one. The input must be the binary image stored in one LDM and the output the LAM where the resulting analogue image will be stored. The generated image has the maximum analogue value for pixels that are 1 in the binary input image, and has the minimum analogue value for pixels that are 0 in the input image.

The implementation of this conversion function requires storing the binary image to convert in the digital mask memory. This binary image conditions the writing operation in the analogue memory during the writing process of a global reference equal to the maximum analogue value (V_{max}). Initially, the analogue memory was written with the minimum analogue value (V_{min}).

3.2.3. IMAGE SHIFTING

The Image Shifter block is a multiplexer which connects the local node v_lcl (see Figure 10) for one of the neighboring pixels to the data node **Iadn**. Its topology is depicted in Figure 16.

The local node v_lcl is driven to the neighboring pixels through the interconnection bus v_lcl<7:0> following the orientation described in Figure 17. The shifting direction is configured by the control bus pm<7:0>. During the shifting process of gray level image, one bank of analogue memories is connected to the local node v_lcl in reading configuration and the other bank of analogue memories is connected to the local analogue data node **Iadn** in writing configuration. The number of gray level shifting operations is limited by the accumulation of errors associated to analogue memorization. These errors are different pixel to pixel because of mismatch effect, resulting in a degradation of image quality when the number of iterations is high. Error consequences will be addressed in Section 3.3.1.

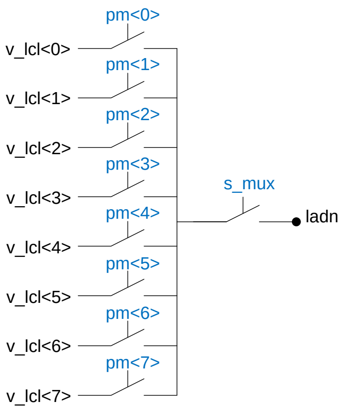


Figure 16. IMAGE SHIFTER ARCHITECTURE.

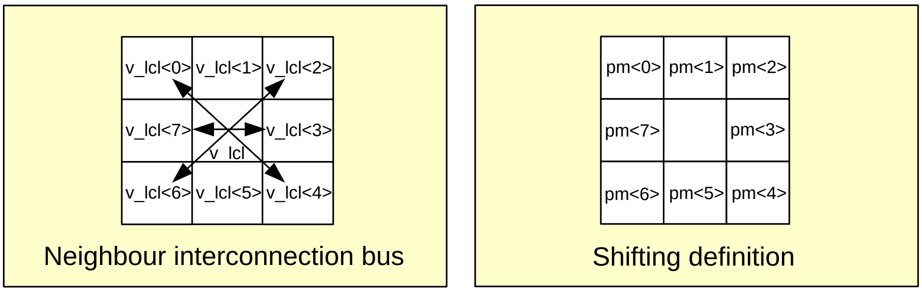


Figure 17. SHIFTING INTERCONNECTION.

The shifting of a binary image is carried out with this multiplexer connecting the local digital node **Iddn** to the local node v_lcl of Figure 13. Thus, the data is sent to a specific neighbor defined by the control bus pm<7:0>. With this implemented architecture (**Iddn** is connected to v_lcl), the digital data cannot be received by a digital memory but by an analogue one, for instance, the auxiliary analogue memory bank. After the shifting process, the data is transferred from the auxiliary analogue memory to a digital memory.

Figure 18 is an alternative architecture for binary image shifting. In this case, one digital memory can be connected in reading configuration to the local node v_lcl and another one will be connected to the **Iadn** node in the neighbor pixel in writing configuration.

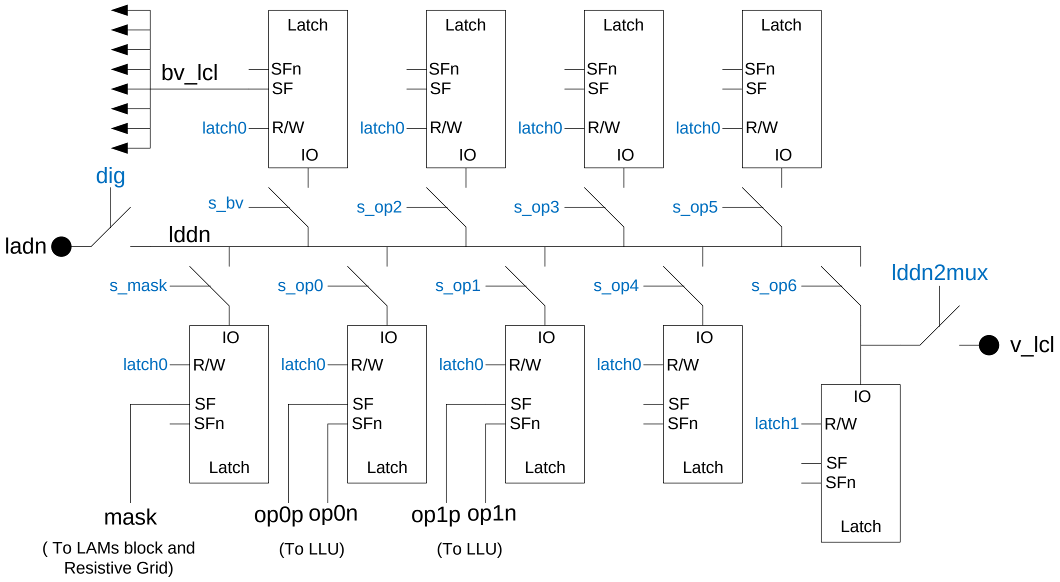


Figure 18. ENHANCED TOPOLOGY TO IMPLEMENT THE BINARY SHIFTING.

3.2.4. OPTICAL SENSOR

Figure 19 shows the optical sensor topology, consisting of:

- a p-well photodiode with active area of 3x3µm^2 rendering 1.0/V(lux·sec) sensitivity for green light (λ=555nm),
- a reset NMOS transistor to pre-charge the integration node, denoted by Floating Diffusion (FD) node, read section 2.2.3 of Chapter 2, where the photocurrent gets integrated,
- a PMOS source follower buffer with body eliminated to obtain a better characteristic of linearity and,
- a NMOS selection transistor.

Note that the electrical operation of the photo-diode and hence the process of light capture is controlled by two programmable voltages V_{pch} and V_{hpch} involved in the electrical operation of the reset transistor M_R. There are two possible programming modes for these voltages, namely:

- V_{hpch} can be configured to either switch between the power supplies (VDD, VSS) or to sweep from VDD to VSS passing through several intermediate analogue levels during the sensing process. The latter case yields *linear acquisition*, while the former yields a *compressive acquisition* procedure that enables High Dynamic Range (HDR) images be captured.
- Regarding V_{pch}. If the reference V_{pch} value is such that the M_R transistor operates in strong inversion during the complete reset process, a *hard reset* has been applied to the photodiode. Conversely, if the reference V_{pch} value is such that the M_R transistor operates in weak inversion region at the end of reset process, a *soft reset* has been applied to the photodiode. The main difference between both reset modes is the noise contribution of reset transistor [Tian01] (Section 3.3.4.2)

The operation mode of Q-Eye is *global shutter*. At the same time, all pixels in the Q-Eye transfer the photo-generated data to the local analogue bus **Iadn** to store this voltage in the analogue memories. This transfer is impacted by the threshold voltage of the buffer transistor. The Fixed Pattern Noise (FPN) caused by random mismatches in the threshold voltage of this buffer is corrected by implementing a Correlated Double Sampling (CDS) operation using the **LAMs** and the **MAC** blocks of the Q-Eye cell (Figure 1).

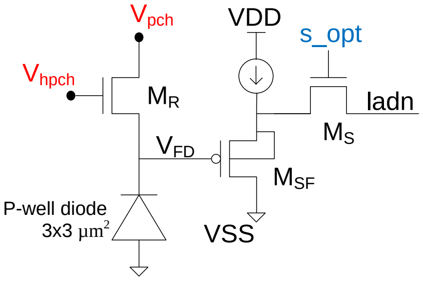


Figure 19. OPTICAL SENSOR TOPOLOGY.

3.2.4.1. LINEAR SENSING PROCESS

Figure 20 depicts the control signal of Q-Eye pixel during a linear sensing process. Two different phases can be defined for each exposure, namely:

- Reset,
- Exposure

Voltages in the **FD** node and the **Iadn** node are read at the end of each phase. During the reset reading sub-phase, the value at the output of source follower and **Iadn** node is:

$$V_{reset} = V_{reset}^{FD} - \Delta V_{SF} \qquad \textbf{Eq. 83}$$

Where, ΔV_{SF} is the shifting introduced by the source follower:

$$\Delta V_{SF} = V_T(V_{SB}) + \sqrt{\frac{I_B}{C_{ox} \cdot \beta_{sat} \cdot W_{SF} \cdot L_{SF}}}$$ **Eq. 84**

Since $V_{SB} = 0$, there is not body effect and the threshold voltage coincides with V_{T0}, which means that ΔV_{SF} does not depend on the voltage value at the source of M$_{SF}$. This is so done to preclude signal-dependent errors to appear, namely gain errors and non-linearity errors.

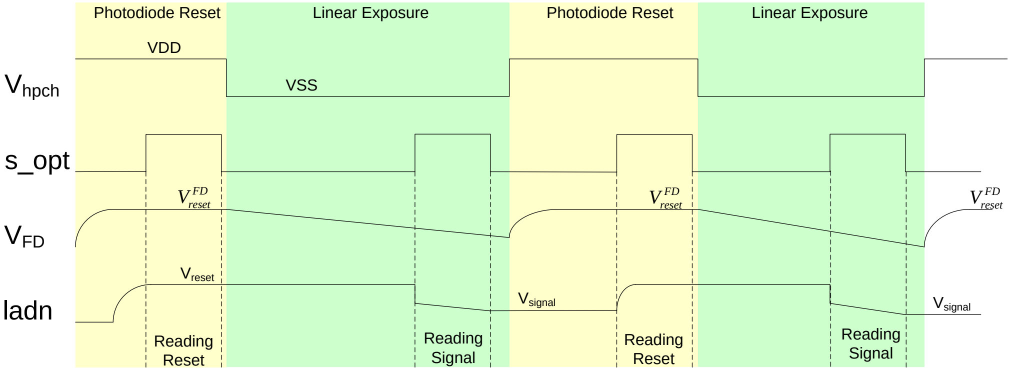


Figure 20. CONTROL SIGNALS FOR LINEAR EXPOSURE MODE.

On the other hand, during the exposure reading sub-phase, the signal value at the output of source follower and **Iadn** node is:

$$V_{signal} = V_{signal}^{FD} - \Delta V_{SF}$$ **Eq. 85**

The two voltage values read this way are stored in two different **LAM** positions and subtracted by the *MAC* block, to obtain the pixel value,

$$V_{im} = V_{reset} - V_{signal} = V_{reset}^{FD} - V_{signal}^{FD}$$ **Eq. 86**

Note that ΔV_{SF} has been eliminated because it is signal-independent and has hence the same value during both phases.

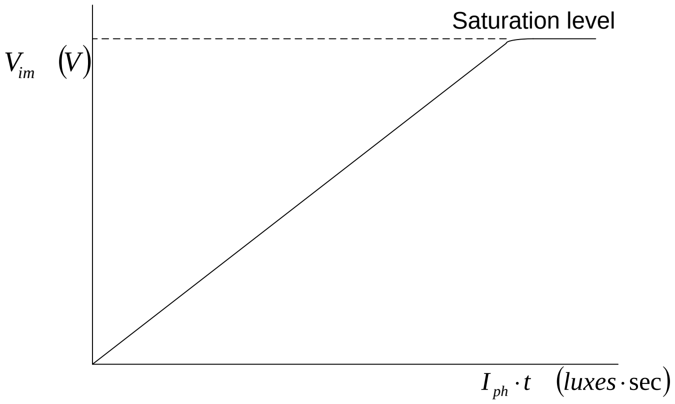


Figure 21. LINEAR SENSING MODE CHARACTERISTIC.

Figure 21 shows the pixel voltage as a function of the illumination. This figure is readily obtained by noticing that the signal in the equation above is built by the temporal evolution of FD node voltage during exposure, namely:

$$V_{signal}^{FD}(t) = V_{reset}^{FD} - S \cdot I_{ll} \cdot t$$ **Eq. 87**

Where S is the responsivity of Q-Eye pixel expressed in $\frac{V}{lux \cdot \sec}$, I_{ll} is the illuminance at the sensor plane given in $luxes$ and t is the integration time. Combining equations Eq. 86 and Eq.87,

$$V_{im} = S \cdot I_{ll} \cdot t \qquad \textbf{Eq. 88}$$

3.2.4.2. Compressive, High-Dynamic Range Sensing Process

The main difference between this mode and the previous one is that the reset transistor is not biased OFF at the beginning of the exposure phase. Instead, as Figure 22 shows, its gate is driven during the exposure time by several programmed intermediate levels between power supplies (VDD, VSS). This technique is called *stepped reset* [Gama02].

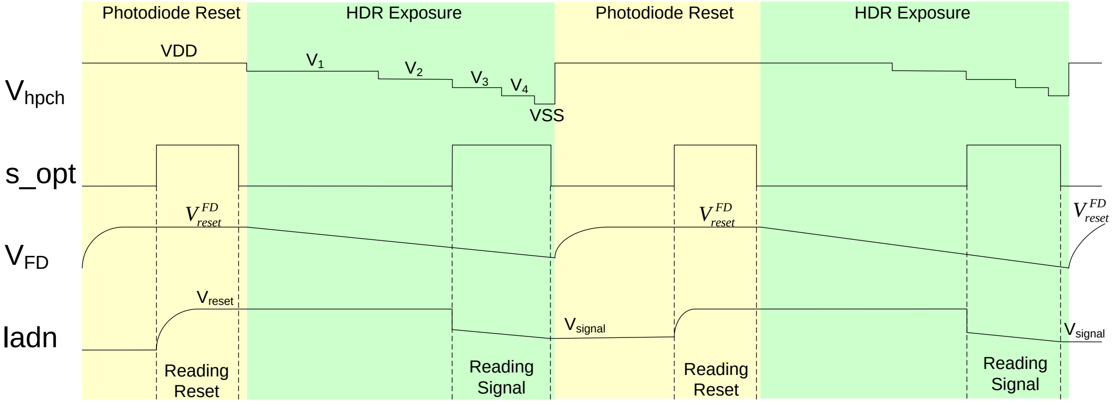


Figure 22. Control signals for HDR exposure mode.

This way a *piece-wise linear* compressive acquisition characteristic gets implemented whose shape is controlled by programming the intermediate analogue levels and their corresponding time intervals – see Figure 23. Considering four intermediate levels, the characteristic response of pixel for the HDR sensing mode is described by the equation:

$$V_{im} = \begin{cases} S \cdot I_{ph} \cdot t & I_{ph}^{\min} < I_{ph} \leq \dfrac{V_{reset} + V_T - V_1}{S \cdot t_1} = I_{ph\_1} \\ V_{reset} + V_T - V_1 + S \cdot I_{ph} \cdot (t - t_1) & \dfrac{V_{reset} + V_T - V_1}{S \cdot t_1} < I_{ph} \leq \dfrac{V_1 - V_2}{S \cdot (t_2 - t_1)} = I_{ph\_2} \\ V_{reset} + V_T - V_2 + S \cdot I_{ph} \cdot (t - t_2) & \dfrac{V_1 - V_2}{S \cdot (t_2 - t_1)} < I_{ph} \leq \dfrac{V_2 - V_3}{S \cdot (t_3 - t_2)} = I_{ph\_3} \\ V_{reset} + V_T - V_3 + S \cdot I_{ph} \cdot (t - t_3) & \dfrac{V_2 - V_3}{S \cdot (t_3 - t_2)} < I_{ph} \leq \dfrac{V_3 - V_4}{S \cdot (t_4 - t_3)} = I_{ph\_4} \\ V_{reset} + V_T - V_4 + S \cdot I_{ph} \cdot (t - t_4) & \dfrac{V_3 - V_4}{S \cdot (t_4 - t_3)} < I_{ph} \end{cases} \qquad \textbf{Eq. 89}$$

V_1, V_2, V_3, V_4 are the programmable intermediate levels of reset control signal V$_{hpch}$. t_1, t_1, t_1, t_1 are the time instants when the reset control signal changes. V_T is the threshold voltage of reset transistor M$_R$. Finally, $I_{ph}^{\min}$ represents the minimum illuminance detectable due to the floor of random noise.

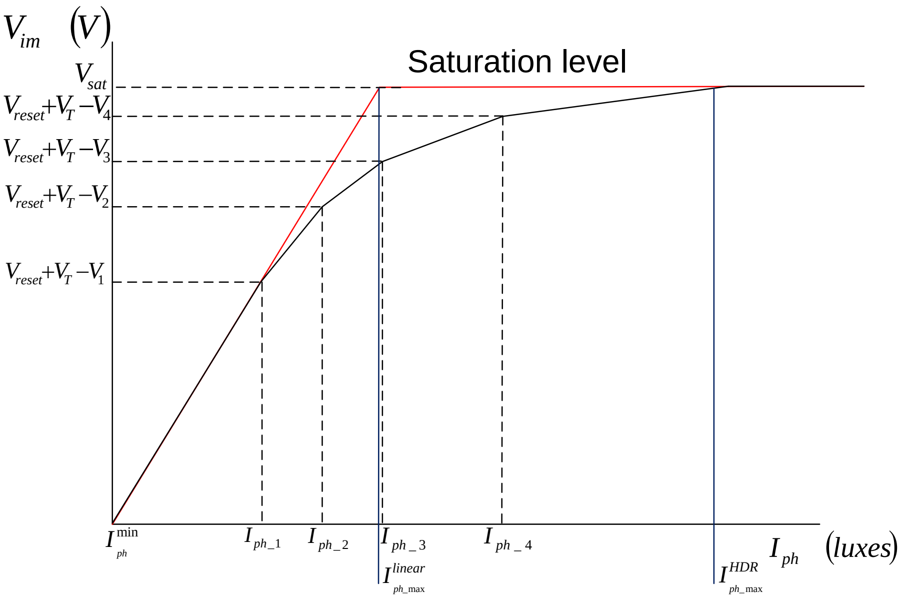


Figure 23. **HDR SENSING MODE CHARACTERISTIC.**

The **Dynamic Range** (DR) in linear mode is given by:

$$DR_{linear} = 20 \cdot \log\left(\frac{I_{ph\_\max}^{linear}}{I_{ph}^{\min}}\right) \qquad \textbf{Eq. 90}$$

Where $I_{ph\_\max}^{linear}$ is the maximum illuminance in linear mode for which the image data reaches the saturation value. In HDR mode, the dynamic range is improved. The increment of decibels is:

$$\Delta DR = 20 \cdot \log\left(\frac{I_{ph\_\max}^{HDR}}{I_{ph\_\max}^{linear}}\right) \qquad \textbf{Eq. 91}$$

the increment depending on the programmed compression law.

3.2.5. MULTIPLIER-ACCUMULATOR CIRCUIT (MAC)

3.2.5.1. ARITHMETIC OPERATIONS

The **MAC** implements the following operation:

$$I_{out} = I_0 + k \cdot (I_1 - I_2) \qquad \textbf{Eq. 92}$$

where:

$$I_0 = V_{offset} - V_{zero} \quad I_0 \in [-V_R, V_R] \quad V_{offset} \in [V_{\min}, V_{\max}] \quad V_{zero} \in [V_{\min}, V_{\max}]$$

$$I_1 = V_{im1} - V_{zero} \quad I_1 \in [-V_R, V_R] \quad V_{im1} \in [V_{\min}, V_{\max}]$$

$$I_2 = V_{im2} - V_{zero} \quad I_2 \in [-V_R, V_R] \quad V_{im2} \in [V_{\min}, V_{\max}]$$

V_{im1} and V_{im2} are the voltages associated to the corresponding input images, V_{zero} is the reference voltage which fixes the zero level in the processing domain. V_{offset} is an analogue internal programmable reference to define the offset contribution. k is a programmable constant whose value can be 2, 1 or 1/2. Addition, subtraction and scaling operations can be developed using this basic **MAC** operation.

Figure 24 shows the electrical implementation of the **MAC** block by using Switched-Capacitor (SC) circuits with control and configuration signals marked in blue. Figure 25 shows the sequence of control signals required to implement a basic operation.

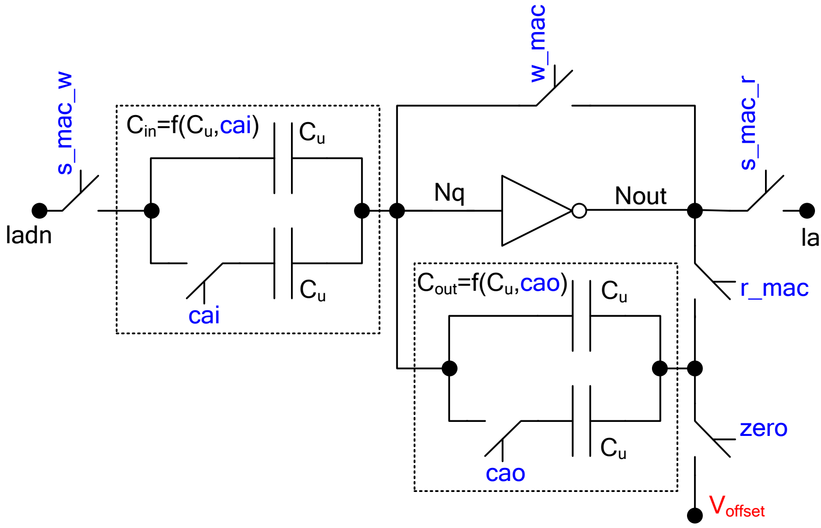


Figure 24. ELECTRICAL IMPLEMENTATION OF MAC.

Three phases can be defined in the MAC operation.

- During **phase 1**, the charge stored in C_{in} and C_{out} is given by the equations:

$$Q_{in}(ph1) = C_{in}(V_{im1} - V_{Nq}) \qquad \textbf{Eq. 93}$$

$$Q_{out}(ph1) = C_{out}(V_{offset} - V_{Nq}) \qquad \textbf{Eq. 94}$$

In this phase 1, the MAC reads the operand V_{im1} from the internal local node Iadn and stores it in the C_{in} capacitor. The C_{out} capacitor is reset to the reference V_{offset}.

- During **phase 2**, the charge stored in C_{in} and C_{out} is:

$$Q_{in}(ph2) = C_{in}(V_{im2} - V_{Nq}) \qquad \textbf{Eq. 95}$$

$$Q_{out}(ph2) = C_{out}(V_{Nout} - V_{Nq}) \qquad \textbf{Eq. 96}$$

In phase 2, the MAC reads the operands V_{im2} and stores it in C_{in}. The capacitor C_{out} is connected to implement the feedback loop storing the charge difference in capacitor C_{in} between phase 1 and phase 2. The output voltage is readily calculated into first-order approach by applying charge conservation principle to high impedance node Nq:

$$V_{Nout} = V_{offset} + \frac{C_{in}}{C_{out}}(V_{im1} - V_{im2}) \qquad \textbf{Eq. 97}$$

- During **phase 3**, the MAC writes in the analogue local node **Iadn** the result given by the arithmetic operation.

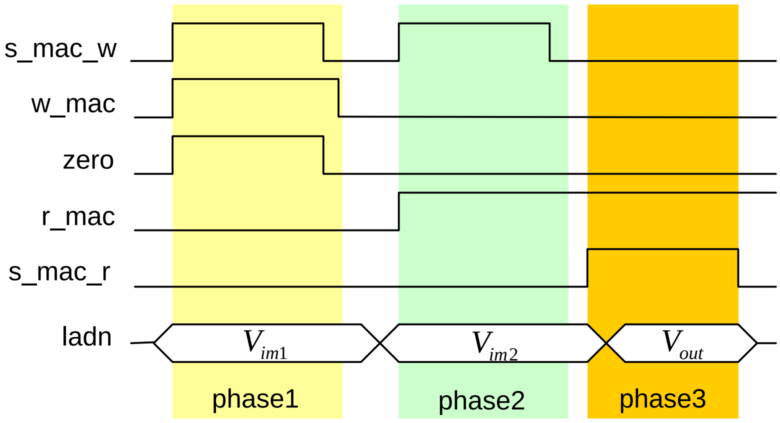


Figure 25. SEQUENCE OF INSTRUCTIONS FOR A **MAC** OPERATION.

The programmable gain factor k is configured through the signal cai and cao. Table 4 describes the gain factor which can be programmed:

cao	cai	C_{in}/C_{out}	k
0	0	C_u/C_u	1
0	1	$2C_u/C_u$	2
1	0	$C_u/2C_u$	½
1	1	$2C_u/2C_u$	1

Table 4. Configuration of gain factor.

The electrical implementation of the inverter and the capacitor within the **MAC** is the simplest possible to preclude penalizing the pixel pitch – see Figure 11. Errors caused by these simple circuits, and particularly the non-linearity of the MOS capacitors must be taken into account during the design phase and will be addressed in section 3.3.2.

3.2.5.2. THRESHOLD OPERATION

The **MAC** circuit can also be used to implement the analogue-to-binary conversion or threshold operation. The sequence of control signals required is represented in Figure 26. Since the threshold operation is described by the equation:

$$V_{out} = \begin{cases} V_{\max} & if\left(V_{im1} - V_{im2}\right) > 0 \\ V_{\min} & if\left(V_{im1} - V_{im2}\right) < 0 \end{cases} \qquad \textbf{Eq. 98}$$

The **MAC** SC circuit behaves as an auto-zero comparator with this control sequence.

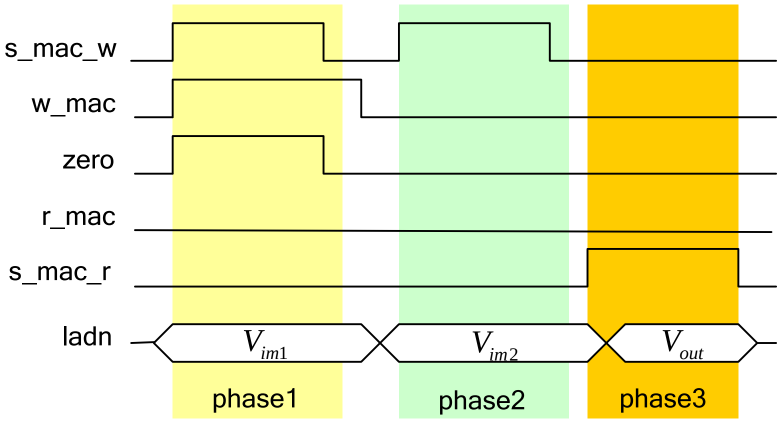


Figure 26. SEQUENCE OF INSTRUCTIONS FOR A THRESHOLD OPERATION.

During **phase 3**, the result of comparison operation is put by the **MAC** circuit in the local node **Iadn** to pass the analogue data to a digital memory, interconnecting the internal cell nodes **Iadn** and **Iddn**. For this threshold operation, the possibility of connecting a programmable analogue global reference to each analogue local node **Iadn** in parallel is very useful. This function permits to define different threshold levels. The threshold level can be defined by an image, in this case, several threshold levels are considered locally.

3.2.6. Logic and morphological operators

3.2.6.1. Local logic unit (LLU)

This programmable block implements a large catalog of logic operation through programming. It consists of a programmable multiplexor controlled by the logic operands Op1, Op0 and whose inputs are for configuration signals, namely: ht00, ht01, ht10, ht11. These configuration signals define the truth table of combinatory operation. See Figure 27 and Table 5.

Op0	Op1	Out
0	0	ht00
0	1	ht01
1	0	ht10
1	1	ht11

Table 5. Definition of binary operation.

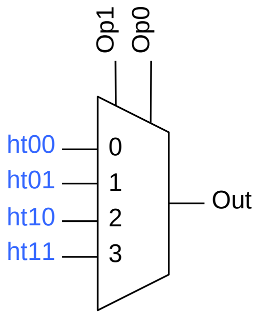


Figure 27. LLU implementation.

The operands are provided by two digital memories. Different combinatorial operations (NOR, NAND, XOR, AND, OR, etc.) can be implemented by properly setting ht00, ht01, ht10 and ht11. For instance:

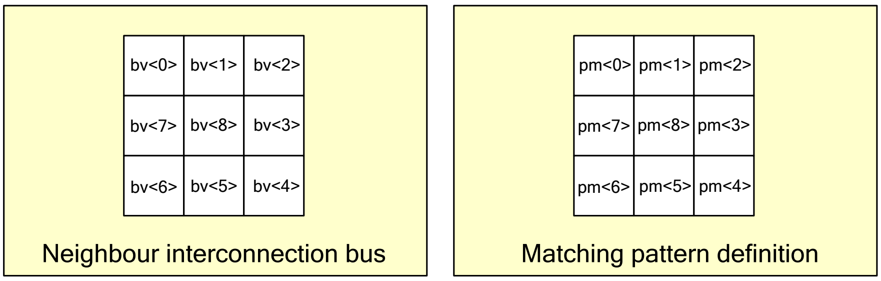
- NOR operation requires ht00=1, ht01=0, ht10=0 and ht11=0,
- NXOR operation requires ht00=1, ht01=0, ht10=0 and ht11=1.

3.2.6.2. Morphological operators

The basic operation of binary morphology, known like *Hit-and-Miss* operation [Russ92] is developed by the **HitAndMiss** functional block whose architecture is described in Figure 28. It checks whether the 3x3 neighborhood of a cell matches a specified pattern or not.

The inputs to this block are the image to be analyzed (that must be stored in a specific *LDM*) and the pattern to match. The 3x3 local input image is driven by the bus bv<8:0> in Figure 28 and the pattern is defined by the configuration signals *pm<8:0>* in the same figure.

Through the interconnection bus bv<8:0>, the neighbor pixel values are considered in the morphological operation following the scheme depicted below



bv<0>	bv<1>	bv<2>
bv<7>	bv<8>	bv<3>
bv<6>	bv<5>	bv<4>

Neighbour interconnection bus

pm<0>	pm<1>	pm<2>
pm<7>	pm<8>	pm<3>
pm<6>	pm<5>	pm<4>

Matching pattern definition

The result is an image with white pixels at points whose 3x3 neighbourhood matches the input pattern.

$$f_{morph}(bv<8:0>)=1 \Leftrightarrow bv<8:0>= pm<8:0>$$
$$f_{morph}(bv<8:0>)=0 \Leftrightarrow bv<8:0>\neq pm<8:0> \qquad \textbf{Eq. 99}$$

Three states can be defined as the value of a pixel for the pattern matrix, namely:

- 1 (white),
- 0 (black) and,
- DNC (Do Not Care*),* meaning that the value of that pixel is irrelevant to the matching.

The DO Not Care pattern is defined through the configuration bus pm<8:0> in a similar manner like the Matching pattern. In the DNC pattern, 1 means that the corresponding operands do not affect to the morphological operation.

As a very simple example, the following pattern looks for pixels that are south-western corners inside an image.

$$\begin{bmatrix} 0 & 1 & DNC \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

This Matching pattern is defined in the **HitAndMiss** blocks in two steps. Into first step, the DNC pattern defined through the configuration bus pm<8:0> is stored in the dynamic digital memories (DNC memories). The DNC pattern for this example is:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Into second step, the rest of coefficients in the Matching matrix are defined through the bus pm<8:0>. This matching matrix in this case is:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

The operation of **HitAndMiss** block is controlled by the signals low, high, dnnc, dpnc, s_morph, and pm<8:0>.

The sequence of instructions to develop the morphological operation is described in Figure 29 and Table 6.

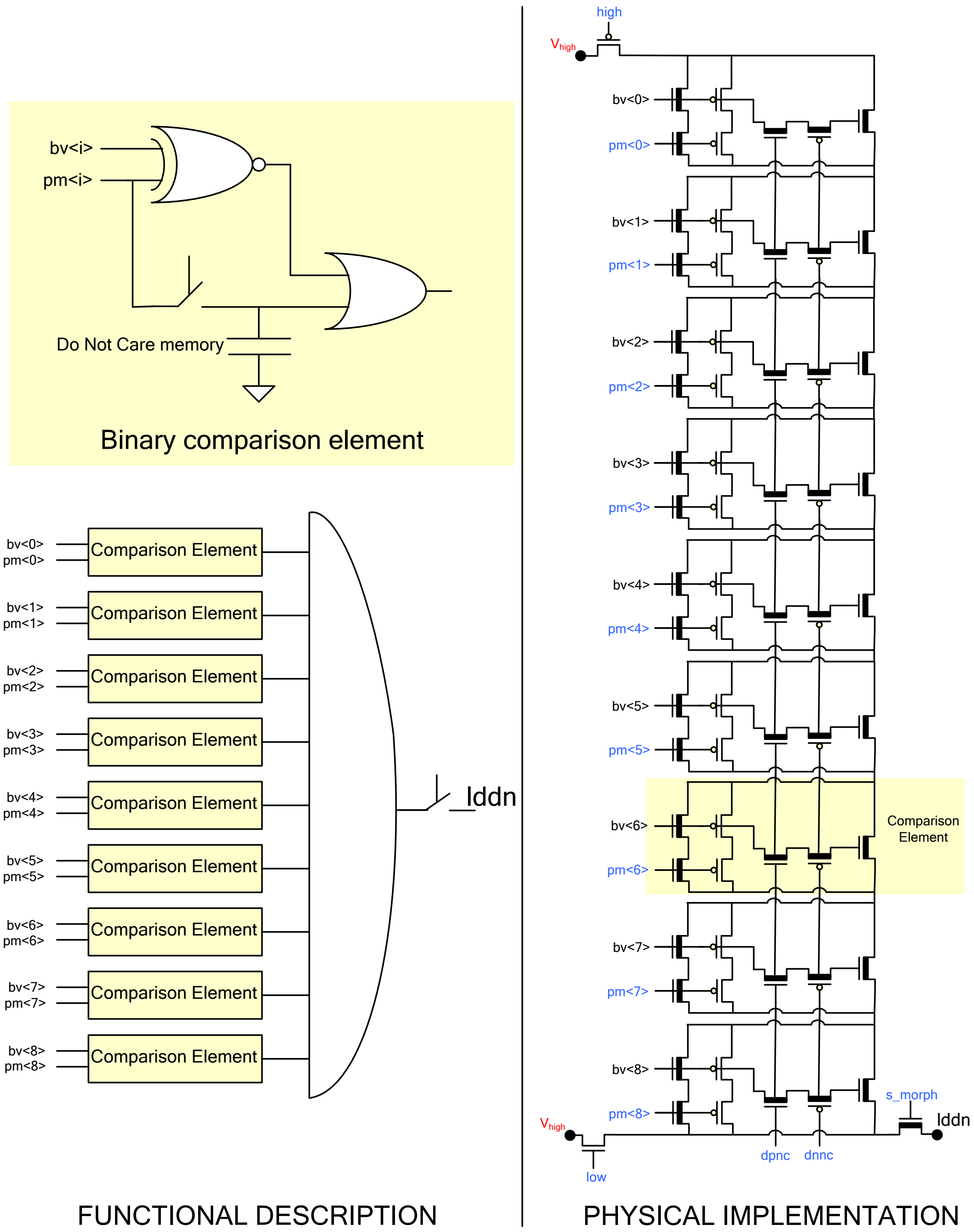


Figure 28. HITANDMISS FUNCTIONAL BLOCK.

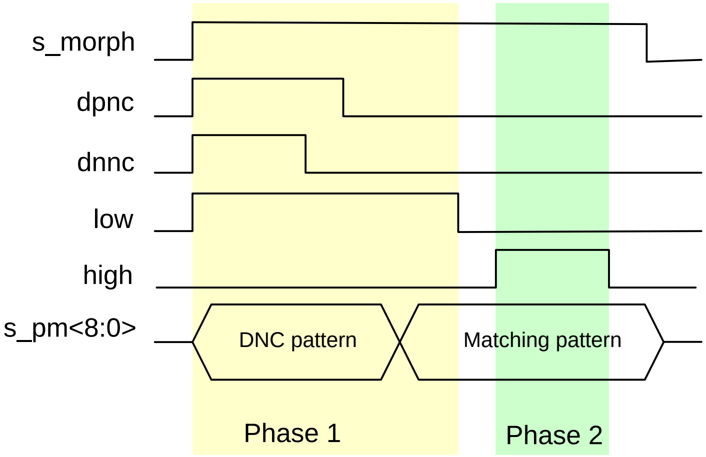


Figure 29. SEQUENCE OF INSTRUCTIONS FOR HITANDMISS OPERATION.

Nº Instruction	s_morph	dnnc	dpnc	low	high	pm<8:0>
1	0	0	0	0	0	0
2	1	1	1	1	0	dnc_pattern
3	1	0	1	1	0	dnc_pattern
4	1	0	0	1	0	dnc_pattern
5	1	0	0	1	0	match_pattern
6	1	0	0	0	0	match_pattern
7	1	0	0	0	1	match_pattern
8	1	0	0	0	0	match_pattern
9	0	0	0	0	0	match_pattern

Table 6. Sequence of instructions for HitAndMiss operation.

During **phase 1**, the DNC pattern is loaded in the dynamic digital memories and the local digital data node **Iddn** and **HitAndMiss** block are pre-charged to a low reference voltage (V_{low}) -- see Figure 28 This reference corresponds to a 0 digital value. At the end of this phase, the matching pattern is fixed in the **HitAndMiss** block when the pre-charge process is taking place.

During **phase 2**, a high reference voltage (V_{high}) is applied to the **HitAndMiss** block as Figure 28 shows. If the matching pattern and pixel neighbourhood coincide, the output of morphological block and local node **Iddn** are charged to the high reference. This reference corresponds to a 1 digital value.

The result of morphological operation will be stored through the data node **Iddn** in a local digital memory. The reference voltage fixed in the **Iddn** will be regenerated to digital values by the local digital memory.

3.2.7. CONVOLUTIONS

The **MAC** block together with the analogue shifting block permits the implementation of *spatial filters* defined by 3x3 templates.

In the implementation of spatial filters, each contribution associated to a neighbor requires a multiplication-accumulation operation defined by the equation:

$$\frac{C_{in}}{C_{out}}\left(V_i - V_{zero}\right) \qquad \textbf{Eq. 100}$$

V_i is the signal voltage associated to the corresponding neighbor. The weight of filter is defined by the programmable factor $\frac{C_{in}}{C_{out}}$. The sign of the neighbor contribution is determined by the phase of **MAC** operation in which signal voltage is read, see Figure 25 and Eq.97.

For example, equation Eq.100 represents a positive contribution. The operation with opposite contribution sign to Eq. 100 is:

$$\frac{C_{in}}{C_{out}}\left(V_{zero}-V_i\right) \qquad \textbf{Eq. 101}$$

Due to the circuit topology used for the implementation of **MAC** block, the weight or scaled factor used during the consecutive multiplication-accumulation process must remain constant. Under these conditions, the convolution operation can be implemented as the sum of intermediate values resulting from the multiplication-accumulation processes associated to each group of neighbors with the same contribution weight.

Figure 30 and Table 7 show the sequence of instructions required to carry out three consecutive multiplication-accumulation operations in the **MAC** block, described mathematically by:

$$V_{out}=\frac{C_{in}}{C_{out}}\left[\left(V_1-V_{zero}\right)+\left(V_2-V_{zero}\right)-\left(V_3-V_{zero}\right)\right] \qquad \textbf{Eq. 102}$$

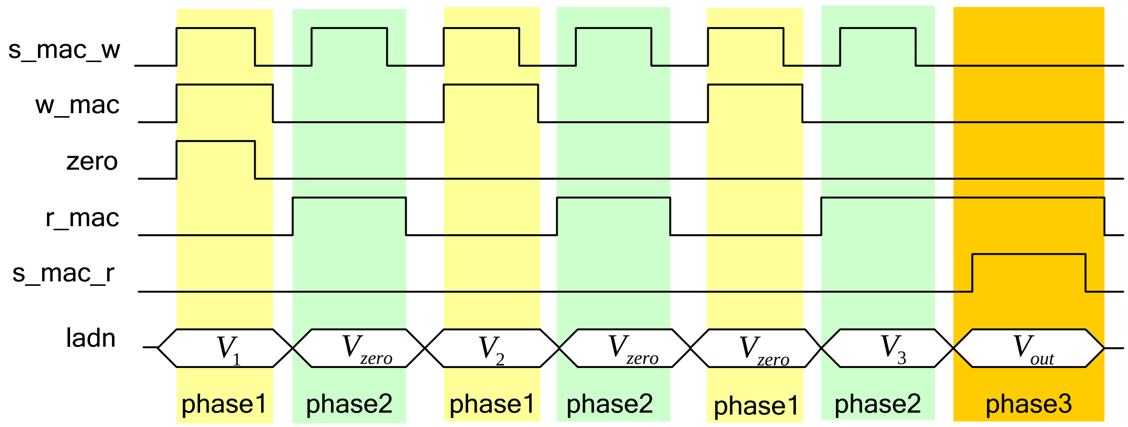


Figure 30. CONSECUTIVE MULTIPLICATION-ACCUMULATION OPERATION.

Nº Instruction	r_mac	w_mac	zero	s_mac_w	s_mac_r	ladn
1	0	0	0	0	0	V_1
2	0	1	1	1	0	V_1
3	0	1	0	0	0	V_1
4	0	0	0	0	0	V_1
5	1	0	0	0	0	V_{zero}
6	1	0	0	1	0	V_{zero}
7	1	0	0	0	0	V_{zero}
8	0	0	0	0	0	V_2
9	0	1	0	1	0	V_2
10	0	1	0	0	0	V_2
11	0	0	0	0	0	V_{zero}
12	1	0	0	0	0	V_{zero}
13	1	0	0	1	0	V_{zero}

14	1	0	0	0	0	V_{zero}
15	0	0	0	0	0	V_{zero}
16	0	1	0	1	0	V_{zero}
17	0	1	0	0	0	V_{zero}
18	0	0	0	0	0	V_3
19	1	0	0	0	0	V_3
20	1	0	0	1	0	V_3
21	1	0	0	0	0	V_3
22	1	0	0	0	1	V_{out}
23	1	0	0	0	0	V_{out}
24	0	0	0	0	0	V_{out}

Table 7. Instructions for multiplication-accumulation operation.

The signal voltages belonging to the pixel neighbors are connected to the local node **ladn** through the interconnection bus bv<7:0> which is input of the analogue multiplexer. The analogue multiplexer is controlled by the configuration bus pm<7:0>. The interconnection scheme between the pixel and the neighborhood is shown in Figure 17 The reference voltage V_{zero} is applied to **ladn** node by the Generic reference block.

3.2.8. DIFFUSION GRID

3.2.8.1. RESISTIVE DIFFUSION GRID

A resistive grid connecting pixel is included within the Q-Eye to emulate the implementation of *Gaussian filters* [Fern12] [Kana07]. Assuming a 2-Dimensional symmetric Gaussian function $G_{2-D}(x, y)$ with *variance* σ , described by the equation:

$$G_{2-D}(x, y) = \frac{1}{2 \cdot \pi \cdot \sigma^2} e^{-\frac{x^2+y^2}{2 \cdot \sigma^2}} \qquad \textbf{Eq. 103}$$

the *convolution mask* required to implement it is calculated from the *impulse response* which results into a matrix of parameters whose values depend on σ. The larger the variance, the larger the size of the mask needed to accurately implement the filter. In practice the mask size gets truncated to a limited *neighborhood*. The matrix below displays the matrix values within a *radius-2* neighborhood for $\sigma = 1$:

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0133
0.0030	0.0133	0.0219	0.0133	0.0030

Outside this radius, the coefficients are considered negligible.

The resistive grid implements Gaussian filters as the outcome of a *diffusion* process. It is extremely efficient in both speed and power consumption [Fern12]. Assuming the grid driven by a constant input image equal to 0 whose central pixel is 1 (normalized input), the *diffusion length* n_{diff} is defined as the radius of the circumference, expressed in number of pixels, where the value associated to the pixels are $1/\sqrt{2}$ below (3dB fall) of the final value for the central pixel. On the one hand, this length depends on the time constant of the RC network implementing the grid. On the other hand, this diffusion length can be shown to be proportional to the variance parameter σ :

$$n_{diff} = k \cdot \sigma \qquad \textbf{Eq. 104}$$

Hence, the variance and the associated convolution mask can be synthesized by properly setting the RC time constant. Let us consider for simplicity the case of 1-dimensional diffusion network – see Figure 31.

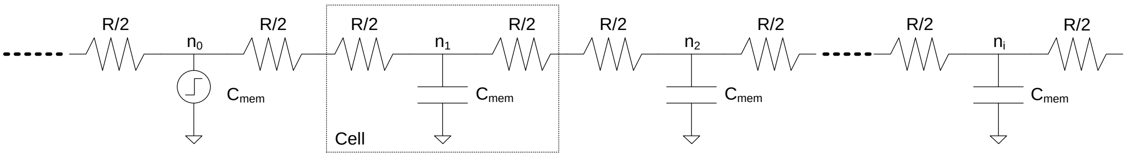


Figure 31. 1-DIMENSIONAL DIFFUSION NETWORK MODEL.

Assume the network is finite, although large, and that a unitary step is applied to one cell, located almost infinitely far away from the end of the network. The time constant for a network of n resistors can be calculated considering the *open-circuit time constant* approach [Sedr04] [8]. In this case, the time constant associated to each node of network from the input node n_0 is given by the equation:

$$\tau_i = R \cdot C_{mem} \sum_{n=1}^{i} n = R \cdot C_{mem} \cdot \frac{i \cdot (i+1)}{2} \qquad \text{Eq. 105}$$

The temporal evolution of a particular node with time constant τ_i is:

$$V_{out}(i) = 1 - e^{-\frac{t}{\tau_i}} \qquad \text{Eq. 106}$$

Consequently, the time required for each node to reach the $1/\sqrt{2}$ factor of input step is:

$$t_i = -\ln\left(1 - \frac{1}{\sqrt{2}}\right) \cdot \tau_i = 1.23 \cdot \tau_i = 1.23 \cdot R \cdot C_{mem} \cdot \frac{i \cdot (i+1)}{2} \qquad \text{Eq. 107}$$

Therefore, the following relation between diffusion length and diffusion time is obtained:

$$t_{diffusion} = 0.615 \cdot R \cdot C_{mem} \cdot n_{diff} \cdot \left(n_{diff} + 1\right) = \kappa \cdot \tau \cdot n_{diff} \cdot \left(n_{diff} + 1\right) \qquad \text{Eq. 108}$$

where τ is the constant time of the diffusion network and κ is a constant of proportionality.

The diffusion grid is composed by a CMOS resistance, and has several configuration and control signals as it is depicted in Figure 32. The control signal s_rg connects the local grid node Nrg to the data node **ladn.** The control signal rg_me launches the diffusion process; and the configuration signal e_m_rg enables the mask operation during the diffusion process.

The resistive grid operation involves three phases:

- The first phase is a reset phase of the diffusion grid where all internal nodes of network are fixed to a specific global reference, for instance V_{zero} .
- During second phase, the data node **ladn** is initialized to the input image which will be filtered. The input operand is driven by a local analogue memory.
- During third phase, the internal capacitance of local grid node Nrg is connected to the **ladn** node remaining the local analogue memory node connected in writing configuration to the data node **ladn** and local grid node. And simultaneously, the diffusion process is launched. When the diffusion process finishes, the resulting filtered image is loaded in this analogue memory. Therefore, the capacitance associated to a particular node of network during the diffusion process is the capacitance of the analogue memory.

[8] The dominant high-frequency time constant is estimated as the sum of the individual time constants due to each of the capacitance when all other capacitances are set to zero. To find the individual time constant for a given capacitance, independent voltage sources are replaced with ground (independent current sources are opened), and the resistance seen by the capacitor is determined.

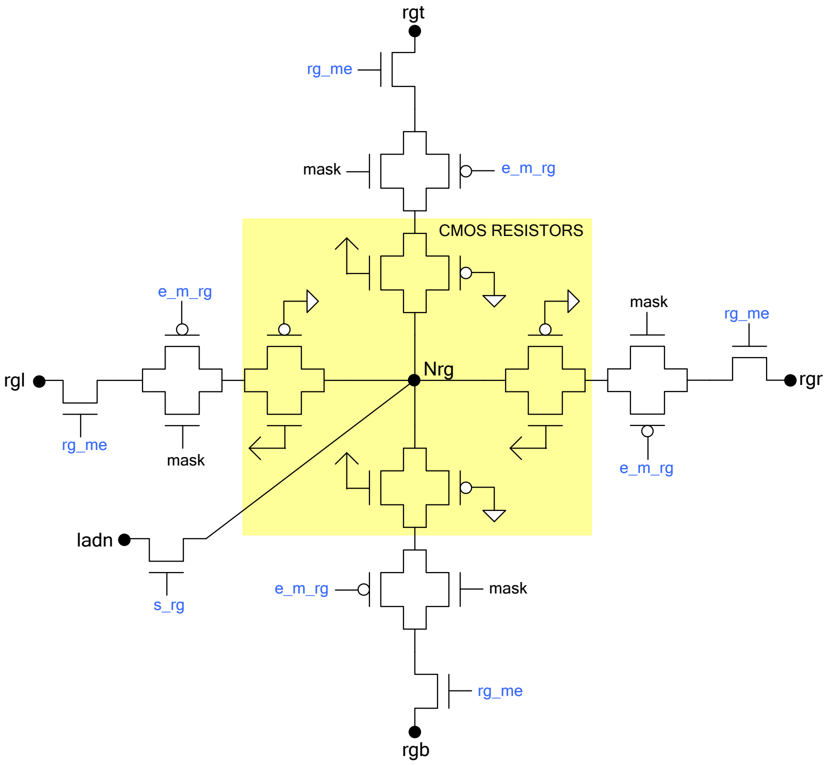


Figure 32. DIFFUSION GRID ARCHITECTURE.

The diffusion operation can be masked, and only pixels pointed out by a binary image defined as a mask are updated. The binary mask image is stored in a specific local digital memory. To optimize the Q-Eye pixel area, the binary mask used in the **LAMs** bank is defined by the same local digital memory which stores the mask for the resistive grid

Figure 33 and Table 8 represent the instruction sequence to carry out a diffusion process. In this case, the control signals of **LAMs** block are shown for clarity.

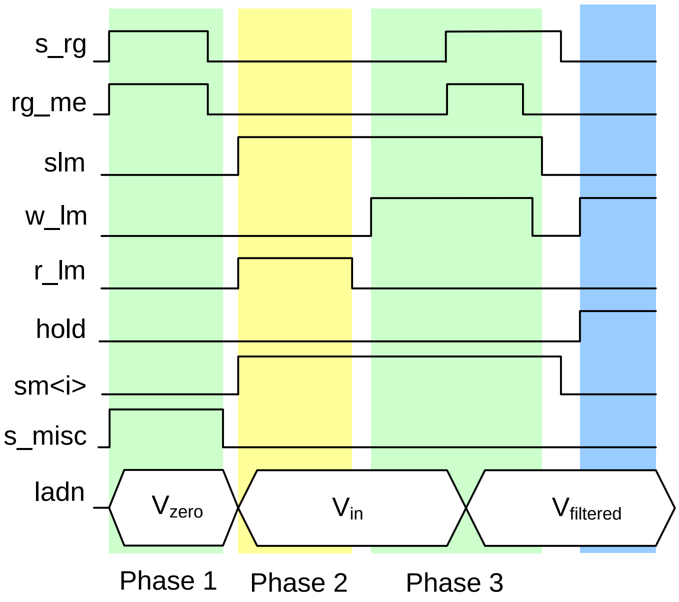


Figure 33. INSTRUCTIONS FOR RESISTIVE GRID OPERATION.

Nº Instruction	s_misc	s_rg	rg_me	slm	w_lm	r_lm	hold	sm<i>	ladn
1	0	0	0	0	0	0	1	0	
2	1	1	1	0	0	0	1	0	V_{zero}
3	1	0	0	0	0	0	1	0	
4	0	0	0	0	0	0	0	0	
5	0	0	0	1	0	1	0	1	V_{in}
6	0	0	0	1	0	0	0	1	V_{in}
7	0	0	0	1	1	0	0	1	V_{in}
8	0	1	1	1	1	0	0	1	$V_{filtered}$
9	0	1	0	1	1	0	0	1	$V_{filtered}$
10	0	1	0	1	0	0	0	1	$V_{filtered}$
11	0	0	0	0	0	0	0	0	
12	0	0	0	0	1	0	1	0	

Table 8. Instructions for a diffusion operation.

The control signal rg_me which launches the diffusion process is generated in a different way by the system control block **PCU**. This signal does not come directly from the PBCTRL instruction register but the instruction bit in the **PCU** vector associated to the rg_me signal goes to a specific block which generates the pulse in rg_me signal to implement the diffusion. This block is a mono-stable which can generate a single pulse controlled by the PCU instruction bit rg associated to the rg_me signal. Figure 34 depicts the control signal generation associated to the resistive grid operation. The width of this pulse is configured by a configuration register included in the mono-stable block.

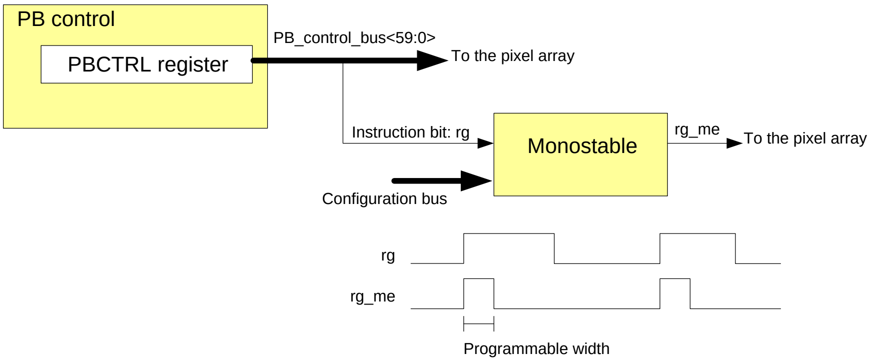


Figure 34. DIFFUSION PROCESS CONTROL.

The time constant associated to the diffusion network $R \cdot C_{mem}$ is small because the implementation of larger values of either resistances or capacitances implies more area for the physical integration and a decrement of Q-Eye pixel pitch. Therefore, there is a tradeoff between implementation area (pixel pitch) and time constant of diffusion network.

Considering that the diffusion length is inversely proportional to time constant of the diffusion network, this last parameter together with the diffusion time determine the minimum diffusion length attainable by the diffusion operation.

In the particular case of the Q-Eye system, the time constant of the diffusion network is around 5ns (nanoseconds). But, the minimum diffusion time is one system clock period in case that the diffusion control signal is generated by the control block **PCU**. So, the minimum diffusion length is limited by the system clock period which is 20ns.

The diffusion time required to implement Gaussian filters with small variance parameters is very short, lesser than the clock period of digital control block operation. The **PCU** block cannot generate pulses smaller than the system clock period and a specific block must be designed to generate very short diffusion control pulses. This is the reason why the specific block (monostable) has been designed in order to generate short diffusion times.

The Gaussian filters are used in image processing systems for improving the quality of the image or for preparing the image for further processing through for instance, edge detection. These applications required a good control of the diffusion time, which means a good control on the value of the variance associated to the low pass filter.

3.2.8.2. SC DIFFUSION GRID

Resistances used for the diffusion can be implemented by SC circuits. Including an additional switch and control signal to the Image Shifter multiplexer, a SC diffusion network with 3x3 neighborhood can be implemented, see Figure 35. This SC diffusion network allows directional diffusion processes selecting only one direction with the interconnection control bus pm<7:0>.

The value of equivalent resistance associated with the SC circuit is:

$$R_{eq} = \frac{1}{C_{v\_lcl} \cdot f_{clk}} \qquad \textbf{Eq. 109}$$

C_{v_lcl} is the capacitance associated with the local node *v_lcl* of Image Shifter and f_{clk} is the switching frequency of control signals.

The diffusion length is controlled by the constant time $R_{eq} \cdot C_{mem}$. Consequently, the diffusion length is configured by the switching frequency f_{clk}.

The diffusion operation with the shifter multiplexer involves three phases, similar to the resistive grid block described before. The first phase is a reset phase where all internal nodes of SC network are initialized to a global reference. The second one is the initialization phase where the input image is loaded into the **ladn** node of Q-Eye cell. And the third phase is the diffusion process. During the diffusion process, an analogue memory remains connected to the **ladn** node in writing configuration. Figure 36 and Table 9 describe the sequence of instructions to develop a diffusion process, for simplicity, only one direction has been considered.

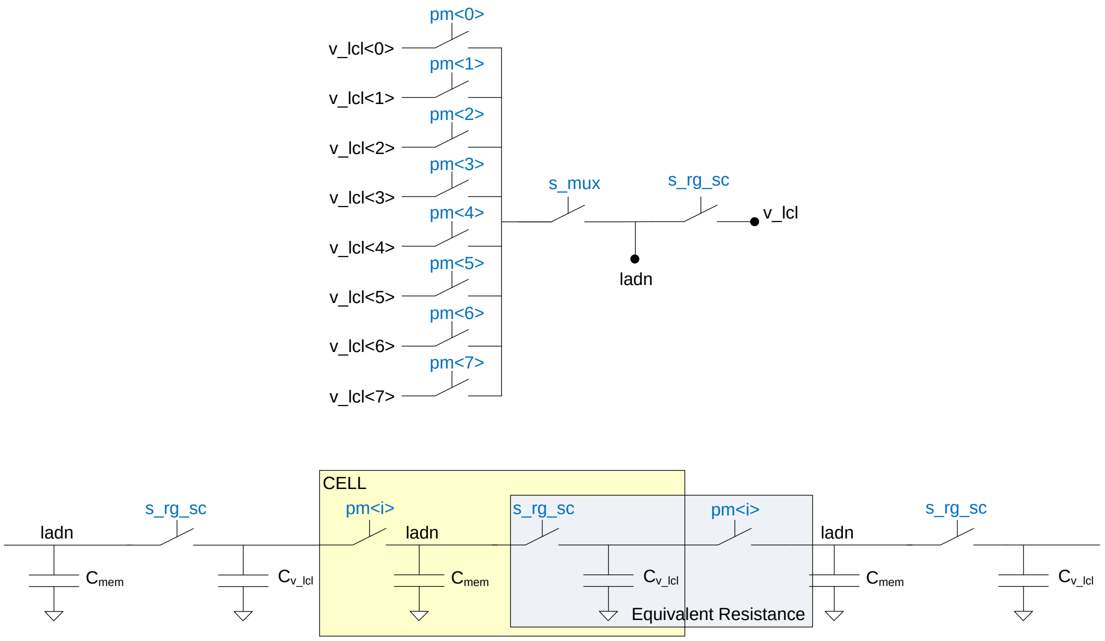


Figure 35. SC DIFFUSION NETWORK STRUCTURE.

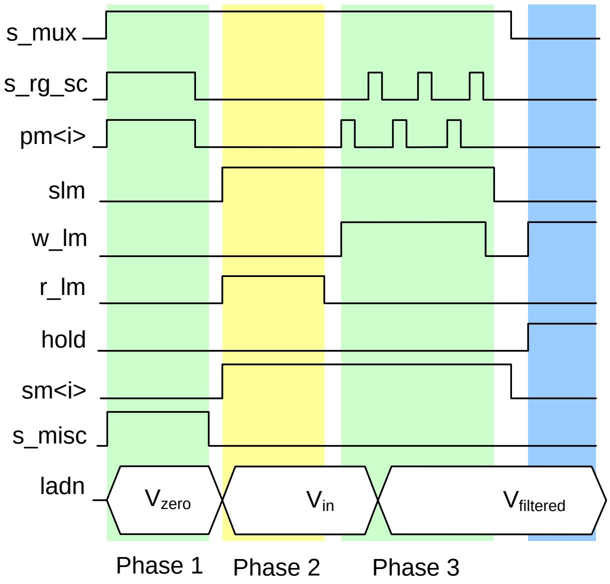


Figure 36. CONTROL SIGNALS FOR THE SC DIFFUSION OPERATION.

Nº Instruction	s_misc	s_mux	s_rg_sc	pm<i>	slm	w_lm	r_lm	hold	sm<i>	ladn
1	0	0	0	0	0	0	0	1	0	
2	1	1	1	1	0	0	0	1	0	V_{zero}
3	1	1	0	0	0	0	0	1	0	
4	0	1	0	0	0	0	0	0	0	
5	0	1	0	0	1	0	1	0	1	V_{in}
6	0	1	0	0	1	0	0	0	1	V_{in}
7	0	1	0	0	1	1	0	0	1	V_{in}
8	0	1	1	0	1	1	0	0	1	$V_{filtered}$
9	0	1	0	0	1	1	0	0	1	$V_{filtered}$
10	0	1	0	1	1	1	0	0	1	$V_{filtered}$
11	0	1	0	0	1	1	0	0	1	$V_{filtered}$
12	0	1	1	0	1	1	0	0	1	$V_{filtered}$
13	0	1	0	0	1	1	0	0	1	$V_{filtered}$
14	0	1	0	1	1	1	0	0	1	$V_{filtered}$
15	0	1	0	0	1	1	0	0	1	$V_{filtered}$
…	0	1	…	…	1	1	0	0	1	$V_{filtered}$
	0	1	0	0	1	0	0	0	1	$V_{filtered}$
	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	1	0	1	0	

Table 9. **SC diffusion instruction sequence.**

3.3. CHARACTERIZATION AND MODELING OF ERRORS

3.3.1. ANALOGUE MEMORY ERRORS

The LAM block of Figure 10 employs the concept and principles of the S&H circuits of Figure 37 [Greg94]. We will use this core circuit for error calculation. It works either in **writing** or on **reading** mode depending on the control clocks shown in Figure 38. The following errors impact these modes, respectively:

- **Writing** (sampling-tracking) (switches SW1 and SW2 ON, switch SW3 OFF):
  - Temporal settling error
  - Feedthrough and charge injection
  - Electronic noise
- **Reading** (holding) (switches SW1 and SW2 OFF, switch SW3 ON):
  - Temporal settling error
  - Offset and gain error (offset and finite gain of amplifier)
  - Leakages
  - Electronic noise

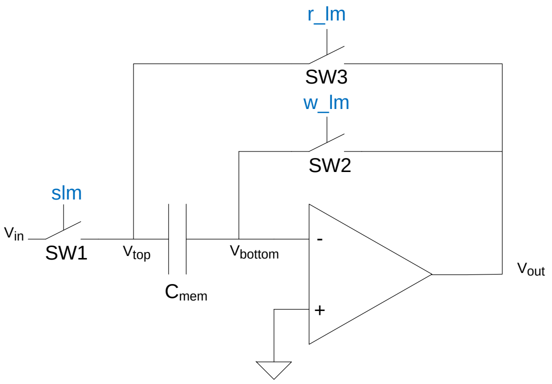


Figure 37. **GREGORIAN'S S&H [GREG94].**

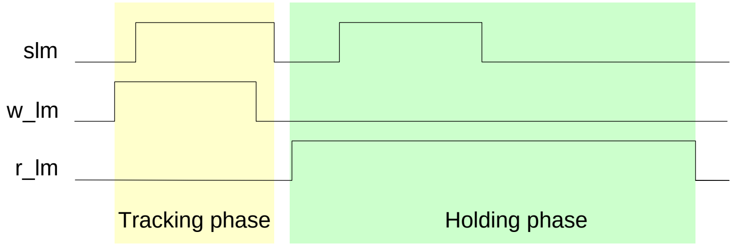


Figure 38. **CONTROL PHASES.**

3.3.1.1. TEMPORAL SETTLING ERROR

Settling errors are due to incomplete transient of the nodes in the S&H circuit. To analyze them, we first recast the equations governing the balance of charge at the node v_{bottom} corresponding to the bottom plate of the sampling capacitor. The charge at this node has two components, namely:

- that stored in the sampling capacitor itself,
- that stored in the input parasitic capacitance of the opamp.

During the **writing phase** these components are respectively:

$$Q_{mem}(wr) = C_{mem}\left[-v_{in} + v_{bottom}(wr)\right] \quad \textbf{Eq. 110}$$

$$Q_{in}(wr) = C_{in}\left[v_{bottom}(wr)\right] \quad \textbf{Eq. 111}$$

where $v_{bottom}(wr)$ is the voltage of bottom capacitor plate at writing phase.

During the **reading phase** these charges are:

$$Q_{mem}(rd) = C_{mem}\left[-v_{out} + v_{bottom}(rd)\right] \quad \textbf{Eq. 112}$$

$$Q_{in}(rd) = C_{in}\left[v_{bottom}(rd)\right] \quad \textbf{Eq. 113}$$

Where, $v_{bottom}(rd)$ is the voltage value of bottom plate at reading phase.

Note that the bottom node remains isolated during the reading phase. It means that the charge at this node cannot change, thereby yielding:

$$v_{out} = v_{in} - \left(1 + \frac{C_{in}}{C_{mem}}\right)\left[v_{bottom}(wr) - v_{bottom}(rd)\right] \qquad \textbf{Eq. 114}$$

Under the assumption that the opamp is ideal (infinite DC gain and no dynamics), the bottom node is at v_{ref} in both phases - v_{ref} is the voltage at positive opamp input. In practice, the value at this node shows errors that depend on time and that increase as the transient duration decreases, namely:

$$v_{bottom}(wr) = v_{ref} + \varepsilon_{wr} \qquad \varepsilon_{wr} = \Delta V_{wr}\, e^{-\frac{t}{\tau_{wr}}} \qquad \textbf{Eq. 115}$$

$$v_{bottom}(rd) = v_{ref} + \varepsilon_{rd} \qquad \varepsilon_{rd} = \Delta V_{rd}\, e^{-\frac{t}{\tau_{rd}}}$$

These deviations provoke both **offset** and **gain errors** at the Input/Output characteristic of analogue memory. Into first approximation these offset and gain errors are, respectively:

$$\Delta V_{wr} \approx \alpha_{wr} + \beta_{wr} \cdot v_{in} \qquad \Delta V_{rd} \approx \alpha_{rd} + \beta_{rd} \cdot v_{in} \qquad \textbf{Eq. 116}$$

Note also that the *mismatch* between analogue memories of different cells introduces a random component in these errors introducing *spatial noise* in the stored and read images.

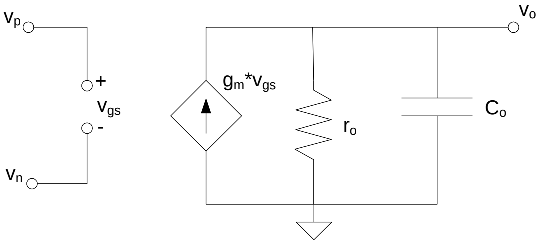


Figure 39. OTA MODEL.

These temporal errors are calculated by assuming that the OTA model of Figure 39, where g_m is the transconductance, r_o is the output resistance and C_o is the output capacitance. Then, the S&H operation gets described by the circuit of Figure 40 during writing mode and by the circuit of Figure 41 during reading mode. There r_{sw1}, r_{rw2} and r_{sw3} are the resistances associated to the switches, and C_{in} is the parasitic capacitance at the OTA input.

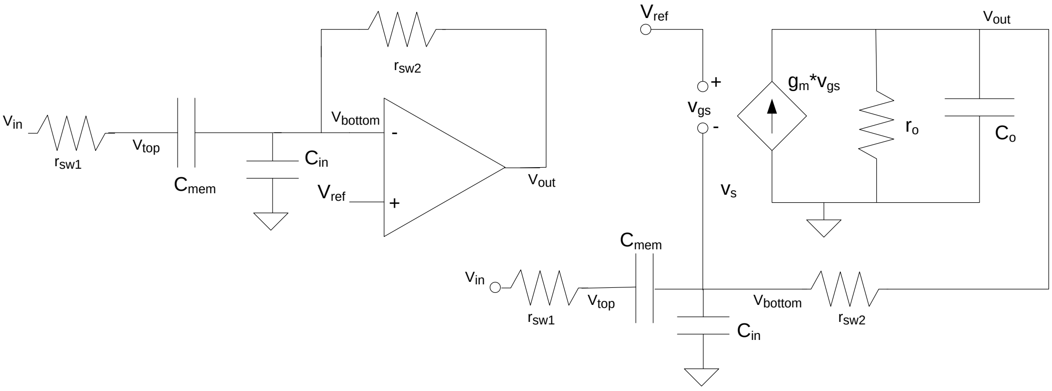


Figure 40. CIRCUITAL MODEL FOR SAMPLE OR WRITING PHASE.

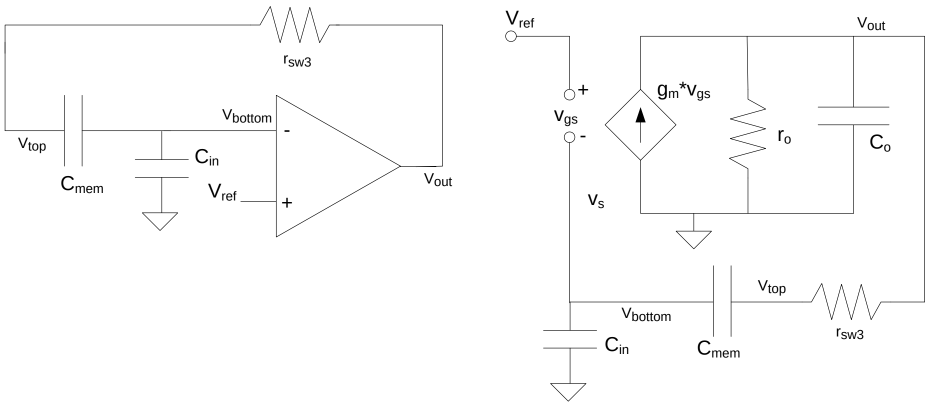


Figure 41. CIRCUITAL MODEL FOR THE READING PHASE.

Let us make the following assumptions:

- The ON resistance of SW1 (r_{sw1}) can be neglected [9]
- $g_o << g_m, g_{sw2} \;\; (g_{sw2} = \frac{1}{r_{sw2}})$
- $g_{sw2} >> g_m$

Then, the dynamic of S&H circuit in the writing phase is described by the following first order transfer function, read Appendix A:

$$H_{writing}(s) = \frac{1}{1 + \frac{C_o + C_{mem} + C_{in}}{g_m} s} = \frac{1}{1 + \tau_{wr} \cdot s} \qquad \text{Eq. 117}$$

And the acquisition error associated to the temporal response is described by the expression Eq. 118, when a step function starting at $t = 0$ and size ΔV_{wr} is applied.

$$v_\varepsilon(t) = \varepsilon_{wr} = -\Delta V_{wr} \cdot e^{-\frac{t}{\tau_{wr}}} \qquad \text{Eq. 118}$$

Regarding the **reading phase** and assuming that the switch SW3 is designed with impedance such that its effect on the dynamic of circuit is negligible, we obtain the following transfer function:

$$H_{reading} = \frac{\left(1 + \frac{C_{in}}{C_{mem}}\right)}{s \cdot \frac{C_{eq\_rd}}{g_m} + 1} = \frac{\left(1 + \frac{C_{in}}{C_{mem}}\right)}{s \cdot \tau_{rd} + 1} \qquad \text{Eq. 119}$$

The settling error during the reading phase can be calculated in the same way that the acquisition error during the sampling phase is obtained from the transfer function. Considering an input step function starting at $t = 0$ and size ΔV_{rd}:

$$v_\varepsilon(t) = \varepsilon_{rd} = -\Delta V_{rd} \cdot e^{-\frac{t}{\tau_{rd}}} \qquad \text{Eq. 120}$$

[9] The ON resistance of SW1 can be neglected provided this switch is wide enough. This is actually limited by the injection of its channel charge into the capacitor. However, this charge is basically precluded by turning SW1 OFF only after SW2 is fully OFF.

3.3.1.2. CHARGE INJECTION AND CLOCK FEEDTHROUGH

When the switch SW2 in Figure 37 is turned OFF at the end of sampling phase, the voltage stored at the capacitor changes due to two different phenomena:

- Channel **charge injection**; i.e. part of the charge which builds the MOS channel is routed to the sampling capacitor.
- **Clock feedthrough**, i.e. the coupling of the clock control signals through parasitic capacitances into the storing node – see Figure 42.

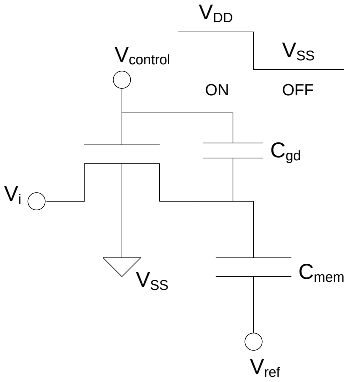


Figure 42. NMOS SWITCH.

CHARGE INJECTION. The amount of charge injected into the memory capacitor C_{mem} depends on the speed and shapes of the clock signal edges and the relative impedances seen at the access transistor terminals. Its exact calculation is extremely complicated or even unfeasible [Wegm87]. It can be approximated by using the following expression for the channel charge of a NMOS transistor while in the ON state:

$$Q_{ch} = -C_{ox} \cdot W \cdot L \cdot \left[V_{DD} - V_i - V_T\left(V_i - V_{SS}\right)_i\right] \qquad \textbf{Eq. 121}$$

Where $V_T\left(V_i - V_{SS}\right)$ models the body-effect:

$$V_T\left(V_{SB}\right) = V_{T0} + \gamma \cdot \left(\sqrt{\phi_B + V_{SB}} - \sqrt{\phi_B}\right) \qquad \textbf{Eq. 122}$$

V_{T0} is the threshold voltage for $V_{SB} = 0$, γ is the body-effect constant, ϕ_B is the surface potential in strong inversion and V_{SB} is the source-bulk voltage drop. From here, the change in the stored voltage happens to be:

$$\Delta V_{ch} = -\delta \cdot \frac{C_{ox} \cdot W \cdot L}{C_{mem}}\left[V_{DD} - V_i - V_T\left(V_i - V_{SS}\right)\right] \qquad \textbf{Eq. 123}$$

where δ represents the undetermined fraction of the channel charge that is injected into the memory capacitor $\left(0 < \delta < 1\right)$. For design purposes, the worst case $\delta = 1$ can be considered. This means that 100% of the channel charge gets injected into the memory capacitor.

CLOCK FEEDTHROUGH. For the second phenomenon, let us consider the capacitive divider formed by the gate-to-drain overlap parasitic, C_{gd}, and the memory capacitor C_{mem}. When the input signal, V_i, is no longer driving the storage node, because the switch have been turned OFF, the control signal, $V_{control}$ may cause some extra voltage degradation by being fed through capacitive coupling to the sampling capacitor [Weiz02]. Once the control signal reaches the level indicated in the following equation:

$$V_{control} = V_i + V_T\left(V_i - V_{SS}\right) \qquad \textbf{Eq. 124}$$

The switch turns OFF and the voltage at the memory capacitor does not follow the input signal V_i. But the control signal still goes down to the power supply V_{SS}, affecting to the stored voltage through the parasitic capacitance between the gate and the source terminals. This results in an error in the stored voltage that can be expressed by:

$$\Delta V_{feed} = \frac{C_{gd}}{C_{mem} + C_{gd}} \left[V_i + V_T \left(V_i - V_{SS} \right) - V_{SS} \right] \qquad \textbf{Eq. 125}$$

Both error contributions are *signal dependent*, and therefore, produce *harmonic distortion* in the more general case. However, in the actual memory cell used for the Q-Eye, the signal dependence is removed and only offset errors remains thanks to the control applied during the writing phase. The injected charge is basically generated when SW2 turns OFF. SW1 is turned OFF after that SW2 is fully OFF. The voltages at terminals of SW2 transistor remain constant always during writing phase. Therefore, the error introduced by SW2 is signal-independent introducing an offset component.

Still these offset errors have to be considered in the design phase because they produce *spatial noise* in the images (FPN) due to *mismatch*. In order to reduce this error large capacitance values must be considered. But large capacity values compromises analog circuit speed. Consequently, there is a tradeoff between feedthrough/injection charge error (attainable accuracy) and speed requirements. In addition, large areas associated to the devices of processing cell imply a degradation of resolution and *fill factor* parameters; therefore there is a tradeoff between area (fill factor, resolution) and accuracy.

3.3.1.3. OFFSET AND FINITE GAIN OF AMPLIFIERS

Offset insensitiveness of Figure 37 is valid only into first order approach. Besides the residual offset created by non-ideal switching, the finite OTA DC gain yields incomplete offset voltage cancellation. It can be readily calculated with the help of Figure 43 and yields:

$$V_{out} \approx \frac{1}{1+\varepsilon} \left(V_{in} + \varepsilon \cdot V_{os} \right) \qquad \textbf{Eq. 126}$$

where:

$$\varepsilon = \frac{1}{A_o} \left(1 + \frac{C_{in}}{C_{mem}} \right) \qquad \textbf{Eq. 127}$$

Considering $\varepsilon << 1$, we obtain:

$$V_{out} = \left(1 - \varepsilon \right) V_{in} + \varepsilon \cdot V_{os} \qquad \textbf{Eq. 128}$$

Showing that the errors associated to the finite gain and offset of the operational amplifier appear as a gain-error and an offset in the read data. These non-idealities provoke fixed pattern noise in the image stored locally in the memories of cell array, see Section 3.4.

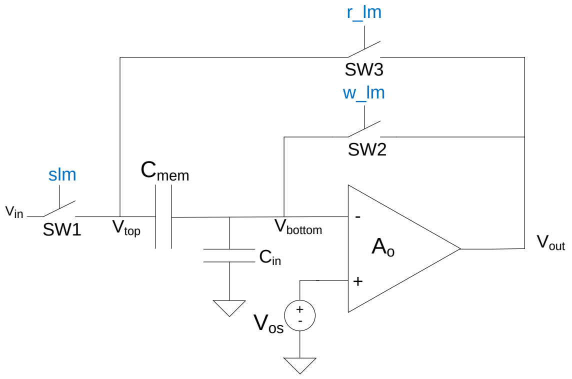


Figure 43. NON IDEALITIES OF OPERATIONAL AMPLIFIER.

3.3.1.4. NOISE LIMITATIONS IN THE MEMORIES

The electronic noise of passive and active components produces random contributions to the errors in the stored voltages and hence impacts the behavior of the Q-Eye processing cell.

Figure 44 illustrates the static models used for the evaluation of each noise source during the writing and reading phases. The **ON-resistances of switches** SW1, SW2 and SW3 generate a thermal, white noise described respectively by the following Power Spectral Densities (PSD):

$$S_{n\_SW1}(f) = v_{n\_SW1}^2 = 4k_B T R_{SW1} \qquad \textbf{Eq. 129}$$

$$S_{n\_SW2}(f) = v_{n\_SW2}^2 = 4k_B T R_{SW2} \qquad \textbf{Eq. 130}$$

$$S_{n\_SW3}(f) = v_{n\_SW3}^2 = 4k_B T R_{SW3} \qquad \textbf{Eq. 131}$$

Where k_B is the Boltzmann's constant $\left(1.38 \cdot 10^{-23} JK^{-1}\right)$, T is the absolute temperature (Kelvin) and R_{SW1}, R_{SW2} and R_{SW3} are the on-resistance of switches.

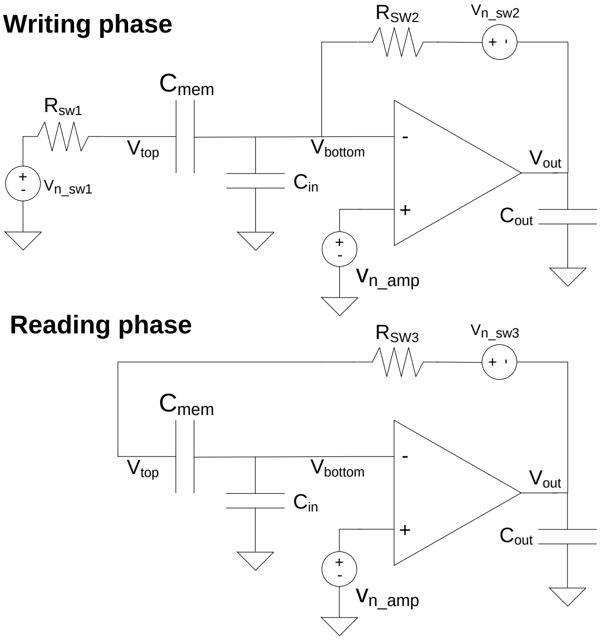


Figure 44. NOISE SOURCES IN ANALOGUE MEMORY PHASES.

Regarding the **amplifier**, its equivalent input noise is modeled by a voltage source v_{n_amp} at the positive input terminal. This is mostly due to the MOSFETs input transistors and has a thermal and a flicker component. The input-referred thermal noise of a MOSFET transistor in saturation is approximately:

$$S_{white\_MOS}(f) = \frac{8}{3} k_B T \frac{1}{g_m} \qquad 0 < f < +\infty \qquad \textbf{Eq. 132}$$

In addition, the flicker (or 1/f) noise, which is nearly independent of the bias condition, is approximately given by:

$$S_{1/f\_MOS}(f) = \frac{K}{C_{ox} WL} \cdot \frac{1}{|f|} \qquad 0 < f < +\infty \qquad \textbf{Eq. 133}$$

Where K is a process and temperature dependent parameter.

The thermal noise PSD function referred to the amplifier input in CMOS technology can be generally described by:

$$S_{white\_amp}(f)=v^2_{white\_amp}=2\cdot\frac{8k_BT}{3g_m}\cdot(1+\eta_{amp})$$ **Eq. 134**

g_m is the amplifier transconductance and η_{amp} is a factor depending on the particular topology.

Both noises (thermal and flicker) are uncorrelated in the same device, so the total noise contribution is given by the direct addition of their PSDs. The total PSD have the generic shape shown in Figure 45.

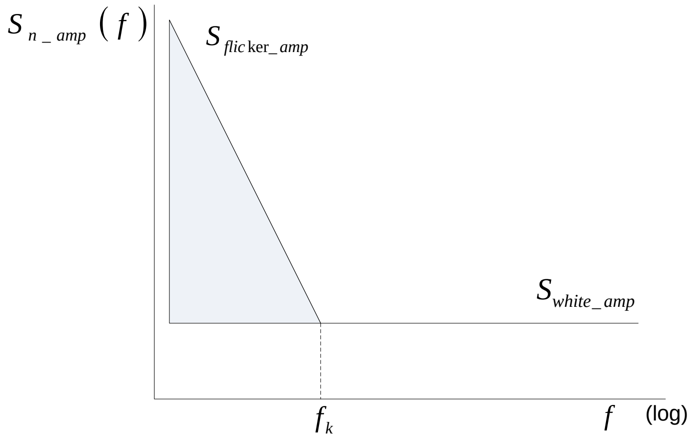


Figure 45. **PSD OF THE AMPLIFIER NOISE.**

The frequency at which flicker noise is equal to the thermal noise is called corner frequency f_k. Below f_k, flicker noise dominates while thermal noise dominates beyond f_k [Raza01]. Taking this into account, the PSD of amplifier noise, referred to its input, can be written as:

$$S_{n\_amp}(f)=v^2_{n\_amp}=S_{white\_amp}\left(1+\frac{f_k}{|f|}\right)=S_{white\_amp}+S_{1/f\_amp}$$ **Eq. 135**

$$S_{1/f\_amp}=\frac{S_{white\_amp}\cdot f_k}{|f|}$$ **Eq. 136**

Considering the different noise contributions presented above, the error of charge trapped into the memory capacitor at the end of the writing phase is given by:

$$Q_{mem}(wr)=C_{mem}\left(v_{in}+v_{n\_SW1}-V_{os}-v_{n\_amp}(wr)\cdot H^{wr}_{amp}\right)$$ **Eq. 137**

Where the H^{wr}_{amp} is the transfer function corresponding to the amplifier in a unit-gain configuration:

$$H^{wr}_{amp}=\frac{1}{1+\tau_{wr}\cdot s}$$ **Eq. 138**

And we have assumed a single-pole model with infinite DC-gain for the amplifier:

$$H_{amp}(s)=\frac{A_0}{1+\tau\cdot s}\qquad A_0\to\infty$$ **Eq. 139**

The noise source v_{n_sw2} does not affect to the charge stored in the memory capacitor because the V$_{bottom}$ node is not influenced by this source during writing phase due to the feedback loop configuration.

Furthermore, the charge stored in the memory capacitor during reading phase is:

$$Q_{mem}(rd)=C_{mem}\left(v_{out}+v_{n\_SW3}-V_{os}-v_{n\_amp}(rd)\right) \quad \textbf{Eq. 140}$$

Taking into account the charge conservation law between the writing and reading phases, the noise at the output of analogue memory during reading phase is described by following equations:

$$Q_{mem}(wr)=Q_{mem}(rd) \quad \textbf{Eq. 141}$$

$$v_{out}(rd)=v_{in}+v_{n\_SW1}+v_{n\_SW3}-v_{n\_amp}(wr)\cdot\frac{1}{1+\tau_{wr}\cdot s}+v_{n\_amp}(rd) \quad \textbf{Eq. 142}$$

Denoting:

$$v_{n\_AZ}=v_{n\_amp}(rd)-v_{n\_amp}(wr)\cdot\frac{1}{1+\tau_{wr}\cdot s} \quad \textbf{Eq. 143}$$

and assuming that all noise sources are no-correlated, the PSD of noise at the output of analogue memory in reading phase is:

$$S_{n\_out}(f)=v_{n\_out}^{2}=v_{n\_SW1}^{2}+v_{n\_SW3}^{2}+v_{n\_AZ}^{2} \quad \textbf{Eq. 144}$$

where, $v_{n\_AZ}^{2}$ is the PSD resulting from the auto-zeroing operation.

The analogue memory topology implements a basic technique used to reduce the offset associated to the amplifier, the Correlated Double Sampling (CDS) technique. This can be described as an auto-zeroing (AZ) operation followed by a S&H process [Enz96]. The basic idea of AZ process is sampling the unwanted offset and noise and then subtracting it from the instantaneous value of the contaminated signal. v_{n_AZ} represents the resulting noise of AZ operation.

The AZ technique [Wieg89] has a baseband transfer function which imposes a zero at the origin of frequency that cancels any offset and strongly reduces the 1/f noise. In addition to this basic high-pass filtering process, if the AZ is a sampling process, the wideband noise of amplifier is aliased down to the baseband, increasing the resulting in-band power spectral density. The resulting PSD $S_{n\_AZ}(f)$ of the AZ operation can be decomposed into two components: one composed by the baseband noise (which is reduced by the AZ process) and the other by the fold-over components introduced by aliasing associated to the periodic sampling of noise at the input of amplifier to compensate the input offset:

$$S_{n\_AZ}(f)=v_{n\_AZ}^{2}=\left|H_{0}(f)\right|^{2}\cdot S_{n\_amp}^{sampled}(f)+S_{fold}(f) \quad \textbf{Eq. 145}$$

$S_{n\_amp}^{sampled}$ is the first-order low-pass filtered PSD of input-referred noise of amplifier, when the noise is sampled with the amplifier in a unity-gain configuration:

$$S_{n\_amp}^{sampled}(f)=S_{n\_amp}(f)\cdot\left|H_{amp}^{wr}(f)\right|^{2}=\frac{S_{n\_amp}(f)}{1+\left(\dfrac{f}{f_{c\_wr}}\right)^{2}} \qquad f_{c\_wr}=\frac{1}{2\cdot\pi\cdot\tau_{wr}} \quad \textbf{Eq. 146}$$

Where f_{c_wr} is the 3-dB noise bandwidth for the writing configuration.

The fold-over component results from the replicas of the original spectrum of noise at the input of amplifier shifted by the integer multiples of sampling frequency of amplifier offset:

$$S_{fold}(f)=\sum_{\substack{n=-\infty \\ n\neq 0}}^{n=\infty}\left|H_{n}(f)\right|^{2}\cdot S_{n\_amp}^{sampled}\left(f-\frac{n}{T_{s}}\right) \quad \textbf{Eq. 147}$$

Where T_s is the offset sampling period, and the square magnitude of the corresponding band transfer function is expressed by:

$$|H_n(f)|^2 = d^2\left\{\left[\frac{\sin(2\pi dn)}{2\pi dn} - \frac{\sin(2\pi dfT_s)}{2\pi dfT_s}\right]^2 + \left[\frac{1-\cos(2\pi dn)}{2\pi dn} - \frac{1-\cos(2\pi dfT_s)}{2\pi dfT_s}\right]^2\right\} \qquad \textbf{Eq. 148}$$

$$d = \frac{T_h}{T_s} \qquad T_s = T_{AZ} + T_h \qquad \textbf{Eq. 149}$$

T_{AZ} is the time interval to sample the amplifier offset and T_h is the hold time.

The baseband transfer function $|H_0(f)|^2$ is given by:

$$|H_0(f)|^2 = d^2\left\{\left[1 - \frac{\sin(2\pi fT_h)}{2\pi fT_h}\right]^2 + \left[\frac{1-\cos(2\pi fT_h)}{2\pi fT_h}\right]^2\right\} \qquad \textbf{Eq. 150}$$

$H_0(f)$ shows a high-pass characteristic. This means that the baseband transfer function imposes a zero at the origin of frequency which cancels the amplifier offset and reduces the 1/f noise.

The CDS operation can be described as an AZ operation follow by a sample and hold operation. Therefore, the signal at the output of a circuit using CDS is sampled and held. Despite this, the effect of CDS on the amplifier offset and noise is very similar to that of the AZ operation. The baseband transfer function implements a high-pass filtering operation in the same way the AZ does. On the other hand, although the band transfer functions for $n \neq 0$ are different from those obtained for AZ process, the fold-over component provoked by aliasing is comparable because the wideband noise has already been sampled once.

Taking into account that the input-referred noise PSD of amplifier contains both a white and a 1/f noise component,

$$S_{n\_amp}^{sampled}(f) = \left(S_{white\_amp} + S_{1/f\_amp}\right)\cdot\left|H_{amp}(f)\right|^2 = S_{white\_amp}^{sampled} + S_{1/f\_amp}^{sampled} \qquad \textbf{Eq. 151}$$

The PSD of resulting voltage from AZ process can be expressed by:

$$S_{n\_AZ}(f) = S_{n\_AZ}^{white}(f) + S_{n\_AZ}^{1/f}(f) \qquad \textbf{Eq. 152}$$

$$S_{n\_AZ}^{white}(f) = |H_0(f)|^2 \cdot S_{white\_amp}^{sampled} + S_{fold\_white} \qquad \textbf{Eq. 153}$$

$$S_{n\_AZ}^{1/f}(f) = |H_0(f)|^2 \cdot S_{1/f\_amp}^{sampled} + S_{fold\_1/f} \qquad \textbf{Eq. 154}$$

If the amplifier's broadband white noise $S_{white\_amp}^{sampled}$ during the writing phase is considered as an ideally low-pass filtered white noise having an equivalent noise bandwidth $BW_{eq\_amp}^{wr}$, the aliasing effect introduced by the sampling process implies that the original noise power spectrum is shifted by multiples of sampling frequency and summed, resulting:

$$S_{fold\_white}(f) = \left(2 \cdot BW_{e\_amp}^{wr} \cdot T_s - 1\right) \cdot S_{white\_amp} \cdot \sin c^2(\pi \cdot f \cdot T_s) \qquad \textbf{Eq. 155}$$

The equivalent noise bandwidth associated to the white component of amplifier's noise $S_{white\_amp}^{sampled}$ is defined by:

$$BW^{wr}_{eq\_amp} = \frac{1}{S^{2}_{white\_amp}} \int_{0}^{\infty} \left| S^{sample}_{white\_amp}(f) \right|^{2} df = \int_{0}^{\infty} \frac{1}{1+\left(\frac{f}{f_{c\_wr}}\right)^{2}} df = \frac{\pi}{2} f_{c\_wr} = \frac{1}{4 \cdot \tau_{wr}} \tag{Eq. 156}$$

A similar analysis can be developed for the first-order low-pass filtered 1/f noise component $S^{sampled}_{1/f\_amp}$. The fold-over component for the 1/f noise can be approximated in the Nyquist range by:

$$S_{fold\_1/f}(f) \approx 2 \cdot S_{white\_amp} \cdot f_{k} \cdot T_{s} \left[1 + \ln\left(\frac{2}{3} \cdot f_{c\_wr} \cdot T_{s} \right) \right] \cdot \sin c^{2}(\pi \cdot f \cdot T_{s}) \tag{Eq. 157}$$

The folded-back component of the 1/f noise will be generally submerged by the aliased white noise [Enz96] [Fisc82], [Gobe83], so that it can be neglected for noise computation in most practical cases.

Therefore, the PSD of resulting voltage from AZ process is given by:

$$S_{n\_AZ}(f) \approx \left| H_{0}(f) \right|^{2} \cdot S^{sampled}_{n\_amp}(f) + S_{fold\_white}(f) \tag{Eq. 158}$$

If the under-sampling factor $2 \cdot BW_{eq\_amp} \cdot T_{s}$ is much larger than unit, since the baseband term $\left| H_{0}(f) \right|^{2}$ is bounded by 1.6, the autozero white noise $S^{white}_{n\_AZ}$ is dominated by the aliased broadband noise component:

$$S_{n\_AZ}(f) \approx S_{fold\_white}(f) \approx 2 \cdot BW_{eq\_amp} \cdot T_{s} \cdot S_{white\_amp} \cdot \sin c^{2}(\pi \cdot f \cdot T_{s}) \tag{Eq. 159}$$

Considering the effect of CDS operation over the amplifier's noise and the periodic operation of analogue memory (periodic sampling of offset), the PSD of noise at the output of analogue memory in reading phase is:

$$S_{n\_out}(f) = S_{n\_SW1}(f) + S_{n\_SW3}(f) + S_{fold\_white}(f) \tag{Eq. 160}$$

Under these conditions, the total power of noise at the output of analogue memory is:

$$P_{out} = \int_{0}^{\infty} S_{n\_out}(f) df = S_{n\_SW1} \cdot BW_{eq\_SW1} + S_{n\_SW3} \cdot BW_{eq\_SW3} + \int_{0}^{\infty} S_{n\_AZ}(f) df \tag{Eq. 161}$$

Taking into account that:

$$\int_{0}^{\infty} \sin c^{2}(\pi \cdot T_{s} \cdot f) df = \frac{1}{\pi \cdot T_{s}} \cdot \frac{\pi}{2} = \frac{1}{2 \cdot T_{s}} \tag{Eq. 162}$$

The following equation is obtained to describe the total power of noise:

$$P_{out} = S_{n\_SW1} \cdot BW_{eq\_SW1} + S_{n\_SW3} \cdot BW_{eq\_SW3} + S_{white\_amp} \cdot BW^{wr}_{eq\_amp} \tag{Eq. 163}$$

In general, the particular transfer function of each noise source to the output of circuit has a great complexity which depends on the considered resistive and capacitive parasitic. Moreover, the equivalent bandwidth for each noisy source differs from each other. This complex behavior can be approximated suppressing the impact of the switch on resistance in the determination of the transfer functions. Under this conditions, the system behaves as a single pole transfer function with a common equivalent bandwidth for all the noisy sources.

In the case of the analogue memory, the transfer function has a different pole in the writing phase and reading phase:

$$P_{out} = \left(S_{n\_SW1} + S_{white\_amp} \right) \cdot BW^{wr}_{eq\_amp} + S_{n\_SW3} \cdot BW^{rd}_{eq\_amp} \tag{Eq. 164}$$

The equivalent bandwidths for the writing and reading configurations are defined as:

$$BW_{eq\_amp}^{wr} = \int_0^{\infty} \left| H_{amp}^{wr}(f) \right|^2 df = \int_0^{\infty} \frac{1}{1+(2 \cdot \pi \cdot \tau_{wr} \cdot f)^2} df = \frac{1}{4 \cdot \tau_{wr}} = \frac{g_m}{4 \cdot C_{eq\_wr}} \qquad \textbf{Eq. 165}$$

$$BW_{eq\_amp}^{rd} = \int_0^{\infty} \left| H_{amp}^{rd}(f) \right|^2 df = \int_0^{\infty} \frac{1}{1+(2 \cdot \pi \cdot \tau_{rd} \cdot f)^2} df = \frac{1}{4 \cdot \tau_{rd}} = \frac{g_m}{4 \cdot C_{eq\_rd}} \qquad \textbf{Eq. 166}$$

Consulting the equations Eq. 117 and Eq. 119 the equivalent capacitances are:

$$C_{eq\_wr} = C_o + C_{in} + C_{mem} \qquad \textbf{Eq. 167}$$

$$C_{eq\_rd} = C_{in} + C_o \cdot \left(1 + \frac{C_{in}}{C_{mem}} \right) \qquad \textbf{Eq. 168}$$

Therefore, the total power of noise at the output of analogue memory is:

$$P_{out} = \left[4 \cdot k_B \cdot T \cdot R_{SW1} + 2 \cdot (1 + \eta_{amp}) \frac{8 \cdot k_B \cdot T}{3 \cdot g_m} \right] \cdot \frac{g_m}{4 \cdot C_{eq\_wr}} + 4 \cdot k_B \cdot T \cdot R_{SW3} \cdot \frac{g_m}{4 \cdot C_{eq\_rd}} \qquad \textbf{Eq. 169}$$

The S&H circuits are design so that their settling is constrained by the amplifier bandwidth and not by the switch resistances. Under this condition, the following approximation can be done:

$$P_{out} \approx (1 + \eta_{amp}) \cdot \frac{4}{3} \cdot \frac{k_B \cdot T}{C_{eq\_wr}} \qquad \textbf{Eq. 170}$$

This power of noise at the output can be reduced by using a larger sampling capacitor. This is exactly the opposite of what can be made for improving the acquisition time. Therefore, there is a tradeoff between accuracy and operation speed.

The random electronic noise is a fundamental parameter in Vision Systems, Cameras and Image sensors because determines the maximum Signal to Noise Ratio (SNR) and Dynamic Range (DR) attainable by the system.

In general, like we can observe in the study of random noise in the analogue memory, a tradeoff between power consumption and random noise exits in the design of the mixed-signal blocks which composes the Q-Eye processing cell. In a similar way, there is a tradeoff between area and random noise figure.

3.3.1.5. LEAKAGES IN HOLD MODE

The circuit of Figure 37 assumes one amplifier per memory data. Bear in mind, however that in the LAM employed at the Q-Eye the opamp is shared by all the memories in the LAM – see Figure 10. It means that reading does not happen immediately after writing, but the data remain stored until they are read upon demand of the processing algorithm. Figure 46 depicts the hold phase configuration considering only one memory element for simplicity. During the hold period, leakage currents of the switches SW2 and SW4 may degrade the stored voltage. However, the former is not relevant because to bottom node voltage is set by the opamp feedback action. Actually, this switch remains ON to preclude crosstalk between data stored in the analog memories happening due to diffusion leakage associated to the bottom plates of capacitors. Regarding the leakage due to the SW4, it has two components, namely:

- The OFF-current due to sub-threshold conduction (I_{sub}):

$$I_{sub} = I_{DS} = I_0 e^{\frac{q \cdot V_{GS}}{k_B T}} \left(1 - e^{\frac{q \cdot V_{GS}}{k_B T}} \right) \qquad \textbf{Eq. 171}$$

It is minimized by applying a reference voltage V$_{\text{hold}}$ to control the voltage drop between the drain and the source terminals. The SW4 switch is implemented by a low threshold voltage transistor in order to optimize the signal range of the processing cell. Therefore its sub-threshold conduction is not-negligible. This sub-threshold current depends on drain-source voltage (VDS) too. Then, with the reference voltage V_{hold} the sub-threshold current is control through the drain-source voltage of SW4 transistor.

- The reverse-current (I_{pn}) of the reverse-biased junction formed by the n^+ diffusion of SW4 and the p-type substrate:

$$I_{pn} \approx q \cdot A \cdot \left(\frac{D_p p_{n0}}{L_p} + \frac{D_n n_{p0}}{L_n} \right) \quad \textbf{Eq. 172}$$

Where, q is the charge of an electron ($1.602 \cdot 10^{-19} C$), A is the pn-junction area, D_p and D_n are the diffusion coefficients for holes and electrons, L_p and L_n their diffusion lengths and p_{n0} and n_{p0} are the minority-carrier concentrations in each side of junction.

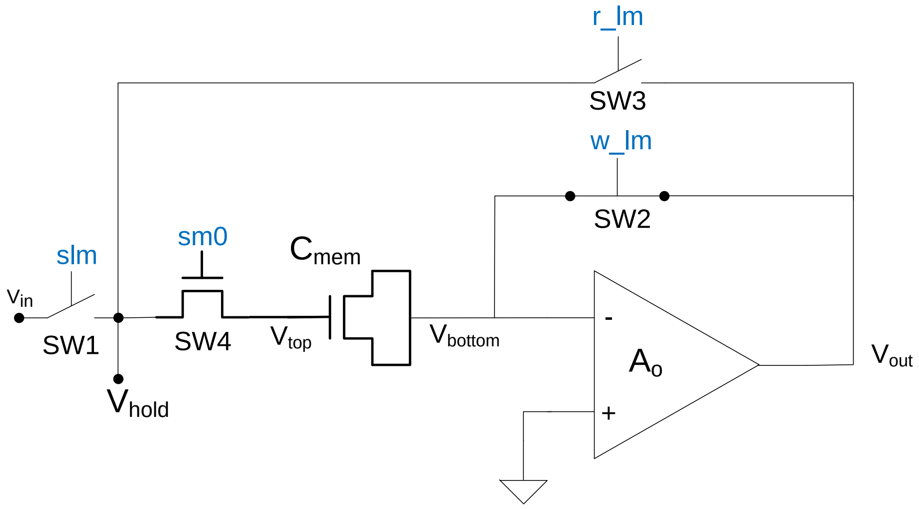


Figure 46. HOLD PHASE CONFIGURATION OF ANALOGUE MEMORY.

The injected charge is given by:

$$\Delta Q_{leak} = I_{leak} \cdot \Delta t_{off} \quad \textbf{Eq. 173}$$

Where I_{leak} is the leakage current and Δt_{off} is the time interval during which the current is integrated. This interval is the time during which the selection switch is OFF. The voltage error of the output data during the reading phase generated by the injected charge is:

$$\Delta V_{leak} = \frac{\Delta Q_{leak}}{C_{mem}} = \frac{I_{leak} \cdot \Delta t_{off}}{C_{mem}} \quad \textbf{Eq. 174}$$

During the **reading phase**, the reverse current generated by the diffusion diode associated to the bottom of the MOS capacitor also impacts on the stored charge. The bottom node is a pure capacitive node which is discharged by this reverse diffusion-diode current of capacitor during the reading configuration, Figure 47. In this case, the integration time of leakage current is the reading time. Since this reading time is small (around 400 nanoseconds), this leakage component is small and negligible.

It must be stressed that the most significant impact of leakage is not due to voltage deviations themselves (which if uniform can be corrected through a global offset signal) but by their mismatches, which lead to FPN (spatial noise, see read Section 3.4) - very complicated of compensate.

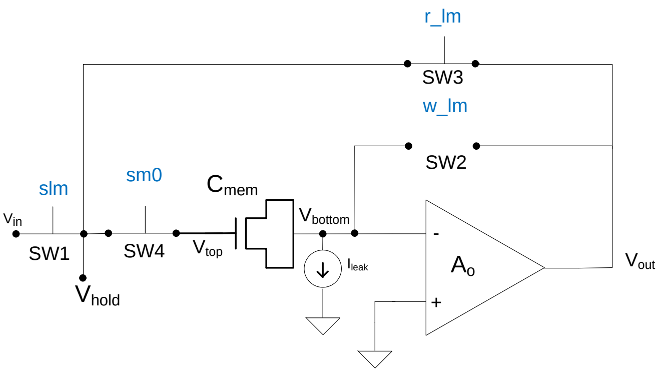


Figure 47. READING PHASE CONFIGURATION.

3.3.2. MAC ERRORS

The MAC being also a SC circuit, its errors are similar to those of the analogue memory.

3.3.2.1. TEMPORAL SETTLING ERROR

Figure 48 shows a conceptual version of the MAC circuit (the actual schematics is shown in Figure 24) for analysis purposes.

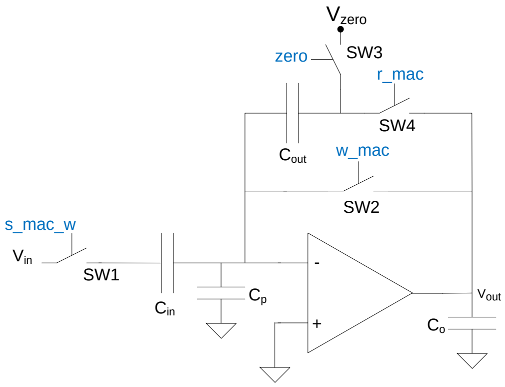


Figure 48. MULTIPLIER-ACCUMULATOR CIRCUIT.

The nominal operation is captured by:

$$V_{out}(phase\_2) = \frac{C_{in}}{C_{out}}(V_1 - V_2) \qquad \text{Eq. 175}$$

where V_1 and V_2 are applied at the input terminal in two consecutive phases (phase 1 and phase 2, respectively) – see Section 3.2.5. The phase 3 represented in the timing diagrams of Figure 25 corresponds to the connection of MAC output to the internal data bus of processing cell **ladn**. Therefore, according to the errors contributions associated to the MAC operation, this phase is not relevant.

Figure 49 shows the circuit model for **phase 1**. Using assumptions similar to those employed for analog memories (Section 3.3.1.1, switch ON-resistances [10] can be neglected and the dynamic

[10] In order to minimize the resistance of switches during design phase, the widths of corresponding transistors must be increased with respect to the minimum value. The increment of transistor areas to implement these switches implies that the charge-injection errors and feedthrough increase as well. Therefore, there is a tradeoff between the time constant associated to MAC switches and charge- injection / feedthrough errors introduced by them. If some resistances of switches cannot be made negligible due to the charge-injection /feedthrough errors introduced in the MAC operation, the resulting transfer function will be higher order, but the effects of settling errors will be similar to the case of a first order behaviour. Therefore, we can study the effect of settling errors in the MAC operation by considering the first order dynamic behaviour.

associated to the amplifier is dominant), the MAC circuit behaves as a single pole system described by:

$$\frac{V_{bottom}(s)}{V_{ref}(s)} = H_{phase\_1}(s) = \frac{1}{\frac{\left(C_{in}+C_{out}+C_o+C_p\right)}{g_m}s+1} = \frac{1}{\tau_{phase\_1}\cdot s+1} \qquad \textbf{Eq. 176}$$

Then, following the application of a unit step of amplitude ΔV_1, the settling error in phase 1 of MAC operation can be defined as follows:

$$v_\varepsilon(t) = \varepsilon_1 = -\Delta V_1 \cdot e^{-\frac{t}{\tau_{phase\_1}}} \qquad \textbf{Eq. 177}$$

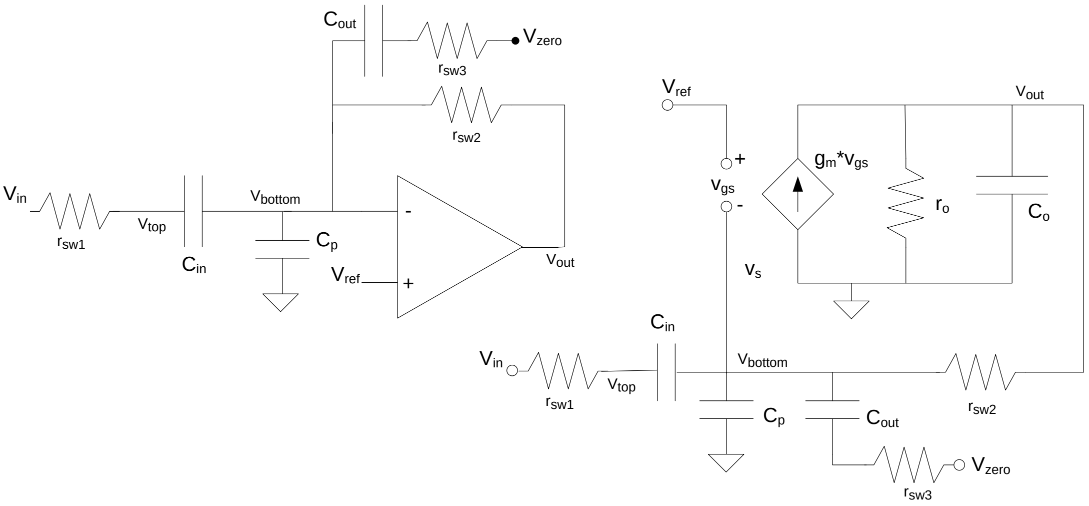


Figure 49. CIRCUITAL MODEL FOR TEMPORAL RESPONSE IN PHASE 1.

Consider now **phase 2** and the corresponding model of Figure 50:

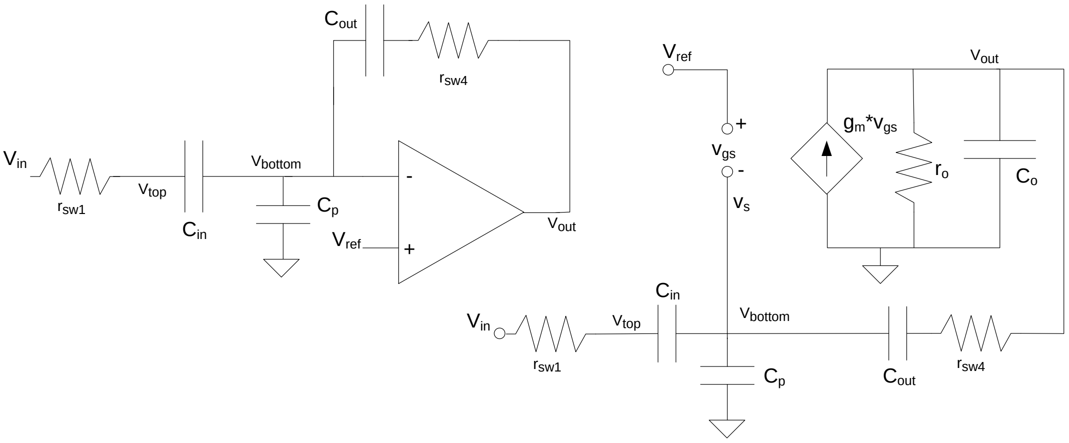


Figure 50. CIRCUITAL MODEL FOR TEMPORAL RESPONSE IN PHASE 2.

Under similar assumptions as for previous phase, the transfer function is calculated as:

$$\frac{v_{out}(s)}{v_{ref}(s)}=H_{phase\_2}(s)=\frac{1+\frac{C_{in}+C_p}{C_{out}}}{\frac{C_{eq\_phase\_2}}{g_m}s+1}=\frac{1+\frac{C_{in}+C_p}{C_{out}}}{\tau_{phase\_2}\cdot s+1} \qquad \text{Eq. 178}$$

with the equivalent capacitance being:

$$C_{eq\_phase\_2}=C_{in}+C_p+C_o\cdot\left(1+\frac{C_{in}+C_p}{C_{out}}\right) \qquad \text{Eq. 179}$$

Consequently, the settling error for a step input of size ΔV_2 is:

$$v_\varepsilon(t)=\varepsilon_2=-\Delta V_2\cdot e^{-\frac{t}{\tau_{phase\_2}}} \qquad \text{Eq. 180}$$

These settling errors (Eq. 177 and Eq. 180) impact first the transferred charges and then the output voltages. The charges stored in the MAC capacitors during phase 1 are:

$$Q_{in}(1)=C_{in}\left[V_1-V_{bottom}(1)\right] \qquad \text{Eq. 181}$$

$$Q_{out}(1)=C_{out}\left[V_{zero}-V_{bottom}(1)\right] \qquad \text{Eq. 182}$$

$$Q_p(1)=C_p\left[-V_{bottom}(1)\right] \qquad \text{Eq. 183}$$

While the charges stored during phase 2 are given by:

$$Q_{in}(2)=C_{in}\left[V_2-V_{bottom}(2)\right] \qquad \text{Eq. 184}$$

$$Q_{out}(2)=C_{out}\left[V_{out}-V_{bottom}(2)\right] \qquad \text{Eq. 185}$$

$$Q_p(2)=C_p\left[-V_{bottom}(2)\right] \qquad \text{Eq. 186}$$

Considering charge conservation law in the v_{bottom} node, the output voltage obtained in the MAC operation is described by the equation:

$$V_{out}=V_{zero}+\frac{C_{in}}{C_{out}}(V_1-V_2)-\left(1+\frac{C_{in}+C_p}{C_{out}}\right)\left[V_{bottom}(1)-V_{bottom}(2)\right] \qquad \text{Eq. 187}$$

where:

$$V_{bottom}(1)=V_{ref}+\varepsilon_1 \qquad \varepsilon_1=\Delta V_1 e^{-\frac{t}{\tau_1}} \qquad \text{Eq. 188}$$

$$V_{bottom}(2)=V_{ref}+\varepsilon_2 \qquad \varepsilon_2=\Delta V_2 e^{-\frac{t}{\tau_2}} \qquad \text{Eq. 189}$$

ΔV_1 and ΔV_2 represent the voltage steps generated at the beginning of phases 1 and 2 due to the capacitive coupling between input node of MAC and bottom node of input capacitor (negative terminal of OTA).

Recasting all previous equations yields:

$$V_{out}=V_{zero}+\frac{C_{in}}{C_{out}}(V_1-V_2)+\Delta v_\varepsilon \qquad \text{Eq. 190}$$

$$\Delta v_{\varepsilon} = -\left(1+\frac{C_{in}+C_p}{C_{out}}\right)\left[V_{bottom}(1)-V_{bottom}(2)\right] \qquad \textbf{Eq. 191}$$

Bear in mind that convolution operations require the MAC block to implement consecutive accumulations – see Section 3.2.7. For instance, to accumulate *n* times the difference $(V_1 - V_2)$, the charges stored in the capacitors of MAC during the phase 1 of iteration *n* are obtained by:

$$Q_{in}(1) = C_{in}\left[V_1 - V_{bottom}(1)\right] \qquad \textbf{Eq. 192}$$

$$Q_{out}(1) = C_{out}\left[V_{out}(n-1) - V_{bottom}(1)\right] \qquad \textbf{Eq. 193}$$

$$Q_p(1) = C_p\left[-V_{bottom}(1)\right] \qquad \textbf{Eq. 194}$$

while the charge stored in capacitors during phase 2 of iteration *n* are:

$$Q_{in}(2) = C_{in}\left[V_2 - V_{bottom}(2)\right] \qquad \textbf{Eq. 195}$$

$$Q_{out}(2) = C_{out}\left[V_{out}(n) - V_{bottom}(2)\right] \qquad \textbf{Eq. 196}$$

$$Q_p(2) = C_p\left[-V_{bottom}(2)\right] \qquad \textbf{Eq. 197}$$

Consequently, the output voltage resulting after *n* multiplication-accumulation processes is expressed by the equation derived from charge conservation law:

$$V_{out}(n) = V_{out}(n-1) + \frac{C_{in}}{C_{out}}\cdot(V_1 - V_2) + \Delta v_{\varepsilon} \qquad \textbf{Eq. 198}$$

where the settling error Δv_{ε} in each MAC operation is considered approximately equal. Under these conditions, taking into account that:

$$V_{out}(n-1) = V_{out}(n-2) + \frac{C_{in}}{C_{out}}\cdot(V_1 - V_2) + \Delta v_{\varepsilon} \qquad \textbf{Eq. 199}$$

By extrapolation, the output voltage after *n* accumulations has an error described in the following equation:

$$V_{out}(n) = V_{zero} + n\cdot\frac{C_{in}}{C_{out}}\cdot(V_1 - V_2) + n\cdot\Delta v_{\varepsilon} \qquad \textbf{Eq. 200}$$

It highlights that the MAC settling error gets accumulated.

ΔV_1 and ΔV_2 parameters depend on input voltages V_1, V_2 and reference V_{zero}. Then, the settling error has signal-independent contributions which generate an offset error and signal-dependent contributions which provoke gain error and non-linearity.

This accumulation of errors generates fixed pattern noise in the resulting images, which increases with the number of iterations. This fact must be considered during the development of processing algorithms in order to avoid the implementation of convolution operations which requires a high number of accumulations in the MAC operator.

3.3.2.2. OFFSET AND FINITE GAIN OF AMPLIFIER, CHARGE INJECTION AND CLOCK FEED-THROUGH

Figure 51 is the circuit model for this study. The starting point of the study is the calculation of the charges at the pertinent capacitors (*C_o* is not required because it is driven by the OTA output) by taking into account the finite OTA gain and the parasitic charges fed by the switches. Routine calculations yields (see section 1 of Appendix B):

$$V_{out}=\frac{1}{1+\varepsilon}\cdot\left[V_{zero}+\frac{C_{in}}{C_{out}}\cdot(V_1-V_2)+\frac{\varepsilon}{1+\frac{1}{A_o}}\cdot V_{os}+\frac{1}{C_{out}}\left(Q_{SW2}+Q_{SW3}+Q_{SW1\_ph1}-Q_{SW1\_ph2}\right)\right]\quad\textbf{Eq. 201}$$

with:

$$\varepsilon=\frac{1}{A_o}\left(1+\frac{C_{in}+C_p}{C_{out}}\right)\quad\textbf{Eq. 202}$$

Where Q_{SW1_ph1} is charge introduced by the MOS transistor SW1 because of charge injection and feedthrough effects when it is turned OFF in phase 1; Q_{SW2} and Q_{SW3} are the charges introduced respectively by the MOS transistors SW2 and SW3 at the end of phase 1; and Q_{SW1_ph2} is the charge injected by the switch SW1 at the end of phase 2.

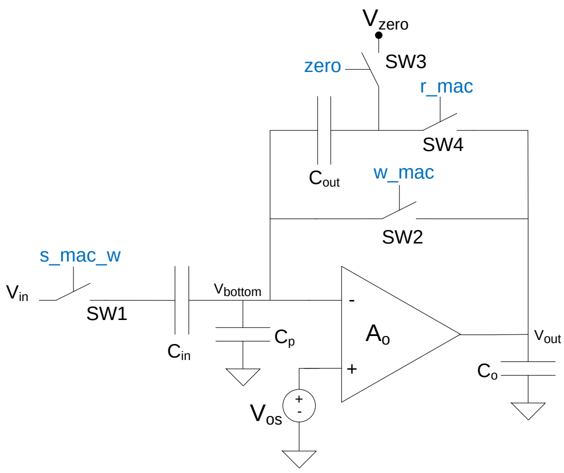


Figure 51. **AMPLIFIER WITH FINITE GAIN AND OFFSET.**

By assuming: $\frac{1}{1+\varepsilon}\approx 1-\varepsilon$, the output voltage is:

$$V_{out}\approx V_{zero}+\frac{C_{in}}{C_{out}}\cdot(1-\varepsilon)\cdot(V_1-V_2)-\varepsilon\cdot\left[V_{zero}-(1-\varepsilon)\cdot V_{os}\right]+(1-\varepsilon)\cdot\frac{Q_{SW3}}{C_{out}}+(1-\varepsilon)\cdot\frac{Q_{SW2}}{C_{out}}+(1-\varepsilon)\cdot\frac{\Delta Q_{SW1}}{C_{out}}\quad\textbf{Eq. 203}$$

where: $\Delta Q_{SW1}=Q_{SW1\_ph1}-Q_{SW1\_ph2}$.

By generalizing to the case of *n* successive accumulations yields:

$$V_{out}\approx(1-\varepsilon)\cdot\left(\sum_{i=1}^{n}(1-\delta)^{i-1}\right)\cdot\left[\frac{C_{in}}{C_{out}}\cdot(V_1-V_2)+\frac{\frac{1}{A_o}}{1+\frac{1}{A_0}}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\frac{\Delta Q_{SW1}}{C_{out}}+\frac{Q_{SW2}}{C_{out}}\right]+$$

$$+(1-\varepsilon)\cdot(1-\delta)^{n-1}\cdot\left(V_{zero}+Q_{SW3}+\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot V_{os}\right)\quad\textbf{Eq. 204}$$

With:

$$\delta = \frac{1}{A_o} \cdot \frac{C_{in} + C_p}{C_{out}} \qquad \textbf{Eq. 205}$$

The term ΔQ_{SW1} has been considered approximately equal in each accumulation.

By taking into account the following approximations:

$$(1-\delta)^{n-1} \approx 1-(n-2)\cdot\delta \qquad \delta << \qquad \textbf{Eq. 206}$$

$$\sum_{i=1}^{n}(1-\delta)^{i-1} = \sum_{i=1}^{n}\sum_{k=0}^{i}\frac{(i-1)!}{k!\cdot(i-1-k)!}(-\delta)^{k} \approx n - \frac{n\cdot(n-1)}{2}\delta \qquad \textbf{Eq. 207}$$

and neglecting second order error terms of, the output voltage after *n* accumulations is:

$$\begin{aligned} V_{out} \approx\ & (1-\varepsilon)\cdot\left(n-\frac{n\cdot(n-1)}{2}\delta\right)\cdot\left[\frac{C_{in}}{C_{out}}\cdot(V_1-V_2)+\frac{\frac{1}{A_o}}{1+\frac{1}{A_0}}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\frac{\Delta Q_{SW1}}{C_{out}}+\frac{Q_{SW2}}{C_{out}}\right]+ \\ & +(1-\varepsilon)\cdot[1-(n-2)\cdot\delta]\cdot\left(V_{zero}+Q_{SW3}+\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot V_{os}\right) \end{aligned} \qquad \textbf{Eq. 208}$$

$$\begin{aligned} V_{out} \approx\ & V_{zero}+n\cdot\left(1-\varepsilon-\frac{n-1}{2}\cdot\delta\right)\cdot\left[\frac{C_{in}}{C_{out}}\cdot(V_1-V_2)+\frac{1}{A_o}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\frac{\Delta Q_{SW1}}{C_{out}}+\frac{Q_{SW2}}{C_{out}}\right]+ \\ & -[\varepsilon-(n-2)\cdot\delta]\cdot V_{zero}+[1-\varepsilon-(n-2)\cdot\delta]\cdot\left(Q_{SW3}+\frac{1}{A_o}\cdot V_{os}\right) \end{aligned} \qquad \textbf{Eq. 209}$$

Therefore, by comparing with the ideal behavior of MAC operation described by the equation:

$$V_{out} = V_{zero} + n\cdot\frac{C_{in}}{C_{out}}\cdot(V_1 - V_2) \qquad \textbf{Eq. 210}$$

We note that the real output data contains:

- an offset term independent from the number of iterations,

$$-(\varepsilon+2\cdot\delta)\cdot V_{zero}+[1-(\varepsilon+2\cdot\delta)]\cdot\left[Q_{SW3}+\frac{1}{A_o}\cdot V_{os}\right]$$

- another offset term dependent of this iteration number.

$$n\cdot\left(1-\varepsilon-\frac{n-1}{2}\cdot\delta\right)\cdot\left[\frac{1}{A_o}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\frac{Q_{SW2}}{C_{out}}\right]$$

- and gain error:

$$n\cdot\left(1-\varepsilon-\frac{n-1}{2}\cdot\delta\right)$$

The offset independent from iteration number is provoked by the charge injected by the switch SW3 and offset of operational amplifier. The offset term dependent of the iteration number is generated by the finite gain associated to the operational amplifier together with the amplifier offset and charge injected by SW2. Moreover, the finite gain of amplifier produces a gain error.

It is worth mentioning that the error term due to the charge injected by the switch SW1 depends on the input voltage. It hence generates gain error and non-linearity. Let us calculate this error. The error introduced by the MOS switch SW1 in each phase of MAC operation is given by:

$$\begin{aligned}\frac{Q_{SW1}}{C_{out}} = V_{SW1} = \Delta V_{ch} + \Delta V_{feed} = -\frac{C_{ox}\cdot W\cdot L}{C_{out}}\left[V_{DD} - V_{in} - V_T\left(V_{in} - V_{SS}\right)\right] - \\ -\frac{C_{in}}{C_{out}}\cdot\frac{C_{gd}}{C_{in}+C_{gd}}\left[V_{in} + V_T\left(V_{in} - V_{SS}\right) - V_{SS}\right]\end{aligned} \quad \textbf{Eq. 211}$$

Consequently:

$$V_{SW1} = k_0 + k_1\cdot V_{in} + k_1\cdot V_T\left(V_{in} - V_{SS}\right) \quad \textbf{Eq. 212}$$

$$k_0 = -\frac{C_{ox}\cdot W\cdot L}{C_{out}}V_{DD} + \frac{C_{in}}{C_{out}}\cdot\frac{C_{gd}}{C_{in}+C_{gd}}V_{SS} \quad \textbf{Eq. 213}$$

$$k_1 = -\frac{C_{ox}\cdot W\cdot L}{C_{out}} + \frac{C_{in}}{C_{out}}\cdot\frac{C_{gd}}{C_{in}+C_{gd}} \quad \textbf{Eq. 214}$$

Therefore:

$$\frac{\Delta Q_{SW1}}{C_{out}} = \Delta V_{SW1} = \frac{Q_{SW1\_ph1}}{C_{out}} - \frac{Q_{SW1\_ph2}}{C_{out}} = V_{SW1\_ph1} - V_{SW1\_ph2} \quad \textbf{Eq. 215}$$

$$\Delta V_{SW1} = k_1\cdot\left(V_1 - V_2\right) + k_1\left[V_T\left(V_1 - V_{SS}\right) - V_T\left(V_2 - V_{SS}\right)\right] \quad \textbf{Eq. 216}$$

This equation shows a non-linearity error due to the body-effect of MOS transistor which is a non-linear phenomenon:

$$V_T\left(V_1 - V_{SS}\right) = V_{To} + \gamma\cdot\left\lfloor\sqrt{\phi_B + \left(V_1 - V_{SS}\right)} - \sqrt{\phi_B}\right\rfloor \quad \textbf{Eq. 217}$$

$$V_T\left(V_1 - V_{SS}\right) = V_{To} + \gamma\cdot\left\lfloor\sqrt{\phi_B + \left(V_1 - V_{SS}\right)} - \sqrt{\phi_B}\right\rfloor \quad \textbf{Eq. 218}$$

Considering $V_{SS} = 0$, the error introduced by the switch SW1 is:

$$\frac{\Delta Q_{SW1}}{C_{out}} = \Delta V_{SW1} = k_1\cdot\left(V_1 - V_2\right) + k_1\cdot\gamma\cdot\left[\sqrt{\phi_B + V_1} - \sqrt{\phi_B + V_2}\right] \quad \textbf{Eq. 219}$$

Developing a small-signal analysis around the zero signal:

$$V_1 = V_{zero} + v_{signal} \quad \textbf{Eq. 220}$$

$$V_2 = V_{zero} \quad \textbf{Eq. 221}$$

The following approximation can be obtained:

$$\gamma\cdot\sqrt{\phi_B + V_1} \approx \gamma\cdot\sqrt{\phi_B + V_{zero}} + \alpha_1\cdot v_{signal} + \alpha_2\cdot v_{signal}^2 + \alpha_3\cdot v_{signal}^3 \quad \textbf{Eq. 222}$$

$$\alpha_1 = \left.\frac{\partial\left(\gamma\cdot\sqrt{\phi_B + V_{zero} + v_{signal}}\right)}{\partial v_{signal}}\right|_{V_{zero}} = \frac{\gamma}{2}\cdot\left(\phi_B + V_{zero}\right)^{-\frac{1}{2}} \quad \textbf{Eq. 223}$$

$$\alpha_2 = \frac{1}{2}\cdot\frac{\partial^2\left(\gamma\cdot\sqrt{\phi_B + V_{zero} + v_{signal}}\right)}{\partial v_{signal}^2}\Bigg|_{V_{zero}} = -\frac{\gamma}{8}\cdot\left(\phi_B + V_{zero}\right)^{-\frac{3}{2}} \qquad \textbf{Eq. 224}$$

$$\alpha_3 = \frac{1}{6}\cdot\frac{\partial^3\left(\gamma\cdot\sqrt{\phi_B + V_{zero} + v_{signal}}\right)}{\partial v_{signal}^3}\Bigg|_{V_{zero}} = \frac{\gamma}{32}\cdot\left(\phi_B + V_{zero}\right)^{-\frac{5}{2}} \qquad \textbf{Eq. 225}$$

Therefore, the output voltage after *n* accumulations at the output of MAC is shown in next equation:

$$\begin{aligned} V_{out} \approx V_{zero} + Offset + n\cdot\left(1-\varepsilon-\frac{n-1}{2}\cdot\delta\right)\cdot\left[1+\frac{C_{out}}{C_{in}}\cdot k_1\cdot(1+\alpha_1)\right]\cdot\frac{C_{in}}{C_{out}}\cdot v_{signal} + \\ + n\cdot\left(1-\varepsilon-\frac{n-1}{2}\cdot\delta\right)\cdot k_1\cdot\left(\alpha_2\cdot v_{signal}^2 + \alpha_3\cdot v_{signal}^3\right) \end{aligned} \qquad \textbf{Eq. 226}$$

Being:

$$\begin{aligned} Offset = -(\varepsilon+2\cdot\delta)\cdot V_{zero} + [1-(\varepsilon+2\cdot\delta)]\cdot\left[Q_{SW3}+\frac{1}{A_o}\cdot V_{os}\right] + \\ + n\cdot\left(1-\varepsilon-\frac{n-1}{2}\cdot\delta\right)\cdot\left[\frac{1}{A_o}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\frac{Q_{SW2}}{C_{out}}\right] \end{aligned} \qquad \textbf{Eq. 227}$$

If the body effect associated to the MOS transistor SW1 is minimized, the corresponding charge injection and feedthrough error introduce only gain error and the non-linear coefficients can be neglected.

3.3.2.3. MISMATCH BETWEEN CAPACITORS

Capacitor mismatch [John97], [Shyu82] produces errors in the capacitance ratio C_{in}/C_{out} that sets the MAC scale factor and hence in the MAC operation. In the Q-Eye system the capacitances C_{in} and C_{out} are programmable:

$$C_{in} = n\cdot C_u \qquad n = 1,2 \qquad \textbf{Eq. 228}$$

$$C_{out} = m\cdot C_u \qquad m = 1,2 \qquad \textbf{Eq. 229}$$

And programmability is implemented as depicted in Figure 52. SW1 is a configuration switch and C_u is the unitary capacitance implemented by a MOS capacitor.

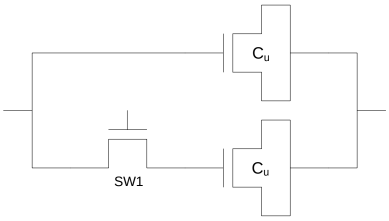


Figure 52. PROGRAMMABLE CAPACITANCE.

Different mismatch phenomena (over-etching phenomenon, lateral diffusion, gradients across the surface of silicon, etc) make the unitary capacitors to change randomly. These changes can be modeled by assuming a Gaussian distribution [Pelg89] [Pelg13] so that variances related with the differences of each capacitance of scale factor are:

$$\frac{\sigma^2(\Delta C_{in})}{C_{in}^2} = \frac{1}{n} \cdot \frac{\sigma^2(\Delta C_u)}{C_u^2} \qquad \textbf{Eq. 230}$$

$$\frac{\sigma^2(\Delta C_{out})}{C_{out}^2} = \frac{1}{m} \cdot \frac{\sigma^2(\Delta C_u)}{C_u^2} \qquad \textbf{Eq. 231}$$

Thus, the scale factor is the presence of mismatch is:

$$\frac{C_{in}}{C_{out}} \approx \frac{n}{m}\left(1 + \sqrt{\frac{1}{n}\frac{\sigma^2(\Delta C_u)}{C_u^2} + \frac{1}{m}\frac{\sigma^2(\Delta C_u)}{C_u^2}}\right) \qquad \textbf{Eq. 232}$$

This equation shows that the mismatch between the capacitors of MAC circuit generates a gain error respect to the nominal scale factor value. In addition, this error is different from cell to cell. Therefore, this mismatch in the gain error among processing cells provoke Fixed Pattern Noise in the images resulting from MAC operations.

The error value depends on the variance of unitary capacitors, which can be calculated by resorting to Pelgrom´s model [Pelg89] as:

$$\frac{\sigma^2(\Delta C_u)}{C_u^2} = \frac{A_{Cox}^2}{C_{ox}^2} \cdot \frac{1}{W_u \cdot L_u} \qquad \textbf{Eq. 233}$$

where the coefficient A_{Cox}^2 is a technological parameter [Maxi01] and the distance term of the model has been neglected upon the assumption that the elements are laid out in close proximity with a layout style [Hast01], that eliminates most sources of systematic errors (e.g., common centroid), read section 2 of Appendix B.

Previous equation shows that the mismatch effect between capacitors decreases when their capacitor area increases. Reducing the error gain generated by the mismatch between capacitors involves an increment in power consumption to maintain the operating speed of MAC circuit, because the polarization current of circuit needs to be increased.

In general, the capacitances associated to the terminals of a MOS transistor depend on the operation point or biasing of this transistor [Cho95]. If the voltage difference $V_{GS} = V_{GD}$ of MOS capacitor is swept, the capacitance between these terminals does not remain constant. It varies depending on the operation region of transistor. A non-linear behavior of MOS capacitors implies that the scale factor changes with signal level generating a non-linear characteristic of MAC operation.

In order to maintain constant the capacitance between the gate and drain-source terminals of MOS capacitor and approximately equal to the value given by the equation:

$$C_u = C_{ox} \cdot W_u \cdot L_u \qquad \textbf{Eq. 234}$$

MOS capacitor is biased in strong inversion region $(V_{GS} >> V_{TH})$.

3.3.2.4. RANDOM NOISE LIMITATIONS IN THE MAC CIRCUIT

This section studies the sources and effects of electronic noise in the Multiplier-Accumulator circuit implemented in the Q-eye processing cell.

Figure 53 illustrates the static models used for the evaluation of each noise source during phase 1 and phase 2.

The noise sources during the phase 1 of MAC operation are the on-resistances associated to the switches SW1, SW2 and SW3 which generate a thermal noise. The power spectral densities which characterize the thermal noise of these switches are given by the equations:

$$S_{n\_SW1}(f)=v_{n\_SW1}^2=4k_BTR_{SW1}$$ **Eq. 235**

$$S_{n\_SW2}(f)=v_{n\_SW2}^2=4k_BTR_{SW2}$$ **Eq. 236**

$$S_{n\_SW3}(f)=v_{n\_SW3}^2=4k_BTR_{SW3}$$ **Eq. 237**

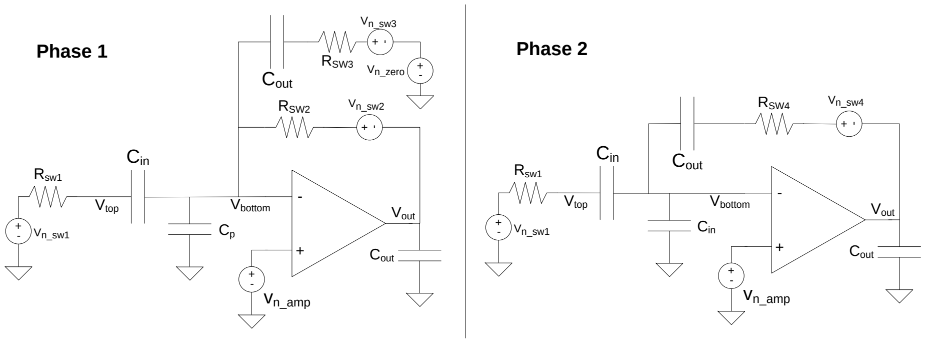


Figure 53. NOISE SOURCES IN MAC BLOCK.

Together with these noise contributions, the reference voltage V_{zero} generated by a Digital Analogue Converter contributes with a noise which usually contains a thermal and flicker (1/f) component. Consequently, the corresponding PSD is:

$$S_{n\_zero}(f)=v_{n\_zero}^2=S_{white\_zero}\left(1+\frac{f_{k\_zero}}{|f|}\right)=S_{white\_zero}+S_{1/f\_zero}$$ **Eq. 238**

The noise of reference voltage V_{zero} can be approximated by a white noise component if the DA converter is conveniently designed with a low corner frequency f_{k_zero}, such that the flicker contribution is negligible.

Finally, the PSD of amplifier noise, referred to its input, can be written as:

$$S_{n\_amp}(f)=v_{n\_amp}^2=S_{white\_amp}\left(1+\frac{f_{k\_amp}}{|f|}\right)=S_{white\_amp}+S_{1/f\_amp}$$ **Eq. 239**

The thermal noise PSD function S_{white_amp} is described by the equation Eq. 134.

Taking into account a single-pole model with infinite DC-gain for the amplifier, see Eq.139, the charge stored in MAC capacitors during phase 1 considering the effect of noise sources is described by the equations:

$$Q_{in}(phase\_1)=C_{in}\cdot\left(V_1+v_{n\_sw1\_ph1}-v_{n\_amp}\cdot H_{amp\_ph1}\right)$$ **Eq. 240**

$$Q_{out}(phase\_1)=C_{out}\cdot\left(V_{zero}+v_{n\_zero}+v_{n\_sw3}-v_{n\_amp}\cdot H_{amp\_ph1}\right)$$ **Eq. 241**

$$Q_p(phase\_1)=C_p\cdot\left(-v_{n\_amp\_ph1}\cdot H_{amp\_ph1}\right)$$ **Eq. 242**

Where, $v_{n\_sw1\_ph1}$ is the random noise of switch SW1 during phase 1 and H_{amp_ph1} is the transfer function for the feedback configuration of amplifier in phase 1 denoted by:

$$H_{amp\_ph1}(f)=\frac{1}{1+\frac{f}{f_{c\_ph1}}} \qquad f_{c\_ph1}=\frac{1}{2\cdot\pi\cdot\tau_{phase\_1}}$$ **Eq. 243**

The noise source v_{n_sw2} does not affect to the charge stored in capacitors because the V$_{bottom}$ node is not influenced by this source during phase 1 due to the feedback loop configuration.

For phase 2 of MAC operation, the random noise is due to the amplifier noise and the thermal noise generated by switches SW1 and SW4. The PSD of SW1 thermal noise is given by Eq 235 and PSD of thermal noise associated to SW4 is:

$$S_{n\_SW4}(f)=v^2_{n\_SW4}=4k_BTR_{SW4} \qquad \textbf{Eq. 244}$$

Therefore, the charge stored in MAC capacitors in phase 2 is indicated by the following equations:

$$Q_{in}(phase\_2)=C_{in}\left(V_2+v_{n\_sw1\_ph2}-v_{n\_amp}\right) \qquad \textbf{Eq. 245}$$

$$Q_{out}(phase\_2)=C_{out}\left(V_{out}+v_{n\_sw4}-v_{n\_amp}\right) \qquad \textbf{Eq. 246}$$

$$Q_p(phase\_2)=C_p\left(-v_{n\_amp}\right) \qquad \textbf{Eq. 247}$$

$v_{n\_sw1\_ph2}$ is the random noise of switch SW1 during phase 2.

Applying the conservation law of charge between both phases, the output voltage resulting from a MAC operation is determined by the equation:

$$V_{out}=V_{zero}+\frac{C_{in}}{C_{out}}\cdot\left(V_1-V_2\right)+v_{n\_out} \qquad \textbf{Eq. 248}$$

$$v_{n\_out}=v_{n\_zero}+v_{n\_sw3}+v_{n\_sw4}+\frac{C_{in}}{C_{out}}\cdot\left(v_{n\_sw1\_ph1}-v_{n\_sw1\_ph2}\right)+\left(1+\frac{C_{in}+C_p}{C_{out}}\right)\cdot\left(v_{n\_amp}-v_{n\_amp}\cdot H_{amp\_ph1}\right)$$

So, the PSD of random noise contained in the output voltage is:

$$S_{n\_out}(f)=v^2_{n\_out}=v^2_{n\_zero}+v^2_{n\_sw3}+v^2_{n\_sw4}+\left(\frac{C_{in}}{C_{out}}\right)^2\cdot v^2_{n\_sw1\_ph1}+$$
$$+\left(\frac{C_{in}}{C_{out}}\right)^2\cdot v^2_{n\_sw1\_ph2}+\left(1+\frac{C_{in}+C_p}{C_{out}}\right)^2\cdot v^2_{n\_AZ} \qquad \textbf{Eq. 249}$$

Where, $v^2_{n\_AZ}$ denotes the PSD resulting from the Autozeroing operation included in the sampled and held process of MAC operation.

Carrying out a similar study developed for the analogue memory, the following equation describing the PSD associated to the Autozeroing process is obtained:

$$v^2_{n\_AZ}=S_{n\_AZ}(f)\approx S_{fold\_white}(f)\approx 2\cdot BW^{ph1}_{eq\_amp}\cdot T_s\cdot S_{white\_amp}\cdot\sin c^2\left(\pi\cdot f\cdot T_s\right) \qquad \textbf{Eq. 250}$$

In this case, T_s the sampling period of amplifier's offset and $BW^{wr}_{eq\_amp}$ the equivalent noise bandwidth during phase 1.

Therefore, the total noise power at the output of MAC circuit is:

$$P_{out}=\int_0^\infty S_{n\_out}(f)df=S_{n\_zero}\cdot BW_{eq\_zero}+S_{n\_SW3}\cdot BW_{eq\_SW3}+S_{n\_SW4}\cdot BW_{eq\_SW4}+$$
$$+\left(\frac{C_{in}}{C_{out}}\right)^2 S_{n\_SW1}\cdot BW^{ph1}_{eq\_SW1}+\left(\frac{C_{in}}{C_{out}}\right)^2 S_{n\_SW1}\cdot BW^{ph2}_{eq\_SW1}+\int_0^\infty S_{n\_AZ}(f)df \qquad \textbf{Eq. 251}$$

Suppressing the impact of the switch on resistance in the determination of the transfer functions, the system behaves as a single pole transfer function with a common equivalent bandwidth for all the noisy sources.

$$P_{out} \approx S_{n\_zero} \cdot BW_{eq\_zero} + \left(S_{n\_SW1} + S_{n\_SW3} + S_{white\_amp}\right) \cdot BW_{eq\_amp}^{ph1} + \left(S_{n\_SW1} + S_{n\_SW4}\right) \cdot BW_{eq\_amp}^{ph2} \quad \textbf{Eq. 252}$$

The equivalent bandwidths for phase 1 and phase 2 configurations are defined as:

$$BW_{eq\_amp}^{ph1} = \int_0^{\infty} \left|H_{amp}^{ph1}(f)\right|^2 df = \int_0^{\infty} \frac{1}{1 + \left(2 \cdot \pi \cdot \tau_{phase\_1} \cdot f\right)^2} df = \frac{1}{4 \cdot \tau_{phase\_1}} = \frac{g_m}{4 \cdot C_{eq\_ph1}} \quad \textbf{Eq. 253}$$

$$BW_{eq\_amp}^{ph2} = \int_0^{\infty} \left|H_{amp}^{ph2}(f)\right|^2 df = \int_0^{\infty} \frac{1}{1 + \left(2 \cdot \pi \cdot \tau_{phase\_2} \cdot f\right)^2} df = \frac{1}{4 \cdot \tau_{phase\_2}} = \frac{g_m}{4 \cdot C_{eq\_ph2}} \quad \textbf{Eq. 254}$$

The equivalent capacitances are given by:

$$C_{eq\_ph1} = C_{in} + C_p + C_o + C_{out} \quad \textbf{Eq. 255}$$

$$C_{eq\_ph2} = C_{in} + C_p + C_o \cdot \left(1 + \frac{C_{in} + C_p}{C_{out}}\right) \quad \textbf{Eq. 256}$$

The MAC circuit has been designed so that their settling is constrained by the amplifier bandwidth and not by the switch resistances. Under this condition, the following approximation can be done:

$$\begin{aligned} P_{out} &\approx S_{n\_zero} \cdot BW_{eq\_zero} + S_{white\_amp} \cdot BW_{eq\_amp}^{ph1} = S_{n\_zero} \cdot BW_{eq\_zero} + \\ &+ 2 \cdot \left(1 + \eta_{amp}\right) \cdot \frac{8 \cdot k_B \cdot T}{3 \cdot g_m} \cdot BW_{eq\_amp}^{ph1} \end{aligned} \quad \textbf{Eq. 257}$$

$$P_{out} \approx S_{n\_zero} \cdot BW_{eq\_zero} + \left(1 + \eta_{amp}\right) \cdot \frac{4 \cdot k_B \cdot T}{3 \cdot C_{eq\_ph1}} \quad \textbf{Eq. 258}$$

This noise power at the output can be reduced by using a larger sampling capacitor. This is exactly the opposite of what can be made for improving the acquisition time. Again we can observe the trade-off among power consumption, area, operation speed and image quality or resolution existing in the design of the mixed-signal blocks which composes the *Q-Eye* processing cell.

The results obtained in the studies developed for the analogue memory and MAC circuit show that a larger capacitance implies a lower noise level but it takes a higher power consumption to achieve a particular operation speed. Moreover, greater capacity requires a greater amount of area to be implemented increasing the pixel pitch (or processing cell pitch) and thus decreasing the resolution for a given area of the sensing array.

3.3.3. RESISTIVE GRID ERRORS

The diffusion network is physically implemented with MOS transistors used as resistors which produces errors. The equivalent ON-resistance of a CMOS switch varies as a function of the voltages at the drain-source terminals of transistors. This dependence of resistance with the voltage at network nodes implies that its time constant τ depends on the signal-level. So, since the diffusion length n_{diff} depends on constant time, this means that the diffusion length also depends on the signal-level at the network nodes.

Considering the approximation in Eq.108 to determine the relation between diffusion length and diffusion time one obtains:

$$\frac{t_{diff}}{\tau} = \kappa \cdot n_{diff} \cdot \left(n_{diff} + 1\right) \approx \kappa \cdot n_{diff}^{2} \qquad \textbf{Eq. 259}$$

Where, t_{diff} is the diffusion time and κ a proportionality constant.

The relation between the maximum and the minimum diffusion lengths determined by the maximum and minimum constant time values is given by:

$$\frac{n_{diff}(\max)}{n_{diff}(\min)} = \frac{\sqrt{\dfrac{t_{diff}}{\tau_{\min}}}}{\sqrt{\dfrac{t_{diff}}{\tau_{\max}}}} = \sqrt{\frac{\tau_{\max}}{\tau_{\min}}} \qquad \textbf{Eq. 260}$$

The maximum and minimum values of the constant time are defined by the maximum and minimum resistances corresponding to the maximum and minimum voltages values associated to the input image. Therefore, the diffusion length of the Gaussian filter implemented by the diffusion network varies depending on the signal level of input image.

Also, the Resistive Grid Block has several switches to control the different operations required to carry out a diffusion process. For example, the reset of network nodes, the initialization with the input image and launching of the diffusion process. The resistance of control switches can be considered negligible respect to the diffusion resistance, because during the design phase the switch transistors are implemented wide enough. But, these control switches introduce feed-though and charge-injection errors. The feed-through and charge-injection errors introduced before the diffusion process have not influence in the low pass filter result, because the spatial random component of these errors is averaged by the filter operation, and the systematic offset component in the resulting image can be cancelled. Nevertheless, the feed-through and charge-injection errors introduced by the transistors controlled by the signal **rg_me**, see Figure 32, affects to the low pass filter result once the diffusion process has finished. The value of the error introduced in the diffused image at the end of the diffusion degrades the quality of the latter.

The feedthrough and charge-injection phenomena cause mainly offset and gain errors on the ideal behavior at each pixel or node in the diffusion network as explained in previous sections. The offset and gain error for each pixel have two components, a systematic one which is equal for all pixels and random one due to the mismatch which is different pixel to pixel. This random contribution generates fixed pattern noise in the image resulting from the diffusion process degrading the image quality. In this latter case, the spatial random component due to the mismatch cannot be cancelled.

Finally, the CMOS resistors have thermal noise as the major source of temporal noise. But, the Gaussian filter operation minimizes the effect of this random noise on the resulting image. So, its effect can be neglected.

3.3.4. Errors in The Optical Sensor

Basically, the noise components for CMOS sensors can be classified into two categories:

- random noise and
- *spatial noise* (fixed pattern noise - FPN).

The random noise describes the temporal variation of the output signal; whereas the fixed pattern noise describes the spatial variation of the 2-D imager plane as a uniform illumination is applied. The former is purely probabilistic and the latter is deterministic for given chip instance.

3.3.4.1. Photodiode error contributions

A number of photons hitting the pixel area during the exposure time create a number of electrons forming a charge which is converted into by the capacitor C_{FD} associated to the FD (Floating Diffusion) node to a voltage; this voltage is then buffered for storage – see Figure 54**.** Major errors affecting this voltage are due to leakage and to the corpuscular nature of light.

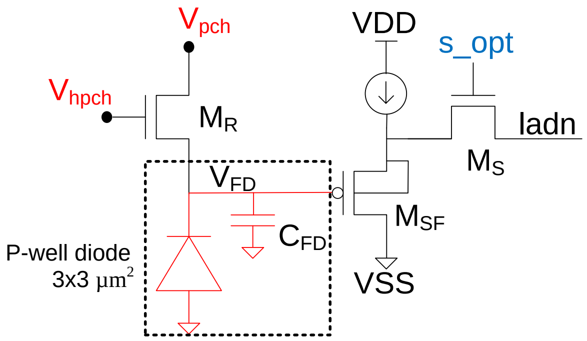


Figure 54. OPTICAL SENSOR BLOCK OF Q-EYE CELL.

DARK CURRENT. In dark conditions, there is a leakage current at the FD node. This leakage is provoked by the reverse-biased saturation currents of the junctions formed by the n-type diffusion areas which implement the FD node and the p-type substrate. The n-type diffusion areas correspond to the photodiode area and source of the reset transistor, see Figure 55.

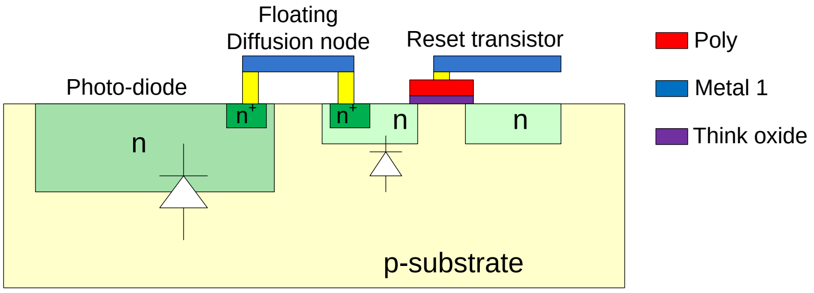


Figure 55. PHYSICAL STRUCTURE.

The dark current pumps charge out of the upper plate of the FD capacitor and produce two effects:

- The mean value of this leakage current for all pixels causes a global and systematic effect on the sensed image which is to reduce the signal swing, the output voltage range of sensor.
- Another systematic and time invariant error introduced is the spatial noise due to the the dark current pixel-to-pixel mismatch. For given exposure time, the different leakage current values in each pixel introduce different offsets per pixels introducing in the sensed image a FPN denoted by Dark Signal Non Uniformity (DSNU).

SHOT NOISE. Both the photo-generated current and the dark current are of discrete nature. Then, the number of electrons stored in the FD capacitor is affected by a temporal noise, the shot noise which is a white noise with an average power described by the equation:

$$q_{shot}^2 = \frac{\left(i_{ph} + i_{dc}\right) \cdot T_{\text{int}}}{q} \qquad \left(electrons^2\right) \qquad \textbf{Eq. 261}$$

Where i_{ph} is the photo-generated current, i_{dc} is the Dark current, T_{int} is the integration or *exposure time* and q is the charge of one electron.

PHOTO-RESPONSE NON-UNIFORMITY. The photo-generated electrons are integrated in the capacitance associated to the FD node performing the charge-voltage conversion which is characterized by the Conversion Gain (CG) parameter:

$$V_{FD} = CG \cdot n_e \qquad \textbf{Eq. 262}$$

Where n_e is the number of photo-generated electrons. Taking into account that:

$$V_{FD} = \frac{q \cdot n_e}{C_{FD}} \qquad \textbf{Eq. 263}$$

the conversion gain parameter is calculated as:

$$CG = \frac{q}{C_{FD}} \qquad \textbf{Eq. 264}$$

being q the electron charge.

The FD capacitance comprises the photodiode capacitance and all other parasitic capacitances connected to that node. In this case, there are the source capacitance of the reset transistor and the gate capacitance of the driver transistor of the source follower. The total capacitance value changes from pixel to pixel due to random process variations thus producing different CG values and hence different pixel responsivities. The overall effect manifests as a FPN for a given uniform illumination denoted as Photo Response Non Uniformity (PRNU).

Other non-ideal effect associated to the Floating Diffusion capacitance is provoked by the non-linear behavior of diffusion capacitances and Source Follower input capacitance. For instance, the capacitance between diffusion regions and substrate has a non-linear behavior described by the equation:

$$C_{bulk\_diff} = \frac{C_j \cdot A_{diff}}{\left(1 - \frac{V_{bd}}{P_B}\right)^{M_J}} + \frac{C_{jsw} \cdot P_{diff}}{\left(1 - \frac{V_{bd}}{P_B}\right)^{M_{JSW}}} \qquad \textbf{Eq. 265}$$

Where C_j, C_{jsw}, M_J, M_{JSW} and P_B are technological parameters, A_{diff} and P_{diff} are respectively the area and perimeter of diffusion and V_{bd} is the difference voltage between bulk and diffusion.

This non-linear behavior of capacitance associated to the floating diffusion node produces a non-linear characteristic in the responsivity curve associated to the pixel. Furthermore, the mismatch of this non-linearity generates FPN in the captured images. However, this non-linearity is usually smaller than the non-linear behavior associated to the output source follower of pixel.

3.3.4.2. RESET TRANSISTOR NOISE CONTRIBUTIONS

In this section, the temporal noise introduced by the reset transistor will be described, see Figure 56. Besides the possibility of controlling the gate voltage of the reset transistor to achieve compressive acquisition, there are to possible reset procedures depending on the voltages applied to the node V_{pch} in case the gate voltage V_{hpch} is set to a high voltage VDD, namely hard and soft reset. In both cases the reset transistor is ON.

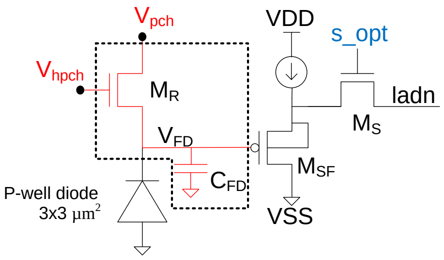


Figure 56. **OPTICAL SENSOR BLOCK OF Q-EYE CELL.**

HARD RESET. This condition occurs when the reset transistor operates in ohmic region during the reset phase. In this case, the conventional frequency domain noise analysis method can be applied and the average power of reset noise is given by the equation:

$$v_{reset}^2 = \frac{k \cdot T}{C_{FD}} \qquad (V^2) \qquad \textbf{Eq. 266}$$

$$q_{reset}^2 = C_{FD}^2 \cdot v_{reset}^2 = k \cdot T \cdot C_{FD} \qquad (electrons^2) \qquad \textbf{Eq. 267}$$

SOFT RESET. If the global reference Vpch is programmed to VDD, the reset transistor is either operating in the saturation region or in subthreshold depending on the FD node voltage. If the FD node voltage is low enough, the reset transistor is in saturation at first and for a very short amount of time before it goes into subthreshold for the rest of reset phase. This reset process is called soft reset. During a soft reset process the reset transistor operates in subthreshold and steady state is not achieved. Then, the conventional frequency domain noise analysis method cannot be applied. The PSD and average power of the reset noise must be calculated by performing time-domain noise analysis considering the MOS transistor subthreshold noise model [Tian01]. For a soft reset process, the reset noise is a white noise with an average power given by the equation:

$$v_{reset}^2 = \frac{1}{2} \cdot \frac{k \cdot T}{C_{FD}} \qquad (V^2) \qquad \textbf{Eq. 268}$$

$$q_{reset}^2 = C_{FD}^2 \cdot v_{reset}^2 = \frac{1}{2} \cdot k \cdot T \cdot C_{FD} \qquad (electrons^2) \qquad \textbf{Eq. 269}$$

Note that the power of temporal noise for soft reset is half of that for hard reset. However, the lower reset noise comes at the expense of image lag. Since steady state is not reached during the reset time, the final reset voltage of FD node depends on its initial value. This image lag phenomenon does not appear in the photodiode if the steady state is reached during the reset phase. The image lag provokes artifacts in consecutive images with significant changes in the signal level between frames. In addition, the typical responsivity curve associated to the pixel operating in soft-reset mode presents a significant non-linearity for low-levels of illumination, which requires calibration at system level in order to obtain images with the sufficient quality for further processing, for instance the white balance in color applications.

To achieve both low reset noise and low lag the reset transistor drain is connected at the beginning of reset phase to a reference voltage instead of directly to the supply voltage, such as the reset transistor operates in ohmic region (hard reset mode). This operation ensures that the reset transistor always starts above threshold, thus eliminating lag. After this stage, the drain of reset transistor is connected again to the supply voltage. Therefore, at the end of reset phase the reset transistor is operating in sub-threshold region and the reset noise is not increased, remaining equal to the soft reset operation.

3.3.4.3. SOURCE FOLLOWER ERROR CONTRIBUTIONS

The Source Follower of Optical Sensor Block (see Figure 57) generates two types of errors: a random error given by the temporal noise and a systematic error which varies pixel to pixel due to the technological process variations. The mean value of this latter error over all pixels produces a global deviation in the sensed image and the differences between pixels generate FPN.

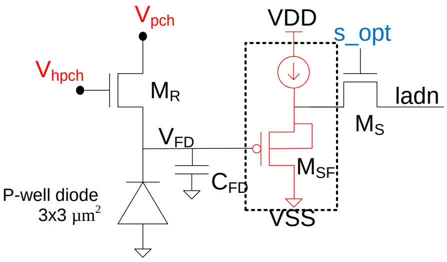


Figure 57. OPTICAL SENSOR BLOCK OF Q-EYE CELL.

STATIC SPATIAL NOISE. The photo-generated signal is affected by three systematic errors which can be attributed to the source follower:

- an offset component due to the shifting existing between the gate and source terminals of driver transistor,
- a gain error
- and a non-linearity error due to the body effect of this one.

The SF voltage gain is given by the equation:

$$G_{SF} \approx \frac{g_{m\_SF}}{g_{m\_SF} + g_{mb\_SF}} \qquad \textbf{Eq. 270}$$

Where g_{m_SF} is the transconductance parameter of the driver transistor M_{SF} and g_{mb_SF} is the parameter which models the body effect on the small-signal M_{SF} drain current. Ideally, the voltage gain is one but due to the body effect it is somewhat less than unity. The mismatch of this gain error between pixels generates a spatial gain error on the image (PRNU) similar to the FPN introduce by the mismatch of conversion gain associated with the photodiode.

The effect of the SF gain error is reduced eliminating the body effect in the PMOS driver transistor; i.e. by connecting the bulk to the source. Besides, the non-linearity of Input/Output SF characteristic is improved when the body effect is eliminated, but the effect of the finite output impedance of transistor which also changes substantially with the drain-source voltage remains. Thus, a certain percentage of non-linearity is present in the input-output SF characteristic.

The effect of the offset introduced by the SF and its mismatch is eliminated when the signal is obtained applying the Correlated Double Sampling (CDS) technique to the reset voltage and the voltage resulting from the integration of the photo-generated charge.

SETTLING ERRORS. In addition to the static errors, the settling errors due to an incomplete temporal response of circuit must be considered. The temporal response of SF is determined by the transfer function in the Laplace domain. The circuit behaves as a single pole system described by:

$$H(s) = \frac{G_{SF}}{\frac{C_o}{g_{m\_SF}} \cdot s + 1} = \frac{G_{SF}}{\tau_{SF} \cdot s + 1} \qquad \textbf{Eq. 271}$$

C_o is the total load capacitance at the S Follower transistor output node, including the capacitance of the LAM capacitor where the data sensor is stored. The time response to a step input of size *E* is:

$$v_{out}(s) = v_{ladn}(s) = \frac{E}{s} \cdot \frac{G_{SF}}{\tau_{SF} \cdot s + 1} \qquad \textbf{Eq. 272}$$

In the temporal domain:

$$v_{out}(t) = E \cdot G_{SF} \cdot \left(1 - e^{-\frac{t}{\tau_{SF}}}\right) \cdot u_o(t) \qquad \textbf{Eq. 273}$$

Then, the settling error is given by the expression:

$$v_{\varepsilon}(t) = -E \cdot G_{SF} \cdot e^{-\frac{t}{\tau_{SF}}} \qquad \textbf{Eq. 274}$$

When the input-output SF characteristic is obtained for a determined readout time (sweeping the input voltage), this settling error provokes an offset and gain error over the characteristic in first approximation. From an image quality point of view, a systematic error in the characteristic (offset and gain error) common to all pixels can be compensated. Nevertheless, due to the differences of behavior among devices, the settling errors are different among pixels. Thus, this mismatch in the offset and gain error generate a FPN, DSNU and PRNU respectively.

TEMPORAL NOISE. In relation to the temporal noise introduced by the Source Follower, both transistors, the driver and the current source, contribute with thermal and flicker noises.

Concerning thermal noise of MOS transistor, the drain-current thermal noise PSD in saturation is given by:

$$S_{i\_white} = \frac{8}{3} \cdot KT \cdot g_m \qquad \textbf{Eq. 275}$$

All symbols have their traditional meaning. Note that the dependencies with the transistor geometries and bias point, as well as technological or process dependencies, are embedded in the small signal transconductance g_m.

Concerning flicker noise of MOS transistor, the drain-current flicker noise PSD is:

$$S_{i\_1/f} = k_F \cdot \frac{g_m^2}{W \cdot L} \cdot \frac{1}{f} \qquad \textbf{Eq. 276}$$

k_F is a strong function of the specific production process, and generally has some bias current dependency. In general, k_F decreases with the bias current and increases for small geometries (close to minimal dimensions). The flicker noise is a decreasing function of transistor geometries.

Both noise sources of each transistor and between transistors are uncorrelated. Therefore, the noise Power Spectral Densities can be added at the output. The total output-current noise of Source Follower is expressed by the equation:

$$S_{n\_i\_SF}(f) = S_{i\_SF\_white} \cdot \left(1 + \frac{f_k}{f}\right) \qquad \textbf{Eq. 277}$$

Where f_k is the corner-frequency at which the two contributions (flicker and thermal) become equal and S_{SF_white} is the thermal component PSD given by the equation:

$$S_{i\_SF\_white} = \frac{8}{3} \cdot k \cdot T \cdot \left(g_{m\_SF} + g_{m\_CS}\right) \qquad \textbf{Eq. 278}$$

g_{m_SF} is the transconductance of driver transistor and g_{m_CS} is the transconductance of current source transistor.

The above output-current noise PSD can be expressed in voltage considering the impedance at the Source Follower output node which is computed by:

$$z_o \approx \frac{1}{\left(g_{m\_SF} + g_{mb\_SF}\right) + s \cdot C_o} \qquad \textbf{Eq. 279}$$

Then, the PSD of noise expressed in voltage is determined by the equation:

$$S_{n\_SF} = S_{n\_i\_SF} \cdot \left|z_o\right|^2 = \frac{8}{3} \cdot k \cdot T \cdot \frac{\left(g_{m\_SF} + g_{m\_CS}\right)}{\left(g_{m\_SF} + g_{mb\_SF}\right)^2} \cdot \frac{1 + \frac{f_k}{f}}{1 + \left(\frac{f}{f_c}\right)^2} \qquad \textbf{Eq. 280}$$

Where f_c is the Source Follower bandwidth:

$$f_c = \frac{1}{2 \cdot \pi} \cdot \frac{\left(g_{m\_SF} + g_{mb\_SF}\right)}{C_o} \qquad \textbf{Eq. 281}$$

The total PSD of noise generated by the source follower and referred at the input of the double sampling circuit can be described as the sum of their thermal and flicker components:

$$S_{n\_SF}\left(f\right) = S_{SF\_white}\left(f\right) + S_{SF\_1/f}\left(f\right) \qquad \textbf{Eq. 282}$$

$$S_{SF\_white}\left(f\right) = \frac{8}{3} \cdot k \cdot T \cdot \frac{\left(g_{m\_SF} + g_{m\_CS}\right)}{\left(g_{m\_SF} + g_{mb\_SF}\right)^2} \cdot \frac{1}{1 + \left(\frac{f}{f_c}\right)^2} \qquad \textbf{Eq. 283}$$

$$S_{SF\_1/f}\left(f\right) = \frac{8}{3} \cdot k \cdot T \cdot \frac{\left(g_{m\_SF} + g_{m\_CS}\right)}{\left(g_{m\_SF} + g_{mb\_SF}\right)^2} \cdot \frac{\frac{f_k}{f}}{1 + \left(\frac{f}{f_c}\right)^2} \qquad \textbf{Eq. 284}$$

The image is obtained from applying the Correlated Double Sampling (CDS) operation between the reset voltage and the voltage resulting from the integration of the photo-generated charge in the Floating Diffusion capacitance. This means that the noise associated to the source follower will be double-sampled (sampled and subtracted). In order to simplify the development and present concepts, an ideal double sampling operation will be assumed. As described in previous Sections (3.3.1.4 and 3.3.2.4), the CDS technique can be described as an Autozeroing (AZ) operation followed by a S&H process. The resulting PSD $S_{n\_CDS}\left(f\right)$ of the CDS operation can be decomposed into two components:

$$S_{n\_CDS}\left(f\right) = v_{n\_CDS}^2 = \left|H_0\left(f\right)\right|^2 \cdot S_{n\_SF}\left(f\right) + S_{fold}\left(f\right) \qquad \textbf{Eq. 285}$$

The first summand represents the base-band component and the second one includes the fold-over components. The noise-shaping transfer function $H_0\left(f\right)$ is given by the equation Eq.150. The noise-shaping transfer function of base-band contribution modules the white and flicker components of $S_{SF}\left(f\right)$ imposing a zero at the origin of frequency that cancels any offset and strongly reduces the 1/f noise. The white and flicker contributions of base-band term are determined by the equations:

$$S_{baseband\_white} = \left|H_0\left(f\right)\right|^2 \cdot S_{SF\_white} \qquad \textbf{Eq. 286}$$

$$S_{baseband\_1/f} = \left|H_0\left(f\right)\right|^2 \cdot S_{SF\_1/f} = \left|H_0\left(f\right)\right|^2 \cdot S_{SF\_white} \cdot \frac{f_k}{f} \qquad \textbf{Eq. 287}$$

The fold-over component $S_{fold}\left(f\right)$ contains the folded-contributions of thermal and flicker noises:

$$S_{fold}\left(f\right) = S_{fold\_white}\left(f\right) + S_{fold\_1/f}\left(f\right) \qquad \textbf{Eq. 288}$$

$$S_{fold\_white} = S_{SF\_white} \cdot \left(\pi \cdot f_c \cdot T_s - 1\right) \cdot \sin c^2\left(\pi \cdot f \cdot T_s\right) \qquad \textbf{Eq. 289}$$

$$S_{fold\_flic\ker} = S_{SF\_white} \cdot 2 \cdot f_k \cdot T_s \cdot \left[1 + \ln\left(\frac{2}{3} \cdot f_c \cdot T_s\right)\right] \cdot \sin c^2\left(\pi \cdot f \cdot T_s\right) \qquad \textbf{Eq. 290}$$

Auto-zeroing or double sampling process must certainly take place with clock frequencies below the source-follower bandwidth and double sampling circuit bandwidth, since the whole circuit (source follower driving the double sampling circuit) must be able to settle within allowed time interval. Moreover, the auto-zeroing operation eliminates the noise power at very low frequencies. Therefore, it is expected that fold-over components dominates over base-band components.

By equating equations Eq. 289 and Eq. 290 a curve in terms of $f_k \cdot T_s$ versus $f_c \cdot T_s$ is obtained. This curve divides the $\left(f_k \cdot T_s, f_c \cdot T_s \right)$ plane in two regions, one of them corresponding to dominant flicker fold-over noise, and the other to dominant white fold-over noise. The boundary curve is defined by the expression:

$$f_k \cdot T_s = \frac{\pi \cdot f_c \cdot T_s - 1}{2 \cdot \left[1 + \ln\left(\frac{2}{3} \cdot f_c \cdot T_s \right) \right]} \qquad \textbf{Eq. 291}$$

Figure 58 shows a plot of expression Eq. 291.

Depending on the relation between corner-frequency f_k and Source Follower bandwidth f_c , one of two contributions (white component or flicker component) dominates.

The temporal noise associated to the source follower together with the temporal noise of photodiode degrades the Dynamic Range (DR) and Signal to Noise Ratio (SNR) parameters of images.

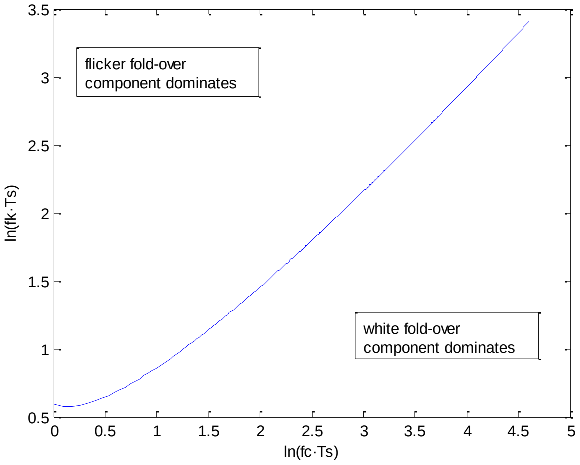


Figure 58. **REGIONS OF FLICKER AND WHITE FOLD-OVER COMPONENTS.**

3.4. EXEMPLARY RESULTS OF DEGRADATION CAUSED BY ERRORS

Roughly speaking errors in the operation of CMOS image sensors, labeled generically as noise despite its origin, can be classified in two categories, namely:

- **Random Temporal Noise** and
- **Fixed Pattern Noise** (FPN) [Gama98].

The former refers to stochastic variations of the output signal in time domain. The latter refers to spatial variations across a uniform 2-D image plane. These latter variations are deterministic from frame to frame, although change randomly from device instance to device instance.

According to their origin, errors analyzed in previous sections can be classified into four categories (see Table 10), namely:

- Offset
- Gain error
- Non-linearity
- Circuit noise

For given signal values, the first three errors are *time-invariant* and thus, amenable for cancellation at system level. Noise, on the contrary, produces time-variant errors which cannot be corrected and which ultimately sets the SNR and the DR of the system.

Source Noise		Random Temporal Noise	Fixed Pattern Noise		
Cell block		Electronic noise	Offset	Gain	Non-linearity
Optical sensor	**Photodiode**	Shot noise associated to photo-generated and dark currents	Dark current	Mismatch of capacitance associated to charge – voltage conversion (C_{FD})	Non-linear behavior of capacitance C_{FD}
	Reset transistor	Thermal noise			
	Source follower	Thermal and flicker noises	Shifting due to threshold voltage of driver	Gain error due to body effect of driver transistor	a) Body effect of driver b) Finite output impedance of transistors
LAM	**Switches**	Thermal noise	Charge injection and feedthrough of switch SW2		
	OpAmp	Thermal and flicker noises	a) Offset and finite gain b) Temporal settling error	a) Offset and finite gain b) Temporal settling error	
	Capacitor		Leakage due to transistors diffusions		
MAC	**Switches**	Thermal noise	Charge injection and feedthrough		Charge injection and feed-through
	OpAmp	Thermal and flicker noises	a) Offset and finite gain b) Temporal settling error	a) Offset and finite gain b) Temporal settling error	
	Capacitors			Mismatch between capacitors C1 and C2	Non-linear behavior of capacitors C1 and C2

Table 10. Errors associated to the non-ideal behavior of circuits.

3.4.1. TIME-INVARIANT ERRORS

Let us first consider time-invariant errors. Each of these errors has two different components:

- a *global* one equal for all cells in the array and,
- a *local* one due to random mismatches among the cells, and which manifests as FPN.

UNIFORM IMAGE. Let us assume that a uniform image gets stored in the analogue memory. The ideal storage operation is described by:

$$im_{out} = im_{in}$$ **Eq. 292**

Where im_{out} and im_{in} represent, in matrix form, the input image during the LAM writing phase and the output image during the LAM reading phase, respectively. Pixel-wise, this ideal operation gets described by

$$im_{out}\left(i,j\right) = im_{in}\left(i,j\right)$$ **Eq. 293**

where (i,j) represent the pixel coordinates. In the presence of gain and offset errors (introduced during both phases) the ideal equation above is transformed into:

$$im_{out}(i,j) = offset(i,j) + (1+\varepsilon(i,j))\cdot im_{in}(i,j) \qquad \textbf{Eq. 294}$$

where each of these errors have a global component and a local one, namely:

$$offset(i,j) = O_{global} + O_{ij} \qquad \textbf{Eq. 295}$$

$$\varepsilon(i,j) = \varepsilon_{global} + \varepsilon_{ij} \qquad \textbf{Eq. 296}$$

OFFSET IMPACT ON ZERO IMAGE. Let us now consider that the input image is zero. In this case there is no gain error and the erroneous output is given by:

$$im_{in} = 0 \Rightarrow im_{out} = offset(i,j) \qquad \textbf{Eq. 297}$$

Figure 59 illustrates the impact of this error, both at perceptual level and through histograms. Images have been scaled for clarity.

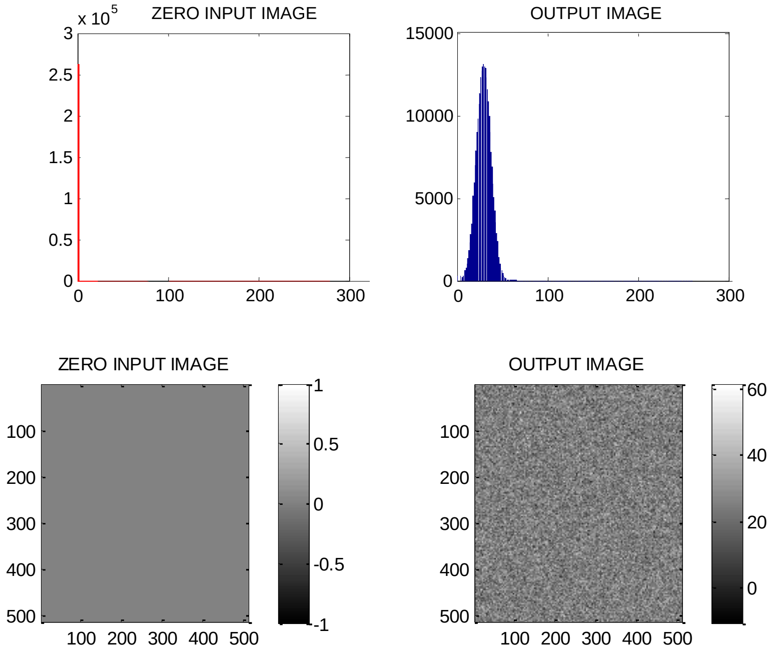


Figure 59. **HISTOGRAM OF IMAGES.**

The input in this figure is a constant zero image whose histogram is concentrated in code zero. The output shows a random offset term (due to mismatching) plus a global term. The former produces a spatial noise which contributes to dark signal non-uniformity, DSNU. Since the non-idealities have been represented by normal distributions, the error follows also a normal distribution whose standard deviation measured for this case is 5 DN (Digital Number of 8 bits).

Regarding the global error term, it is obtained through image averaging. The average operation filters the high frequency components of spatial noise obtaining the global term. In this particular example, the global offset is 20 DN.

OFFSET IMPACT ON ACTUAL IMAGE. Figure 60 illustrates the impact of offset error effect on a real image. Both the global offset terms and the random offset term manifest in the output image.

GAIN-ERROR IMPACT ON UNIFORM NON-ZERO IMAGES. For uniform non-zero images, the spatial noise due to the gain error is added to the offset contribution. While this latter contribution is independent of signal, the magnitude of the gain error increases with the signal level. This phenomenon is shown by the histograms of uniform images corresponding to different signal levels stored at the analogue memories and then read from them. In this example, the offset error has been considered null in order to study only the influence of gain error.

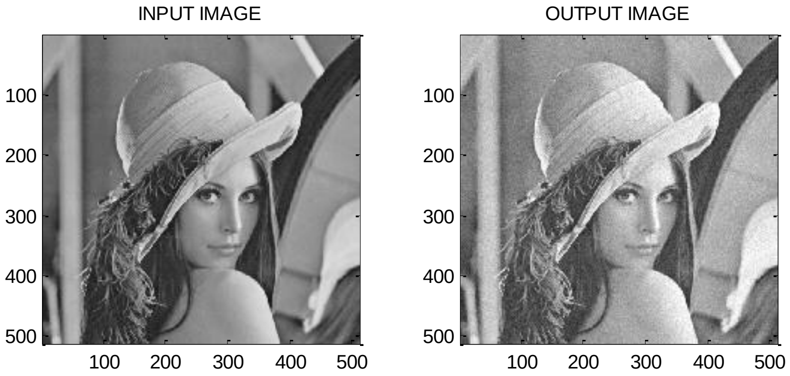


Figure 60. OFFSET ERROR ON REAL SCENE.

The two first histograms in Figure 61 correspond to a constant, zero input image.

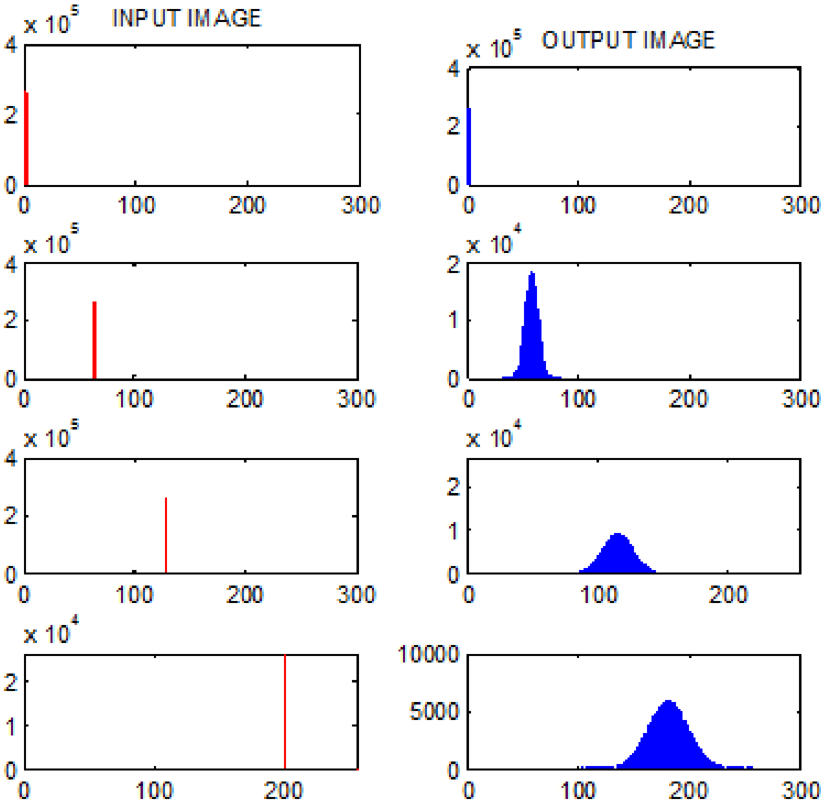


Figure 61. HISTOGRAMS FOR SEVERAL SIGNAL LEVELS.

The input and output histograms coincide since the offset error is assumed null. The other histograms correspond to non-zero, constant images and show the superposition of a global contribution which changes the mean level of image and a local random error contribution due to the mismatch between devices which generates spatial noise. The standard deviation associated to the spatial noise of output images increases with the signal level, see Figure 61.

GAIN-ERROR IMPACT ON UNIFORM NON-ZERO IMAGES. Figure 62 shows the gain error effect on a real image, considering negligible the offset error. The global contribution implies a loss of image contrast.

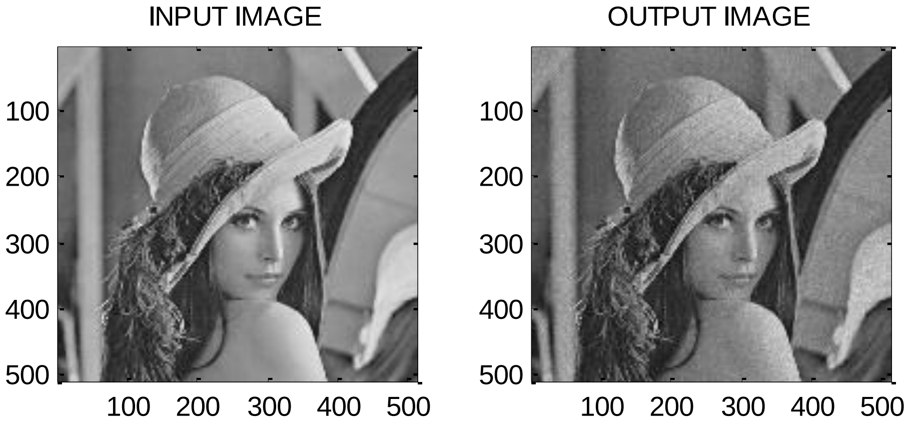


Figure 62. ILLUSTRATING THE VISUAL IMPACT OF GAIN ERROR ON ACTUAL SCENES.

3.4.2. TIME-VARIANT ERRORS

Let us consider now time-variant errors. According to previous studies, there are three main sources of temporal noise:

- Shot noise
- Thermal noise
- Flicker noise

Shot noise is mostly contributed by the photodiode. Its average power is proportional to the number of photo-generated electrons (Eq. 261):

$$q_{shot}^2 = \frac{\left(i_{ph} + i_{dc}\right)\cdot T_{\text{int}}}{q} \qquad \left(electrons^2\right)$$

meaning that the power of shot noise increase with the signal level. **Thermal** and **flicker** noises are contributed by the analogue circuits within the cell, including the circuitry embedded within the optical sensor. Unlike the shot noise, the thermal and flicker noises are independent from the signal level.

Noise sources are added as the signal transverses the system from the optical sensor to the readout section across the LAMs, the MAC, the resistive grid, etc. Let us consider for illustration purposes just the optical sensor. Figure 63 shows a model of the temporal noise.

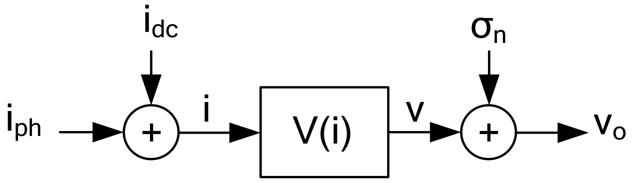


Figure 63. TEMPORAL NOISE ADDITIONS AT CONCEPTUAL LEVEL.

In this figure, V(i) is the *sensor transfer function*:

$$V(i) = \begin{cases} CG\cdot\dfrac{i\cdot T_{\text{int}}}{q} & 0 < i < \dfrac{q\cdot Q_{sat}}{T_{\text{int}}} \\ CG\cdot Q_{sat} & i \geq \dfrac{q\cdot Q_{sat}}{T_{\text{int}}} \end{cases} \qquad \textbf{Eq. 298}$$

Where Q_{sat} is the full well capacity.

The total noise (σ_n^2) is obtained by summing all power functions associated with each noise contribution:

$$\sigma_n^2 = v_{shot}^2 + v_{reset}^2 + v_{readout}^2 + v_{process\sin g}^2 \qquad \left(V^2\right)$$ **Eq. 299**

$$v_{shot}^2 = CG^2 \cdot q_{shot}^2$$ **Eq. 300**

The temporal noise contribution corresponding to the processing of sensed image ($v_{proce\sin g}^2$) has been considered and added to the temporal noise introduced by the sensing process ($v_{shot}^2 + v_{reset}^2 + v_{readout}^2$).

The noise contributions which are independent from the signal level introduce a random noise in the image with a constant standard deviation for any signal level. Nevertheless, the shot noise generates a random noise whose standard deviation is proportional to the root square of signal level. This fact can be observed in the histograms of Figure 64. The standard deviation of temporal noise increases with the signal level due to the shot noise contribution v_{shot}^2 , growing the dispersion. For zero input image the shot noise is null. Therefore, in dark conditions, the resulting temporal noise is associated to the reset noise v_{shot}^2 and the readout noise v_{shot}^2, considering negligible the contribution of shot noise generated by the dark current of photodiode.

The Dynamic Range (DR) quantifies the ability of sensor to adequately image both high lights and dark shadows in a scene. It is defined as the ratio of the largest non-saturating output signal to the smallest detectable output signal. The last one is determined by the standard deviation of image under dark conditions.

DR is typically measured in dBs:

$$DR = 20 \cdot \log_{10} \frac{V_{\max}}{V_{\min}} = 20 \cdot \log_{10} \frac{Q_{sat} \cdot CG}{\sqrt{CG \cdot \frac{i_{dc} \cdot T_{\text{int}}}{q} + v_{reset}^2 + v_{readout}^2}}$$ **Eq. 301**

Usually, the reset and readout noises determine the maximum DR. The application of processing to a sensed image in the analogue domain means increasing the floor of noise and, thus, decreasing the DR.

The Signal to Noise Ratio (SNR) quantifies the quality of image. It is the ratio of output signal power to the average noise power. SNR is expressed in dBs:

$$SNR = 10 \cdot \log_{10} \frac{\left(CG \cdot \frac{i_{ph} \cdot T_{\text{int}}}{q}\right)^2}{\sigma_n^2}$$ **Eq. 302**

SNR increase with the input signal i_{ph} , first at 20dB per decade since read noise dominates for small signal, then at 10dB per decade shot noise due to photo-detector dominates, see Figure 65.

The SNR can be expressed in function of photo-generated electron number:

$$SNR = 10 \cdot \log_{10} \frac{n_e^2}{q_n^2}$$ **Eq. 303**

Being q_n^2 the input referred noise power expressed in electrons.

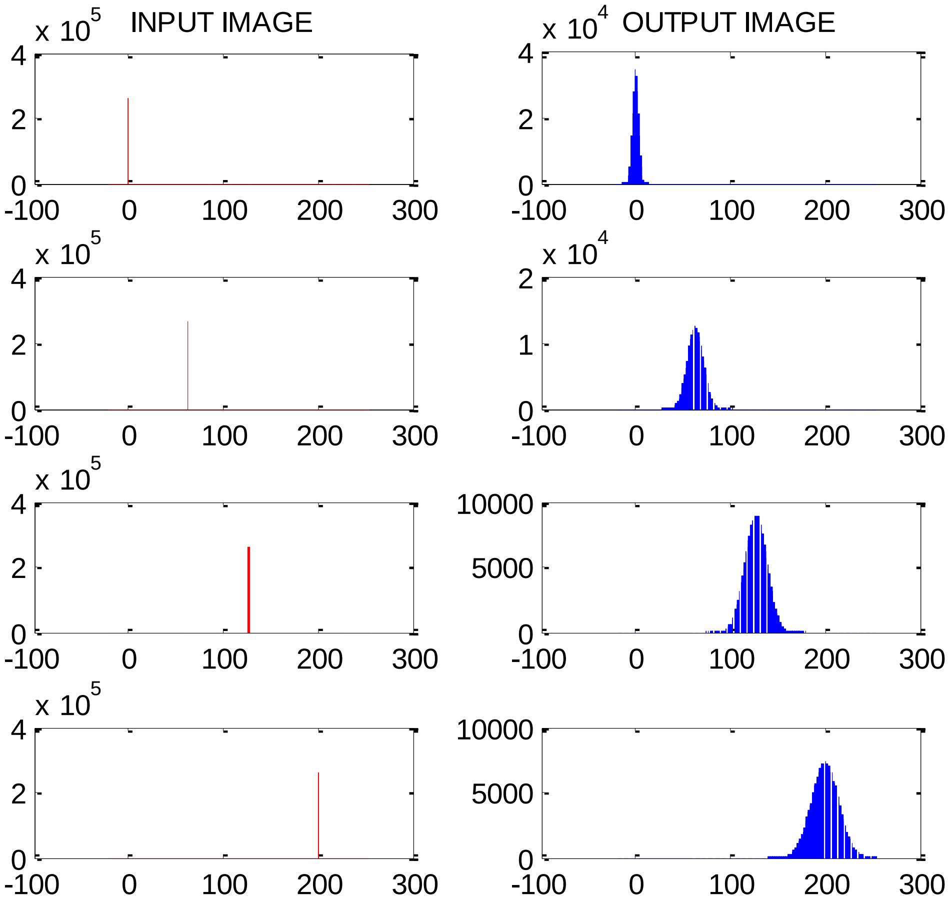


Figure 64. TEMPORAL NOISE EFFECT ON UNIFORM IMAGES.

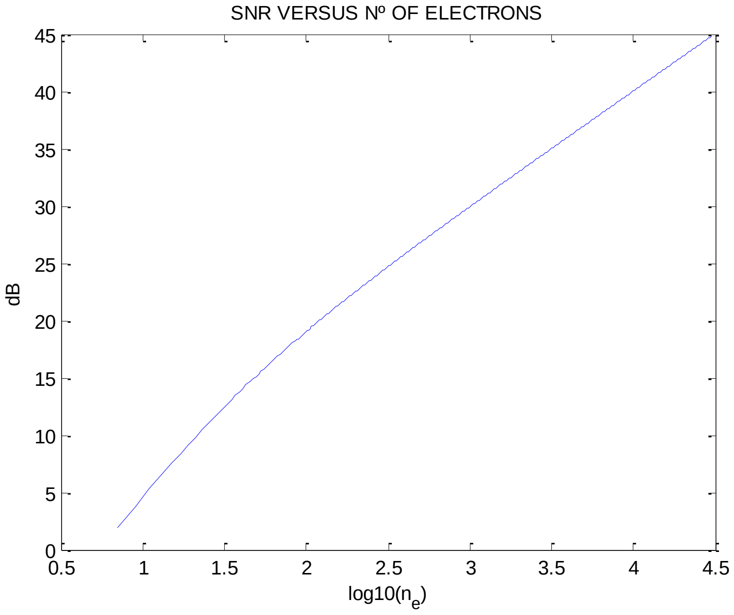


Figure 65. SNR VERSUS NUMBER OF PHOTO-GENERATED ELECTRONS.

Figure 66 represents one of several consecutive captures of same uniform image and their corresponding histogram. The standard deviation of temporal noise is 5 DN of 8 bits.

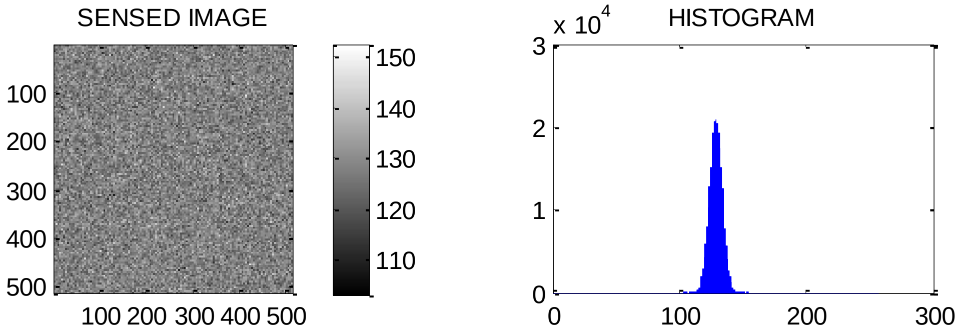


Figure 66. TEMPORAL NOISE EFFECT ON AN UNIFORM IMAGE.

To reduce temporal noise impact a consecutive captures can be averaged:

$$im_{avg} = \frac{\sum_{n=1}^{N} im_{in}(i,j,n)}{N}$$ **Eq. 304**

where the index *n* indicates the temporal sample. In this case, the high frequency components of temporal noise are attenuated, which manifests as a decrease in the dispersion or standard deviation of histogram – see Figure 67.

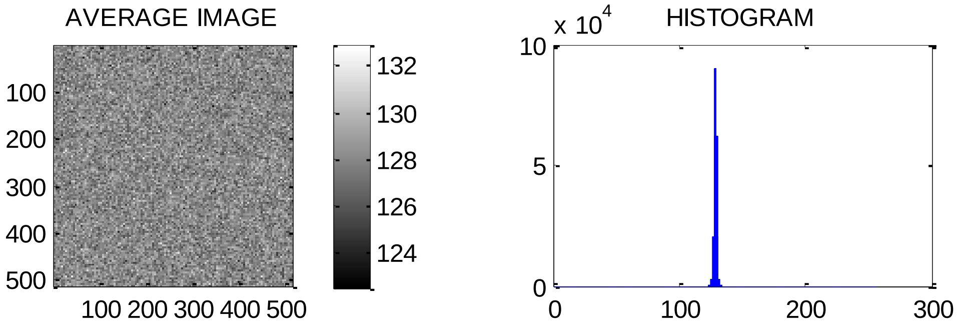


Figure 67. IMAGE RESULTING OF TEMPORAL LOW PASS FILTER.

When a temporal low pass filter is applied to a sequence of images, the temporal noise is significantly reduced. Under these conditions, the FPN of image sensor is perfectly noticeable, [Snoe06]. This is the reason why the human vision perceives the FPN in a video sequence.

3.4.3. PRACTICAL RECOMMENDATIONS FOR PROCESSING IN THE PRESENCE OF ERRORS

The first stage in a Smart Image Sensor is the capture of a real scene. In the case of *Q-Eye* system, the optical sensor generates two images: the real scene and the reset image. The input signal image is obtained applying the Correlated Double Sampling through the MAC block and stored in a Local Analogue Memory (LAM). This sensing process introduces errors in the input signal image due to the non-ideal behavior of Optical Sensor block and processing circuits MAC and LAM.

The resulting image from sensing operation is affected by temporal noise and FPN. And it is the input to the pre-processing operations.

A typical operation flow of the *Q-Eye* is depicted in Figure 68.

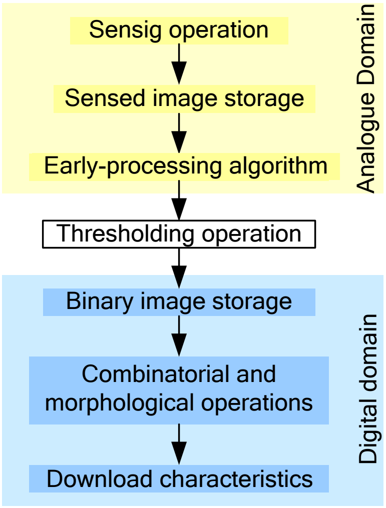


Figure 68. CVIS OPERATION FLOW.

Each analogue circuit operation involves the introduction of temporal noise and some type of systematic errors (offset, gain error or non-linearity), generating spatial noise. Therefore, the basic analogue processing operations (writing LAMs, reading LAMs, MAC operation and diffusions) add temporal and spatial noises to the image obtained in the sensing phase. The power of noise associated to each basic operation which composes the early-processing algorithm is summed to the noise power of image resulting from the previous operation:

$$\sigma_n^2 = \sigma_{sensing}^2 + \sigma_{processing}^2 \quad \textbf{Eq. 305}$$

$$\sigma_{sensing}^2 = v_{shot}^2 + v_{reset}^2 + v_{readout}^2 + v_{FPN}^2 \quad \textbf{Eq. 306}$$

$$\sigma_{processing}^2 = \sum_{i=1}^{N} v_{operation}^2(i) \quad \textbf{Eq. 307}$$

$v_{processing}^2(i)$ is the PSD of total noise introduced by an analogue basic operation. This total noise includes temporal noise and FPN:

$$v_{processing}^2(i) = \sigma_{temporal}^2(i) + \sigma_{FPN}^2(i) \quad \textbf{Eq. 308}$$

In order to minimize the noise introduced by the analogue processing algorithm, the number of basic operations must be reduced as much as possible. For instance, if the early-processing algorithm requires the implementation of a low pass filter, this one can be developed through a convolution, which involves several MAC operations, or through a diffusion process. The latter means a lower number of basic operations. And thus, the image resulting from diffusion process presents a lower noise level.

The noise level in the image resulting from the early-processing must be such that the thresholding operation can be carried out correctly. Once the information is in the digital domain, the binary processing is so much robust in relation to the errors and the number of basic operations (combinational operations, hit and miss operations and shifting) is not limited.

CHAPTER 4 - THE Q-EYE CHIP

The Q-Eye is a CMOS chip conceived for image sensing and processing. It hence belongs to the class of the so-called CMOS Image Sensor (CIS) chip [Font11]. However, while CIS chips are typically fabricated by using dedicated CMOS processes, optimized for high quality photon detection, the Q-Eye is realized in a standard CMOS technology. Another difference between the Q-Eye and a typical CIS is that the Q-Eye is designed to both acquire images (sensing function), and process them while being acquired (processing function). It is hence a kind of Smart-CIS. The processing embedded within the Q-Eye is conceived to perform most significant function belonging to early vision; i.e., to reduce the amount of information contained within the images prior to high-level digital processing. Actually, owing to the embedding of stored-programming features, the Q-Eye is able to prompting action commands in response to the analysis of images and the extraction of the information contained in their variation across space and time. These command actions, decisions in brief, may be considered the primary outputs of the family of image handling chips to which the Q-Eye belongs. Although these chips have many similarities with CISs, the fact that their primary outputs are decisions instead of images compels us to call them CVIS (CMOS VIsion Sensors).

4.1. THE Q-EYE CHIP BLOCK DIAGRAM AND ARCHITECTURE

The Q-Eye is a quarter CIF resolution *vision sensor* which integrates all the necessary elements to acquire, carry out pre-processing algorithms at pixel level, digitize the resulting images and transmit them to an external host via a digital input-output interface. Figure 1 shows the conceptual architecture of Q-Eye. The core component is the cell array, composed of interconnected cells whose operations and associated circuits have been described in Chapter 3. The cell array is controlled by an ensemble of programming circuits and connected to the outside by the input-output block.

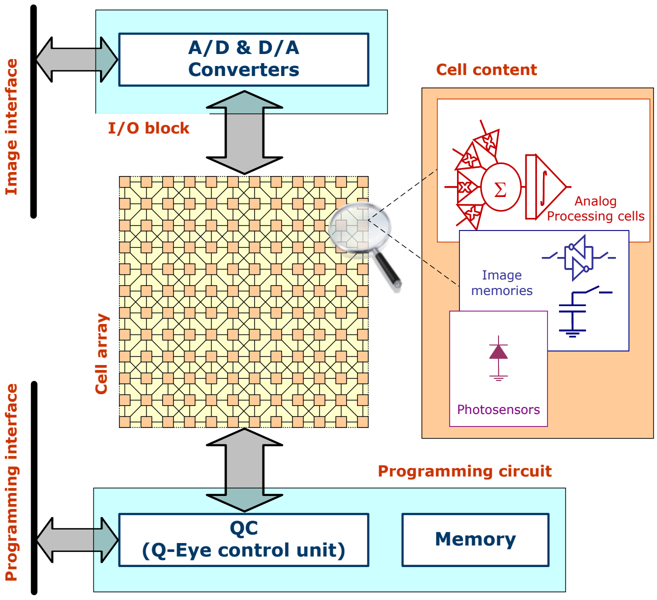


Figure 1. CONCEPTUAL Q-EYE ARCHITECTURE.

The architecture of Q-Eye can be divided into six sections which are labeled in Figure 2, namely:

- the Sensing-Processing Array,
- the I/O Image Interface,
- the Programming Interface,
- the Configuration Registers Block,

- the Control Unit and
- the Miscellaneous On-chip Blocks.

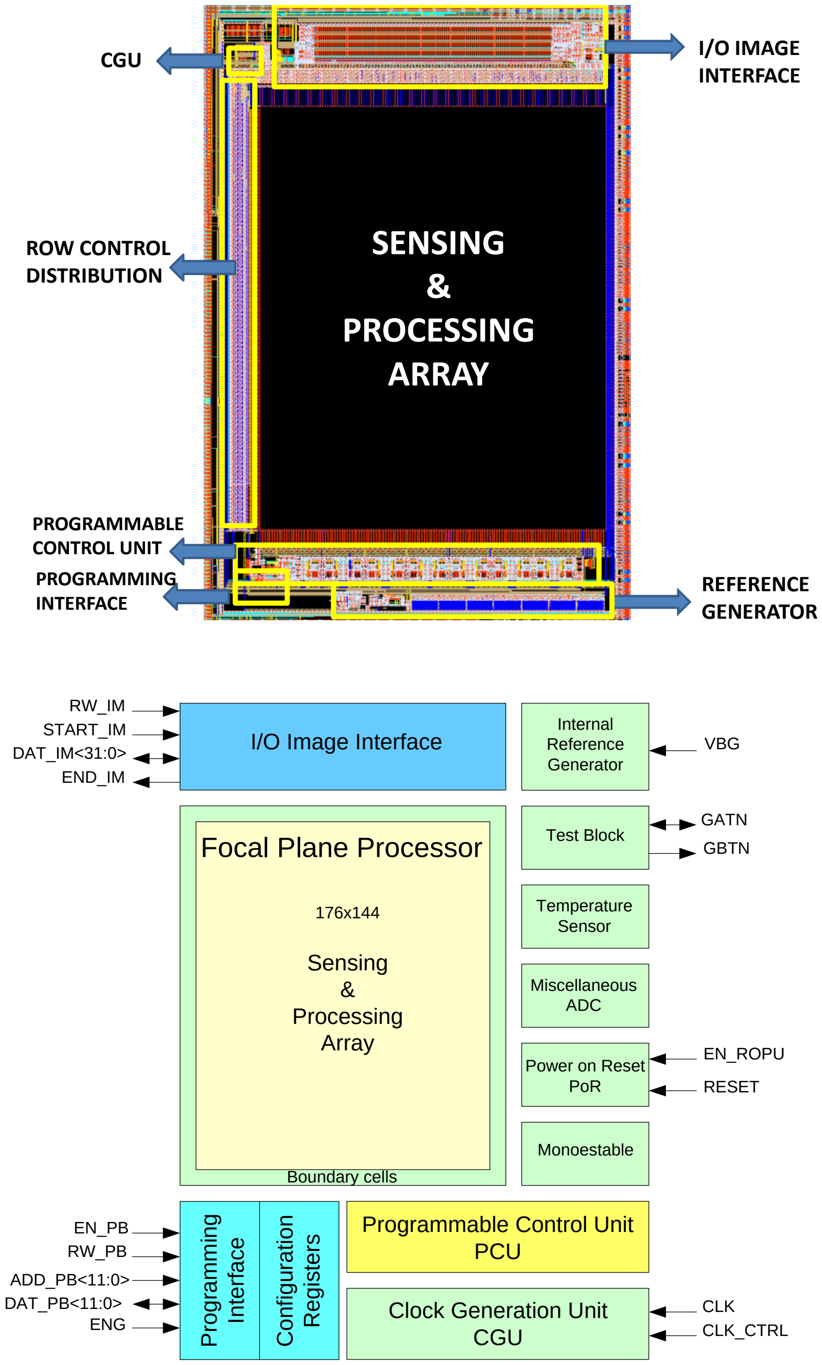


Figure 2. Q-EYE BLOCK TOP VIEW LAYOUT AND BLOCK DIAGRAM.

The **SENSING-PROCESSING ARRAY** is made up of a network of 176x144 interconnected sensing-processing cells plus boundary cells to set the contour conditions. The functions embedded at each array cell and the circuits used for their implementation have been both described in Chapter 3.

The **I/O IMAGE INTERFACE** consists of two main blocks:

- the Binary I/O Block and
- the Gray Level I/O Block.

Both have been optimized seeking for large frame rates. The former handles downloading of binary images and the latter is responsible for downloading images in gray levels which is achieved owing to the concourse of a bank of per column S&Hs and four Flash AD converters.

The main function of the **PROGRAMMING INTERFACE** is configuring and programming the Q-Eye system. The user configures the operation of the system and loads the sequence of instructions for the processing array into the memory section of the Control Unit block.

The **CONFIGURATION REGISTERS BLOCK** is composed by Input/Output port registers which configure:

- the operation of system,
- the Image interface and
- some Miscellaneous Blocks (Power-on-Reset, Clock Generation Unit, Internal Reference Generator, etc.), intended for the generation of internal analog references required to the sensor operation, the system clock implementation and other special functions such as temperature sensing, general purpose AD conversion, etc.……

For example, the configuration registers define the several internal analogue references used by the mixed-signal blocks within the Q-Eye. These configuration registers are written and read by the Programming Interface.

The **CONTROL UNIT** is responsible for the overall control of the Q-Eye system. It generates all the necessary signals to control the processing array, the Image Interface and the Miscellaneous On-Chip Blocks.

The **MISCELLANEOUS ON-CHIP BLOCKS** are intended to reduce the number of external components required for the operation of the system. They include the:

- **Power-on-Reset** (PoR), which generates an initial internal reset pulse,
- the **Clock Generation Unit** (CGU), Monostable Block for controlling of diffusion processes in the array,
- the **Temperature Sensor**,
- a **General Purpose AD Converter** and
- the **References Generator** with a high-accuracy band-gap.

There is a **TEST BLOCK** which consists of two multiplexers. The digital multiplexer connects interesting digital control signals to the **Global Binary Test Node** (GBTN) for debugging. With the same objective of debugging, the analogue multiplexer connects all internal analogue references to the **Global Analogue Test Node** (GATN).

All the miscellaneous blocks and the I/O Image Interface can be configured by the user through the Programming Interface. Moreover, the user controls the PCU block and loads into the PCU memory bank the sequence of instructions using the programming Interface. A FPP instruction is a binary vector where each bit defines one control signal belonging to the sensing-processing array. The image load/download processes are controlled by the signals of Image Interface RW_IM, START_IM and END_IM.

The remaining three sub-sections within this section are to basically provide on how to interface the Q-Eye chip when embedded into camera systems.

4.1.1. EXTERNAL INTERFACES OF THE Q-EYE

Besides the two interfaces explicitly mentioned in previous section, the Q-Eye has a third, implicit one called **Generic interface.** Table 1 lists the external pins of Q-Eye system indicating their corresponding functionalities.

The digital signals of external interfaces, except CTRL_CK, ENG and EN_ROPU, are registered by the external interfaces of the Q-Eye to ensure the synchronization between this system and the external host. Two clocks are involved, namely:

- the external master clock (CLK) and
- the **internal system clock**, called *ck_control.*

PIN NAME	PIN TYPE	Nº OF PINS	PULL UP/DOWN	DESCRIPTION
DAT_PB<9:0>	I/O	10	-	Programming interface: data bus.
ADD_PB<11:0>	I	12	-	Programming interface: address bus.
RW_PB	I	1	PD	Programming interface: Read/Write signal. It configures the data bus for reading data the chip (RW_PB=1) or writing into the chip (RW_PB=0).
EN_PB	I	1	PD	Programming interface: Enable signal. It enables the reading or writing process of the blocks connected to the Programming interface.
DAT_IM<31:0>	I/O	32	-	I/O image interface: data bus.
RW_IM	I	1	PD	I/O image interface: Read/write signal. It configures the I/O bus for reading images from the chip (RW_PB=1) or writing images into the chip (RW_PB=0).
START_IM	I	1	PD	I/O image interface: trigger signal. The rising edge of this signal launches an I/O image process.
END_IM	O	1	-	I/O image interface: The rising edge indicates the I/O image process has finished.
RESET	I	1	PU	Generic interface: Global reset of chip. (Active high)
ENG	I	1	PD	Generic interface: Global enable of chip. (Active high). This state leaves the chip isolated from the outside. All pins to high impedance and input signals internally latched to their last value.
CLK	I	1	-	Generic interface: Master input clock. 50 MHz. 50% duty cycle.
CTRL_CLK	I	1	PU	Generic interface: Clock control signal. It enables the internal programmable delay of master clock. (Active high). For CTRL_CK=0 the delay block is bypassed.
EN_ROPU	I	1	PU	Generic interface: Enable the Reset on Power Up block. (Active high). It enables the circuit used to generate an internal and automatic initial reset after power up.
GBTN	O	1	-	Global binary test node.
GATN	A I/O	1	-	Global analogue test node.
VBG	A I/O	1	-	Band-gap voltage reference.
VDD33	Supply	8	-	3.3V power supply for the I/O pads.
VSS33	Supply	8	-	3.3V GND for the I/O pads. Shorted with VSSA, VAS and VSSPACK.
VDDA	Supply	20	-	1.8V power supply for all internal circuitry. Core power supply.
VSSA	Supply	20	-	1.8V GND for core. Connected to the substrate, and therefore shorted with VSS33, VAS and VSSPACK.
VAD	Supply	1	-	3.3V power supply of pad ring analogue section.
VAS	Supply	1	-	3.3V GND for pad ring analogue section. Shorted with VSS33, VSSA, VSSPACK.
VSSPACK	Supply	2	-	GND connection to package cavity. Shorted with VSS33,VSSA and VAS.

Table 1. External pins of Q-Eye Chip.

These clock signals are equal by default. However, a programmable delay between them can be defined by the user through the Clock Generation Unit block – see Figure 2.

The external digital signals commute with the rising edge of the external master clock CLK, while the synchronization registers store the signals with the falling edge of the internal system clock *ck_control*. Regarding the internal digital circuitry, it works with the rising edge of internal system clock and the synchronization registers capture the signals with the falling edge.

Figure 3 and Figure 4 describe the synchronization scheme and shows examples of timing diagram for input and output signals of external interfaces.

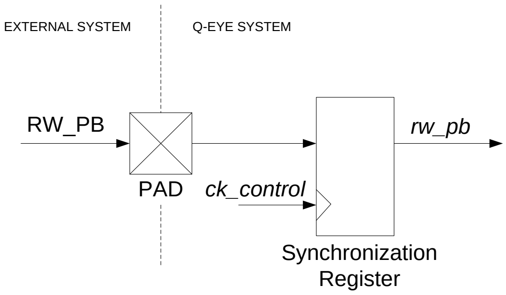


Figure 3. SYNCHRONIZATION OF SYNCHRONOUS DIGITAL SIGNALS.

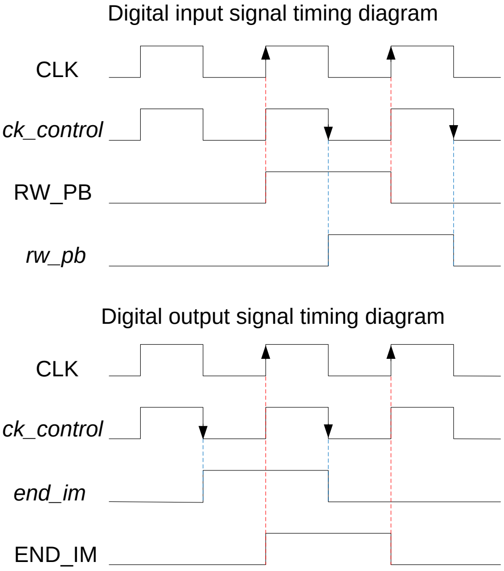

Digital input signal timing diagram
CLK
ck_control
RW_PB
rw_pb
Digital output signal timing diagram
CLK
ck_control
end_im
END_IM

Figure 4. TIMING DIAGRAM FOR DIGITAL SIGNALS OF EXTERNAL INTERFACES.

4.1.1.1. THE GENERIC INTERFACE

This interface is composed by a group of global signals which affect to the operation of complete system, namely:

- The **global reset signal** (RESET) configures automatically the complete Q-Eye system in a determined known default state. This signal is active in high level.
- The **global enable signal** (ENG) is active in high level and enables the external interfaces of Q-Eye system. When this signal is disabled (ENG=0), the external interfaces are disabled as well, the Q-Eye system is isolated from the outside and the input digital signals are latched to their last value.
- The **enable signal for initial internal reset** (EN_ROPU) is active in high level and enables the internal Reset on Power Up block (see Figure 2), which generates an initial and internal reset pulse for the complete system after the power up of this. EN_ROPU signal must be configured in high and does not commute anymore.
- The already mentioned **external master clock** CLK is the reference clock to generate the internal programming interface clock *ck_control*. The system operates at 50 MHz.
- The **configuration system clock signal** CTRL_CLK is active in high level and enables the programmable delay block which generates the internal programming interface clock *ck_control* from the external clock CLK, see section 4.1.3.2. When this signal is disabled (CTRL_CLK=0) the programmable delay block is bypassed and *ck_control*=CLK. If a zero delay is programmed in the clock generation block, the internal programming interface clock *ck_control* is equal to the external master clock CLK. By default, this signal can be set to 0.

4.1.1.2. THE PROGRAMMING INTERFACE

This interface is the responsible for the programming and control of the Q-Eye. The user has access to the configuration registers and instruction memories through this interface.

The interface is composed by:

- A **12-bits address bus** ADD_PB<11:0>. The four Most Significant Bits (MSBs) define the pagination of configuration registers and instruction memories. Table 2 describes this pagination.
- A **10-bits Input/Output data bus** DAT_PB<9:0>. This bidirectional bus is separated into 2 buses, input one and output one, within the Q-Eye chip. They are called dat_pb_i<9:0> and dat_pb_o<9:0> respectively – see Figure 5.
- The Read/Write control signal RW_PB.
- The enabling access control signal EN_PB.

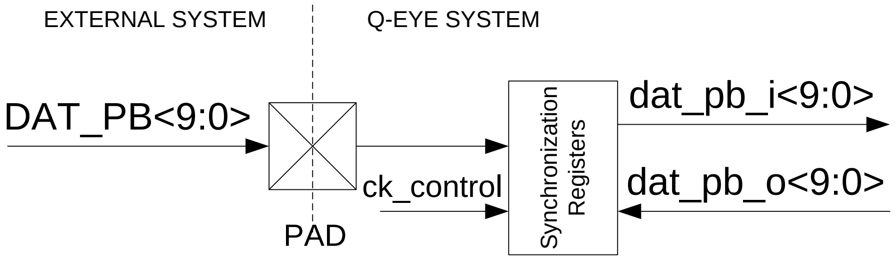


Figure 5. INPUT/OUTPUT DATA BUS OF PROGRAMMING INTERFACE.

ADD_PB<11:8>	Description
0000	Access to the instruction register “PBCTRL” of Programmable Control Unit (PCU)
0001	Access to the mask register “PBMASK” of PCU
0010	Address reserved for the access to the configuration registers
0011	
0100	
0101	
0110	No used
0111	No used
1000	Access to the page 0 of Program Memory “PBMEM” in PCU
1001	Access to the page 1 of Program Memory “PBMEM” in PCU
1010	Access to the page 2 of Program Memory “PBMEM” in PCU
1011	Access to the page 3 of Program Memory “PBMEM” in PCU
1100	Access to the page 4 of Program Memory “PBMEM” in PCU
1101	Access to the page 5 of Program Memory “PBMEM” in PCU
1110	Not used
1111	Not used

Table 2. Addresses for registers and memories in the Q-Eye.

Figure 6 depicts the timing diagram to write a determined configuration register or program memory address. The timing diagram to read a particular configuration register or program memory address is shown in Figure 7.

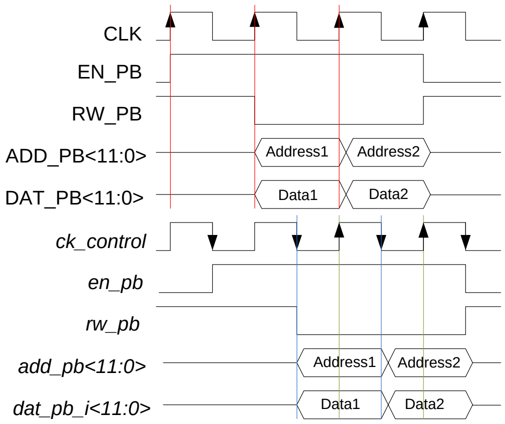


Figure 6. **WRITING PROCESS FOR THE PROGRAMMING INTERFACE.**

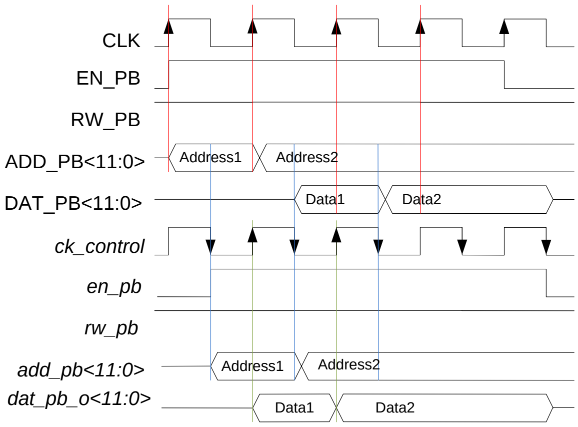


Figure 7. **READING PROCESS FOR THE PROGRAMMING INTERFACE.**

4.1.1.3. THE I/O IMAGE INTERFACE

Images can be loaded or downloaded in the Q-Eye through the I/O image interface. These images can be in gray level or binary. The I/O image interface is composed by:

- An **Input/Output 32 bits data bus** DAT_IM<31:0>. This bidirectional bus is separated into 2 buses, input one and output one, within the Q-Eye chip. They are called *dat_im_i<9:0>* and *dat_im_o<9:0>* respectively. See Figure 8
- The **Read/Write control signal** RW_IM. This signal defines the direction of data bus in the Read/Write process. RW_IM=1 configures the Q-Eye system to read images from it. RW_IM=0 configures the Image Interface to write images into the Q-Eye system.
- The **launching control signal** START_IM. This signal is active with the rising edge and starts the Read/Write process.
- The **ending control signal** END_IM. This signal is an output signal which is active by level and indicates the ending of Read/Write process. When the ending signal is in low level, it indicates that the Image Interface is busy. The high level means the Read/Write process has finished.

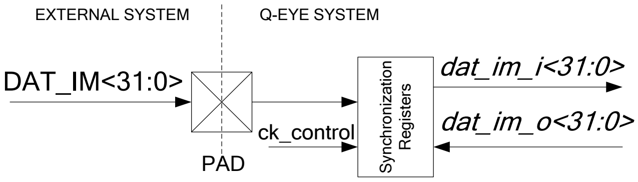


Figure 8. **INPUT/OUTPUT DATA BUS OF IMAGE INTERFACE.**

The I/O Image Interface is configured by the external host through the Programming Interface which access to the configuration registers included in the Image Interface.

In these configuration registers, the user can define the Read/Write mode. The Image Interface can operate in four modes:

- The input gray level image process
- The output gray level image process
- The input binary image process
- The output binary image process

In each mode, the user can configure the left/right and Up/Down direction for the Read/Write process and the number of rows to read or write, defining the initial and final rows. This means that the windowing functionality is available only for rows.

4.1.1.3.1. Reading binary images from the Q-Eye

For binary images, each bit of data bus DAT_IM<31:0> corresponds to a pixel value. Therefore, the system reads or writes 32 pixels in each cycle of external master clock CLK.

Figure 9 depicts the timing diagram for the reading process of a binary image. The I/O Image Interface has a pipeline architecture. Then, 8 cycles of external master clock are required from the rising edge of the launching control signal START_IM to download the first valid data. From the eighth cycle, the data flow is continuous. The first row is the row defined by the initial row register. Each row is composed by the concatenation of five 32 bits words of data bus DAT_IM<31:0>. The last word has only 16 valid bits (the Most Significant Bits), the Last Significant Bits are fixed to zero.

Due to the pipeline architecture of Image Interface, an additional dummy row reading is required at the beginning of reading process and the internal control circuitry required 3 cycles of master clock to start the process, this is the reason of these eight additional cycles.

At the end of the reading process the ending signal END_IM commutes from the low level to the high level indicating the end of download process.

The number N_{read} of external master clock CLK required to read a complete binary image is:

$$N_{read} = \left(N_{row} + 1\right) \cdot 5 + 3$$

Where, N_{row} is the number of row to read and the eight additional cycles are due to the pipeline architecture.

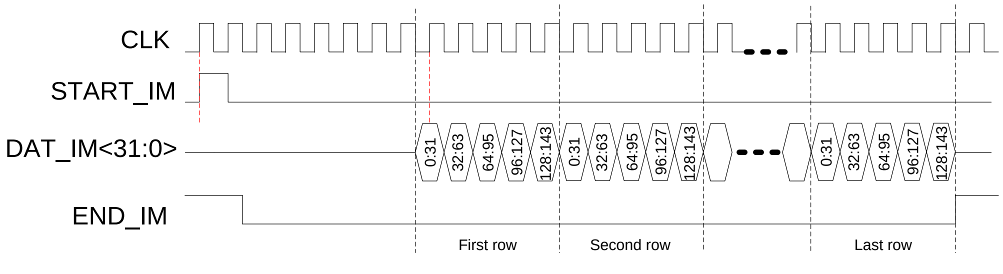


Figure 9. TIMING DIAGRAM FOR BINARY IMAGE READING PROCESS.

4.1.1.3.2. Writing binary images into the Q-Eye

Figure 10 represents the timing diagram of writing process for a binary image. One cycle of external master clock after the rising edge of launching control signal STAR_IM, the external host can start writing into the Q-Eye system the binary data. The Q-Eye composes the first five 32 words to obtain the first row to load into the processing array. Due to the pipeline architecture five additional cycles are required to load the last row into the memories of the array. The ending control signal can be registered in the sixth master clock cycle after writing the last row.

The number N_{write} of external master clock CLK required to write a complete binary image into the Q-Eye is given by:

$$N_{write} = (N_{rows} + 1) \cdot 5$$

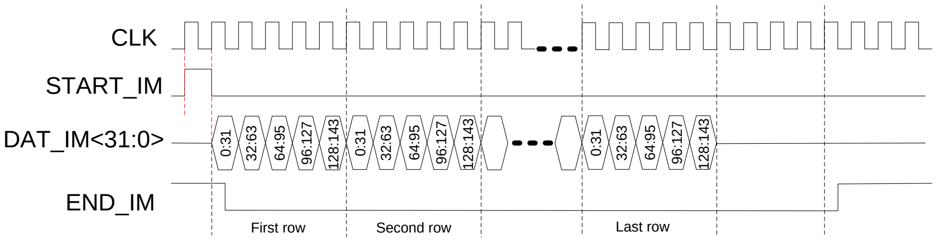


Figure 10. **TIMING DIAGRAM FOR BINARY IMAGE WRITING PROCESS.**

4.1.1.3.3. Reading gray level images from Q-Eye

The Write/Read processes of gray level images follow a pipeline architecture similar to the binary image interface. In this case, the data associated to each pixel is determined by a 8 bits word. Then, the data bus DAT_IM<31:0> is composed by for pixel data of 8 bits. To load a pixel row the external host required 36 cycles of external master clock CLK:

$$\frac{144 \quad columns}{4 \quad columns} = 36 \quad cycles$$

Figure 11 describes the timing diagram for the reading process of gray level images.

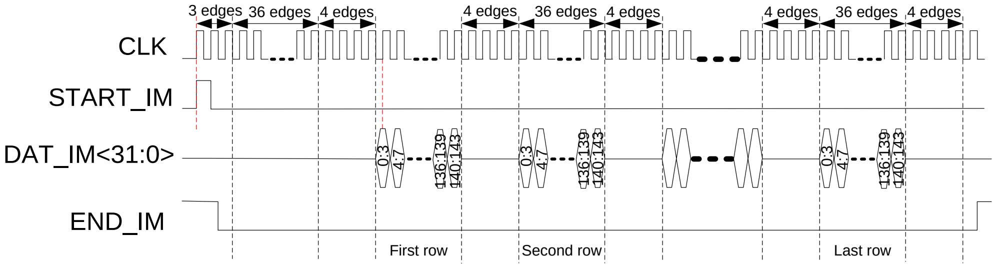


Figure 11. **TIMING DIAGRAM FOR GRAY LEVEL IMAGE READING PROCESS.**

During reading the analogue data of each row is loaded into the analogue memory bank of image interface in parallel, and then, these voltages are converted by the four internal ADCs.

Due to the pipeline architecture of the image interface, an additional dummy row reading is required at the beginning of reading process. In addition, the internal control circuitry required 3 cycles of master clock to start the process.

4.1.1.3.4. Writing gray level images into Q-Eye

Figure 12 depicts the timing diagram of writing process of a gray level image.

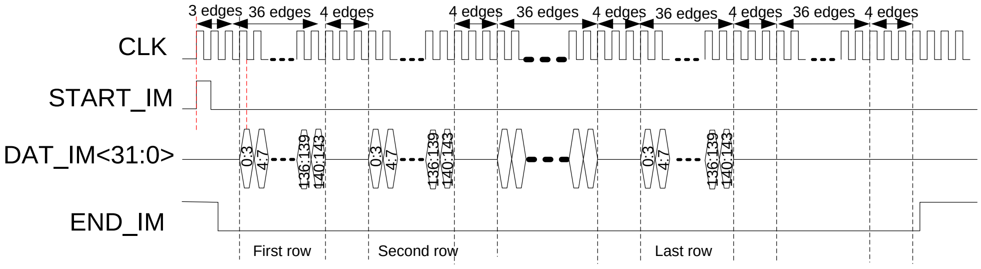


Figure 12. **TIMING DIAGRAM FOR GRAY LEVEL IMAGE WRITING PROCESS.**

The external host can start writing the data into the Q-Eye system after four cycles of the external master clock following the rising edge of launching control signal STAR_IM. The Q-Eye composes each 4 8-bits words to convert them into analogue voltages through the internal DACs. These analogue voltages are stored in analogue memories to load the complete row into the processing array in parallel. Due to this pipeline architecture an additional dummy row is required to load the last row into the memories of the processing array.

4.1.2. CONTROL STRATEGIES: CONFIGURATION REGISTERS AND PROGRAMMABLE CONTROL UNIT

The main internal blocks of the Q-Eye system, which are indicated in Figure 2, have several operation modes and parameters which are defined by the content of I/O registers called Configuration registers. These registers can be read and written by the user through the Programming Interface. For instance, the I/O Image Interface can be configured to read/write binary/gray level images into the Processing array. The operation mode of Image Interface is configured by the user using these I/O registers. In a similar way, others parameters of system like the internal analogue reference, the delays of clock signals in particular internal blocks, etc can be defined by the Configuration registers. The control of Processing Array is developed by the Programmable Control Unit which is controlled and programmed through the Programming Interface. The Configuration register and Programmable Control Unit are described below.

4.1.2.1. CONFIGURATION REGISTER

The configuration registers are Input/Out registers which can be written and read by an external host through the Programming Interface. The configuration register has two Input/Output ports. One port is connected to the Programming Interface, called external port, and the other port is connected to the corresponding internal block of Q-Eye system whose operation mode and configuration is defined by this Input/Output register The latter is called internal port.

Figure 13 describes the ports of configuration registers. The external port is composed by the input data bus dat_pb<9:0>, the output data bus dat_pb_o<9:0>, the address bus add_pb<11:0> and the read/write control signal rw_pb. The internal port includes an input data bus dat_int_i<9:0>, an output data bus dat_int_o<9:0> and a read/write control signal rw_int. In addition to these ports, the configuration register has two additional signals the internal system clock ck_control and the global reset signal reset.

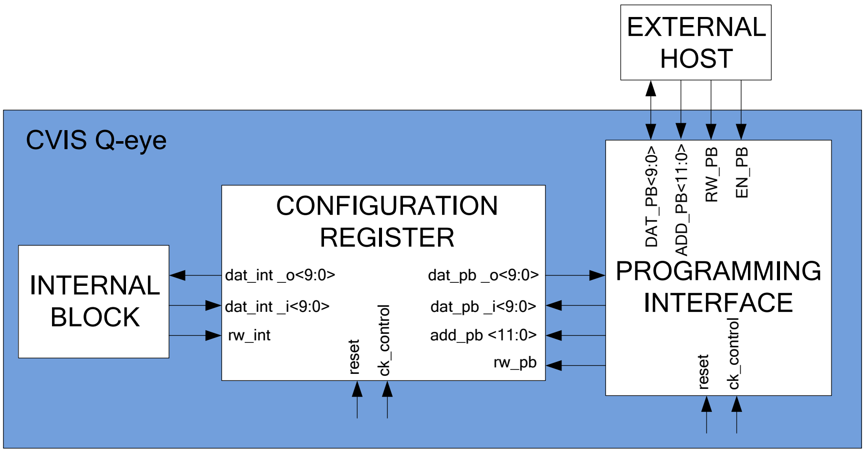


Figure 13. COMMUNICATION PORTS OF CONFIGURATION REGISTER.

The user can access to the configuration register through the external port using the Programming interface, see section 4.1.1.2 The address reserved for the access to the configuration registers is indicated in Table 2:

- add_pb<11:0>=b'0010XXXXXXXX
- add_pb<11:0>=b'0011XXXXXXXX
- add_pb<11:0>=b'0100XXXXXXXX

- add_pb<11:0>=b’0101XXXXXXXX

The internal block can write into the Input/Output register through the input data bus *dat_int_i<9:0>* when the read/write control signal of internal port *rw_int* is set to low level ‘0’.

The Input/Output register is a shared resource between the Programming Interface and the particular internal block. In case of conflict, the external port associated to the Programming Interface has priority over the internal port.

The data stored in the register configures directly the corresponding internal block through the output data bus *dat_int_o<9:0>* of internal port. The output data bus *dat_int_o<9:0>* is updated with the rising edge of internal system clock *ck_control*. The output data bus of internal port takes the data stored in the register when the read/write control signal of internal port *rw_int* is set to high level ‘1’. And it takes the data in the input data bus *dat_int_i<9:0>* when the control signal *rw_int* is set to low level ‘0’.

4.1.2.2. PROGRAMMABLE CONTROL UNIT (PCU)

Proper control is needed to exploit the parallel processing capabilities of the Q-Eye. As already explained in Chapter 3, the Q-Eye operation follows the principle of Single Instruction Multiple Data architectures (SIMD), namely, all pixels perform the same operation in parallel manner under the control of the Programmable Control Unit – see Figure 14. Control is exercised through a vector of sixty control signals, denoted by Focal Plane Processor (FPP) instruction which is carried by the output bus of the PCU *PCU_bus<59:0>*.

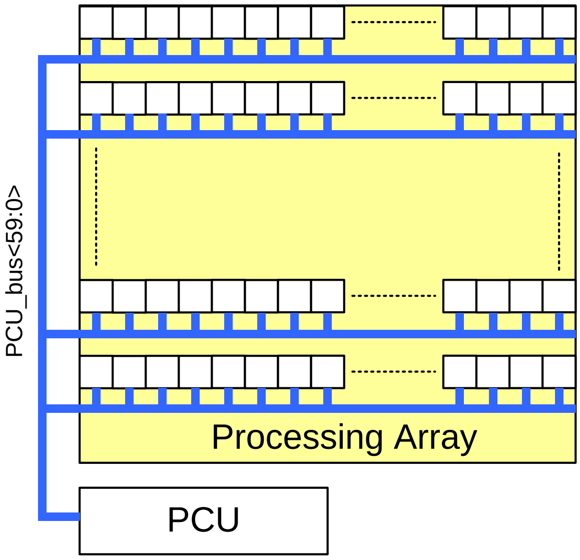


Figure 14. SIMD ARCHITECTURE.

Figure 15 describes the architecture of the PCU which is composed by an SRAM (Static Random Access Memory) block, called PBMEM, where the sequence of FPP instructions is stored, an output 60 bits register, called PBCTRL register, whose output constitutes the array control bus *PCU_bus<59:0>* and a 60 bits register located between the program memory and PBCTRL register, called PBMASK register. The PBCTRL register is updated with the content of the program memory in the address defined by the Programming Interface address bus ADD_PB<11:0>. This writing process is conditioned or masked by the content of PBMASK register. This PBMASK register can be updated from the program memory or instruction memory. The objective of *PBMASK* register is to increase the number of instructions which can be stored in the *PBMEM* block and to increase the capability of programmability. This architecture of the control unit PCU permits to implement more complex algorithms than basic operations, without losing the operation speed that characterizes the parallel processing, ensuring high operation speed and programmability.

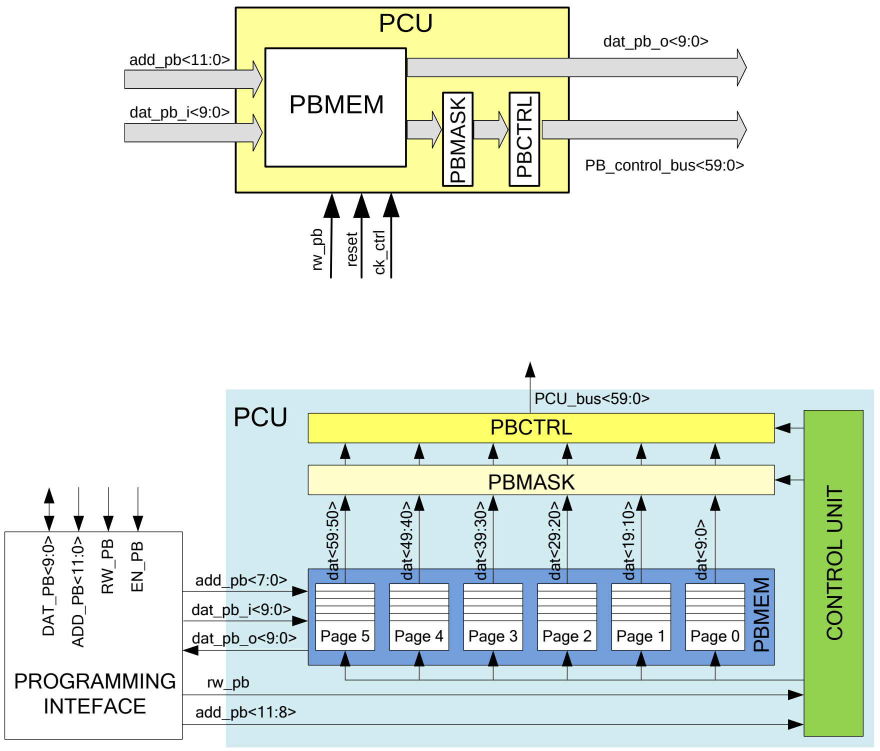


Figure 15. PROGRAMMABLE CONTROL UNIT ARCHITECTURE.

The external host controls the PCU operation through the Programming Interface. The PCU operations are defined by the four Most Significant Bits of address bus ADD_PB<11:8> and the read/write control signal RW_PB, see Table 3. Next, these PCU operations are presented:

- Write the program memory PBMEM through the data bus DAT_PB<9:0>.
- Read the program memory PBMEM through the data bus DAT_PB<9:0>.
- Write the PBCTRL register from the program memory.
- Read the PBCTRL register through the data bus DAT_BUS<9:0>.
- Write the PBMASK register from the program memory.
- Read the PBMASK register through the data bus DAT_BUS<9:0>.

ADD_PB<11:8>	RW_PB	Description
0000	0	Write PBCTRL register from the program memory
0001	0	Write PBMASK register from the program memory
1000	0	Write the page 0 of program memory PBMEM through the data bus DAT_PB
1000	1	Read the page 0 of program memory PBMEM through the data bus DAT_PB
1001	0	Write the page1 of program memory PBMEM through the data bus DAT_PB
1001	1	Read the page 1 of program memory PBMEM through the data bus DAT_PB
1010	0	Write the page 2 of program memory PBMEM through the data bus DAT_PB
1010	1	Read the page 2 of program memory PBMEM through the data bus DAT_PB
1011	0	Write the page 3 of program memory PBMEM through the data bus DAT_PB
1011	1	Read the page 3 of program memory PBMEM through the data bus DAT_PB
1100	0	Write the page 4 of program memory PBMEM through the data bus DAT_PB

1100	1	Read the page 4 of program memory PBMEM through the data bus DAT_PB
1101	0	Write the page 5 of program memory PBMEM through the data bus DAT_PB
1101	1	Read the page 5 of program memory PBMEM through the data bus DAT_PB

Table 3. PCU operations.

The program memory is composed by six SRAM blocks, each one constitutes a memory page. The page address is defined by the four MSBs of Programming Interface address bus ADD_PB<11:8>. And in each memory page, the address word is defined by the eight LSBs in the address bus ADD_PB<7:0> of Programming Interface. The address word defined by ADD_PB<7:0> is the same for the six memory pages.

The program memory has 256 words of 60bits. Therefore, to write a 60 bits word define by the address ADD_PB<7:0> is required 6 writing access through the Programming Interface.

The writing process for the instruction register PBCTRL and the mask register PBMASK is similar. The word of program memory which will be written into the PBCTRL or PBMAK register is selected by the eight LSBs of address bus ADD_PB<7:0>. The destination register is defined by the four MSBs of address bus ADD_PB<11:8>. The read/write control signal RW_PB must be et to low level during the writing process. Figure 6 and Figure 7describe the timing diagram for the Programming Interface.

The content of mask register PBMASK conditions the writing process of instruction register PBCTRL. During the writing process of PBCTRL by the program memory, the instruction bits of PBCTRL whose corresponding mask bits of PBMASK register are zero will be updated. The mask bits set to one masks the writing of corresponding bit in the instruction register PBCTRL.

4.1.3. MISCELLANEOUS BLOCKS

These are auxiliary mixed-signal blocks which have been integrated to reduce the number of external components required for in-field operation. They generates internally all analogue references, clock signals and special control signals required by the main internal blocks like the Processing Array, the Image Interface, the Programming Interface and Programmable Control Unit. Some of these miscellaneous blocks implement useful functionalities like the internal temperature sensor.

4.1.3.1. INTERNAL REFERENCE GENERATOR

The mixed-signal blocks of the Q-Eye require analogue references for biasing and operation. The **INTERNAL REFERENCE GENERATOR**, see Figure 16, generates these analogue references. It contains a *band-gap block* that generates a voltage V_{bg} which is in its turn used to generate an analogue reference V_{REF} required by the DACs of the *programmable buffers* shown in Figure 19. Moreover, the band-gap block generates a reference current I_{REF} to obtain the biasing voltage V_{biasg} for the mixed-signal blocks of the Q-Eye.

Figure 17 shows the band-gap topology, where the diodes have been implemented using vertical npn BJTs available in this CMOS technology, see Figure 18. This topology has been chosen to generate a reference voltage and biasing current stabilized over process, power supplies and temperature variations. The resistances of R1 and R2 are equal and all PMOS transistors M1, M2, M3, M4 have the same sizes. Thus, the output voltage of Band-gap is given by the equation:

$$V_{bg} = \frac{R_4}{R_2}\left(V_{d1} + \frac{R_2}{R_3}\Delta V_d\right) \qquad \textbf{Eq. 309}$$

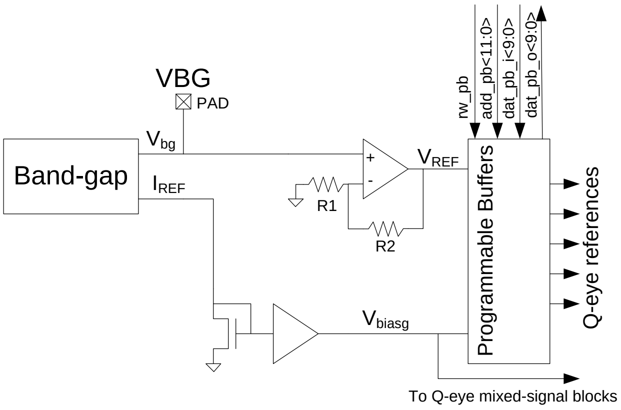


Figure 16. INTERNAL REFERENCE GENERATOR ARCHITECTURE.

Where V_{d1} is the built-in voltage of the diode D1 and $\Delta V_d = V_{d1} - V_{d2}$. The Base-Emitter voltages V_{d1} and V_{d2} are determined by the equations:

$$V_{d1} = V_T \cdot \frac{I_{D1}}{I_S} \qquad \textbf{Eq. 310}$$

$$V_{d2} = V_T \cdot \frac{I_{D2}}{I_S} \qquad \textbf{Eq. 311}$$

where I_S is the saturation current and $U_T = \frac{k \cdot T}{q}$ the thermal voltage. The increment voltage ΔV_d is equal to:

$$\Delta V_d = V_{d1} - V_{d2} = U_T \cdot \ln(N) \qquad \textbf{Eq. 312}$$

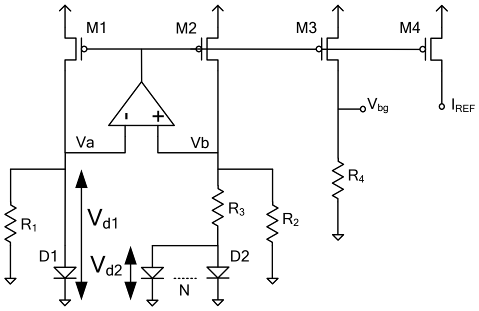


Figure 17. BAND-GAP ARCHITECTURE.

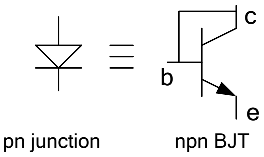


Figure 18. PN JUNCTION IMPLEMENTATION.

The Base-Emitter voltage V_{d1} has a negative temperature coefficient whereas U_T has a positive temperature coefficient. With the resistance ratio $\frac{R_2}{R_3}$ the operation point where the temperature dependence of the band-gap reference becomes negligibly small is determined.

And with the resistance R_4 the absolute value of the Band–gap reference V_{bg} is defined. Therefore, this architecture obtains band-gap reference voltages below the 1.25V obtained with the conventional topology [Banb99], and can be designed with values below 1V.

The **Programmable Buffers** are the responsible for the generation of internal references required by the operation of system, the Processing Array operation and Image Interface operation. These internal analogue voltages or references are described in the Table 4.

Reference	**Description**	**Functional Block**
V_{pch}	Precharge voltage of pixel	Optical sensor in processing cell.
V_{hpch}	Control signal of reset transistor in pixel	Optical sensor in processing cell.
V_{hold}	Reference used during hold configuration in analogue memories	LAM block and Global Reference block in cell.
V_{offset}	Reference to define the offset in MAC operation.	MAC block and Global Reference block in cell.
V_{misc}	General purpose reference	Global Reference block in cell.
V_{bp}	Reference to define the biasing current of processing cell	Biasing of processing cell.
V_{bn}	Reference to define the biasing current of processing cell.	Biasing of processing cell.
V_{max}	Maximum level of analogue signal range.	AD/DA Converters in Image Interface.
V_{zero}	Zero level of analogue signal range.	Sampled & Hold in Image interface.
V_{min}	Manimum level of analogue signal range.	AD/DA Converters in Image Interface.

Table 4. Internal analogue references in the Q-Eye.

Figure 19 depicts the architecture of this buffer. It consists of a 8-bits DAC whose digital input data is defined by the configuration registers. This DAC is a decoder-based converter, where the 256 references are generated by a resistor string. Figure 46 in Section 4.2.1.3.4 depicts this circuit. Regarding the configuration registers, they control the operation of amplifier, their biasing and the *offset calibration block*. The biasing of amplifier is configurable to adapt the bandwidth of output buffer to the load capacitance and speed requirement determined by the function of reference voltage in the Q-Eye system.

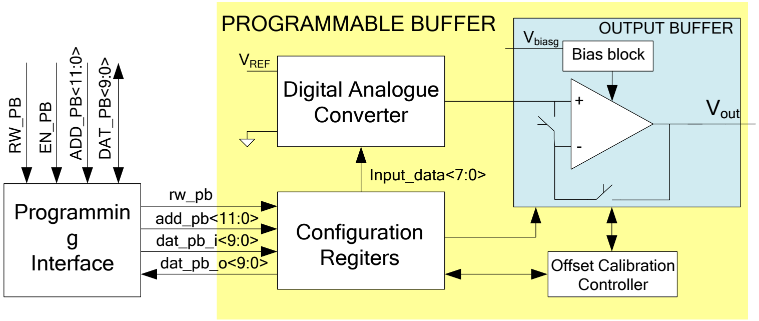


Figure 19. Programmable Buffer architecture.

The digital word with a length of 8 bits $D_{in}=[b_{n-1},b_{n-2},\ldots,b_1,b_0]$ configures the analogue voltage value in basis to the equation:

$$V_{out}=\frac{V_{REF}}{2^n}\left(b_0+b_1\cdot 2+b_2\cdot 2^2+\ldots+b_{n-1}\cdot 2^{n-1}\right) \qquad \textbf{Eq. 313}$$

The output buffer in Figure 19 is a two stage Miller OTA [Malo01], which consists of a folded-cascode first stage and an inverter as second stage. The OTA circuit is shown in Figure 20. Several configuration signals are employed in this circuit. The amplifier can be configured to work with NMOS input transistors, PMOS input transistors or both with the signals “par_p” and “par_n”. The biasing of input transistor pairs is configured through the signals “cfg_bias_parp<1:0>” and “cfg_bias_parn<1:0>”. When the biasing of input pair is changed, the cascode voltages V_{casp} and V_{casn} and bias voltage V_{bias} of folded stage are adapted to the new values of input pair current. The transconductance associated to the input pair (g_{mi}) depends on the bias current of the input transistors. Moreover, the unity gain frequency is determined by this input transconductance and the Miller compensation capacitor C_c .

$$\omega_T = 2 \cdot \pi \cdot f_T = \frac{g_{m1}}{C_c} \qquad \textbf{Eq. 314}$$

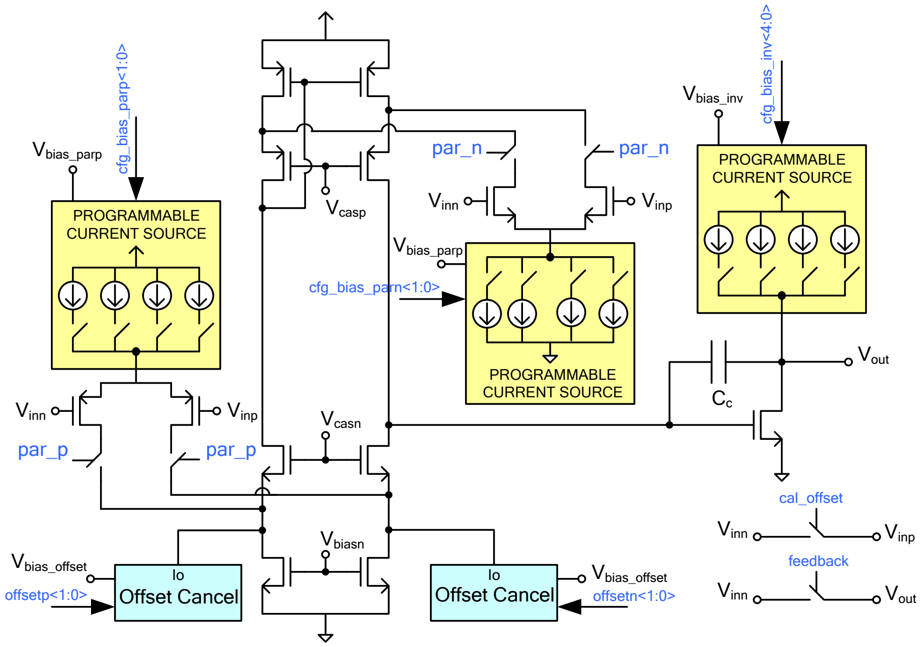


Figure 20. OUTPUT BUFFER CIRCUIT.

Depending on the speed requirements for the analogue reference, the OTA unity gain frequency can be tuned through the bias current of input pair. The dominant and second poles and zero of two stage amplifier depend on transconductance associated to the driver of output inverter (g_{m2}). Considering the small signal equivalent circuit depicted in Figure 21, the poles and zero are given by :

$$p_1 \cong -\frac{1}{g_{m2} \cdot r_2 \cdot r_1 \cdot C_c} \qquad \textbf{Eq. 315}$$

$$p_2 \cong -\frac{g_{m2} \cdot C_c}{C_1 \cdot C_2 + (C_1 + C_2) \cdot C_c} \qquad \textbf{Eq. 316}$$

$$z = \frac{g_{m2}}{C_c} \qquad \textbf{Eq. 317}$$

This transconductance g_{m2} is determined by the bias current of inverter. r_1 and C_1 are the output resistance and output capacitance respectively associated to the first stage. And r_2 and C_2 are the corresponding parameters of second stage.

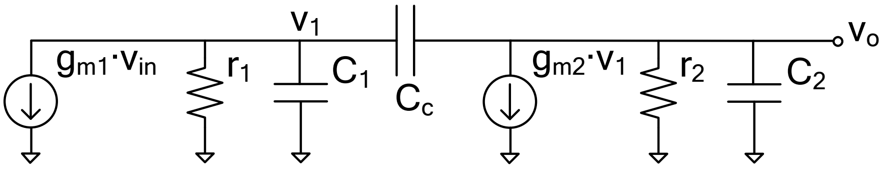


Figure 21. SMALL SIGNAL EQUIVALENT CIRCUIT OF TWO STAGES OTA.

To guarantee amplifier stability, the bias current of the second stage can be tuned with the configuration signals "cfg_bias_inv<4:0>". The second pole and zero of two stage amplifier depends on transconductance associated to the driver of output inverter and the output load capacitance. And this transconductance is determined by the bias current of inverter. The output load capacitance is defined by the functionality of analogue reference. Therefore, to ensure the stability of amplifier, the bias of output stage can be adapted to the load capacitance requirements for each reference. These configuration signals are fixed by the user through the I/O configuration registers included in each **Programmable Buffer**.

There are two configuration signals, "feedback" and "cal_offset" in the registers which configure the operation of amplifier. Activating "feedback" signal, the amplifier works in unit gain configuration. With "feedback" signal set to 0 the amplifier works in open loop. This configuration is used during the automatic offset calibration process. For this offset calibration, "cal_offset" signal is activated to short the inputs of amplifier in order to measure the offset associated to the amplifier.

In the Programmable Buffer, an automatic calibration process has been implemented to compensate the output OTA analogue offset. During the offset calibration process, the OTA works in open loop carrying out the comparison function when its inputs are short-circuit to a constant voltage level defined by the DAC. The **Offset Calibration Controller** modifies the biasing of output braches in the OTA through the control signals "offset<4:0>" and "offset<4:0> in function to the result of comparison operation to compensate the systematic analogue offset, following a *successive approximation algorithm*. Figure 22 describes the offset cancellation block which modified the biasing of output branches to compensate the mismatch between currents by following the algorithm in Figure 23. The automatic offset calibration process is configured and launched from the configuration registers.

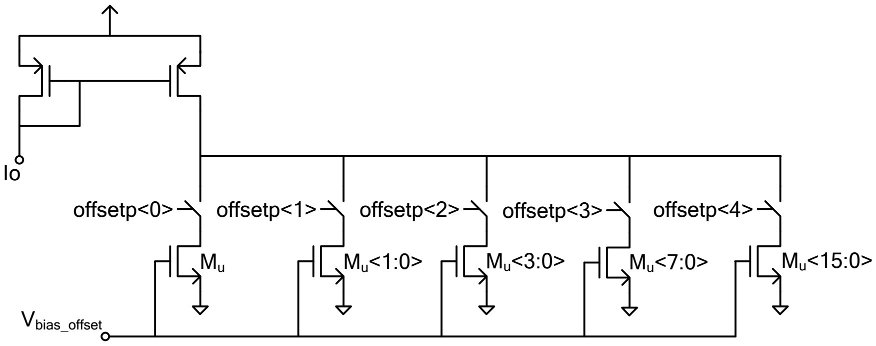


Figure 22. OFFSET CANCEL BLOCK.

Another important system-level aspect is the output noise of reference generators, because this noise may impact the performance of analogue circuits limiting the temporal noise floor and generates artifacts in the processed images. For instance, low frequency noise in the global references of the processing array may provoke line to line noise in one image or frame to frame noise in a sequence of images. Therefore, it is important to minimize the temporal noise associated to the analog reference in order to optimize the dynamic range of captured and processed images during the design phase of circuits. In addition, from a system level point of view techniques to compensate the line-to-line and frame-to-frame noise can be implemented. For instance, in standard CMOS image sensors optical black columns and rows are included to sense the offset error and compensate it by digital processing.

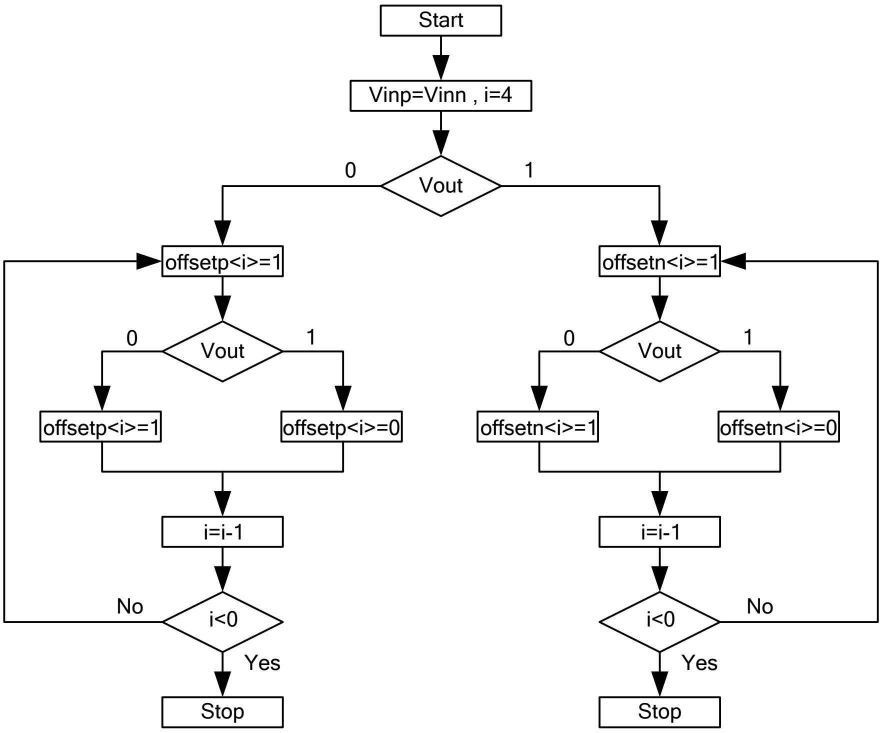


Figure 23. ALGORITHM FOR OTA OFFSET CALIBRATION.

4.1.3.2. CLOCK GENERATION UNIT (CGU)

It generates the four internal clocks used by the Q-Eye:

- the internal system clock *ck_control*,
- the AD converters clock *ck_ad*,
- the DA converters clock *ck_da* and
- the Image Interface Control clock *ck_row_col_io*.

The **INTERNAL SYSTEM CLOCK** *ck_control* is the clock for the configuration and control of Q-Eye system. It is the clock of the **programming interface**, the **programmable control unit** and the **configuration registers**.

The **IMAGE INTERFACE CONTROL** clock *ck_row_col_io* is the clock for the **control machine of image interface**. This control machine controls the operation of the processing array row by row and synchronizes this operation with the operation of the sample & hold bank in thei interface during the read/write process. The AD converters clock ck_ad is the clock of AD converters included in the image interface. And finally, the DA converters clock is the clock of DA converters included in the image interface.

Figure 24 depicts the architecture of **CLOCK GENERATION UNIT**. The internal clocks of Q-Eye system are generated from the external master clock CLK by the CGU. Each internal clock is generated from the external master clock by applying a programmable delay. This delay is configurable through the configurations registers. Each internal clock, except the internal system clock *ck_control*, can be disabled through the corresponding configuration register.

The delay block associated to the internal system clock can be bypassed using the external control signal CTRL_CLK.

4.1.3.3. POWER ON RESET (POR)

This (Figure 25) generates an initial internal reset pulse following power ON to reset all flip-flops and latches included in the digital blocks of Q-Eye system.

Control signals EN_ROPU and RESET belong to the External Interface and they are control by an external host. EN_ROPU enables the initial internal reset pulse and RESET is the external reset of Q-Eye system.

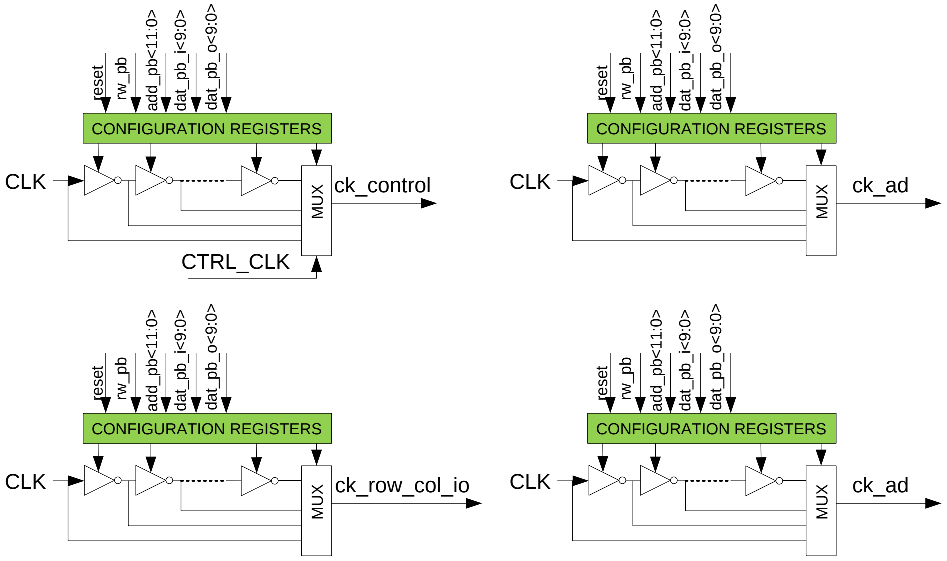


Figure 24. Clock Generation Unit architecture.

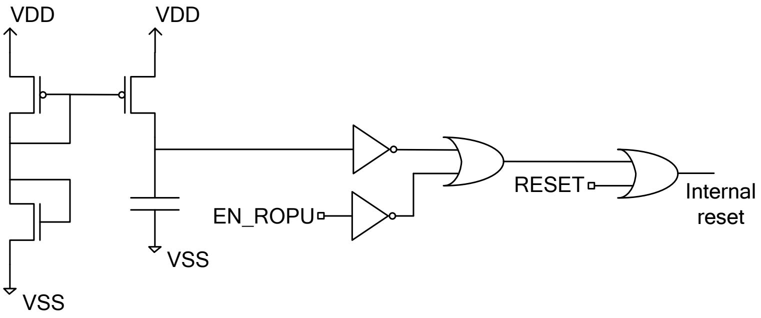


Figure 25. POWER ON RESET ARCHITECTURE.

4.1.3.4. MONO-STABLE

It produces a pulse with programmable width as required to control the diffusion time during the **resistive grid** operation, see section 3.2.8 in Chapter 3. Figure 26 shows the architecture of Mono-stable block. The programmable pulse width can be smaller than the period of internal system clock *ck_control* (20ns), and its value is defined through the programmable current source and capacitance with the configuration registers.

The Mono-stable is controlled by the control signal *rg* which is one bit belong to the FPP instruction. The rising edge of control signal *rg* will generate only one pulse with a width programmed through the configuration registers. And the falling edge resets and configures the Mono-stable for a new pulse generation. The Q-Eye system can be configured to either apply directly the control signal *rg* or to apply the pulse generated by the Mono-stable.

A critical point in the generation of control for the diffusion process is the distribution of this control signal to the processing cells. There is a tradeoff between the time constant associated to the diffusion grid and the area required by the resistances and capacitances which constitute this resistive network. Therefore, due to the limited area available to implement the resistive network element in the processing cell the maximum value of this constant time, which determines the diffusion length together with the diffusion time, is limited and usually is smaller than the period of the internal system clock. Therefore, to implement Gaussian filters with short diffusion lengths it is required short diffusion times [Fern12]. This is the reason to include the Mono-stable block in the Q-Eye system. And this is the reason that does critical the distribution of the diffusion control signal. It is very important that the diffusion control pulse does not degrade during the transmission to the processing cells. The resistance and capacitance parameters of metal network to transmit the diffusion control signal are key parameters in the implementation of diffusion operation.

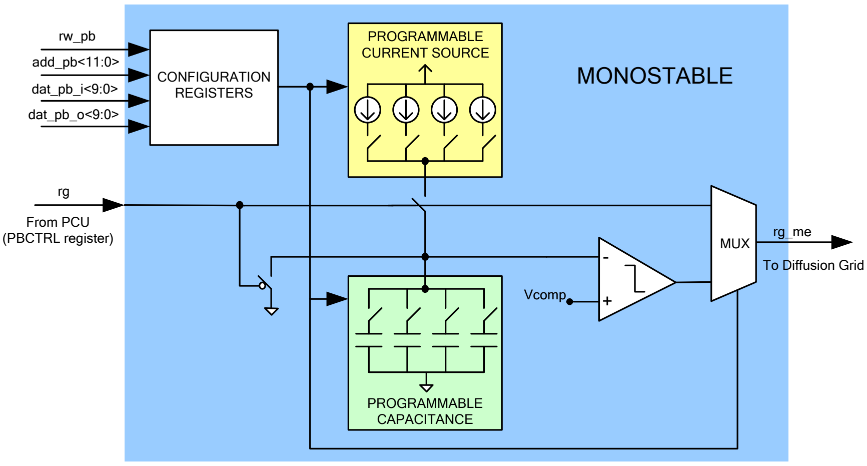


Figure 26. **MONO-STABLE ARCHITECTURE.**

4.1.3.5. MISCELLANEOUS ADC

The Q-Eye system includes a *successive approximation ADC* [Rodr14] [Rodr15] with the architecture of Figure 27. It employs the binary search algorithm of Figure 28 and is used to enable additional control by an external host. Including this general purpose ADC enables additional control functionalities which are relevant for vision applications. For instance, to set the exposure time value depending on the global mean value of image, for auto-calibration, to control chip temperature, etc.

Several internal voltages of the system can be connected to the input of the converter and the result of conversion will be stored in an input/output register belong to the Programming Interface. In this way, an external host can access the value of these internal voltages and acts on the Q-Eye system consequently. Internal voltages which can be connected to this ADC are all internal references, the voltage resulting from the global mean operation in the processing array and the output voltage of temperature sensor.

The internal voltage which will be connected at the input of Miscellaneous ADC is determined through a configuration register. The conversion is launched through a control bit included in the configuration registers associated to this Miscellaneous ADC. When this control bit is switched from low level to high level the conversion starts. The Miscellaneous converter indicates the end of conversion activating a flag bit belong to the configuration registers of Miscellaneous ADC.

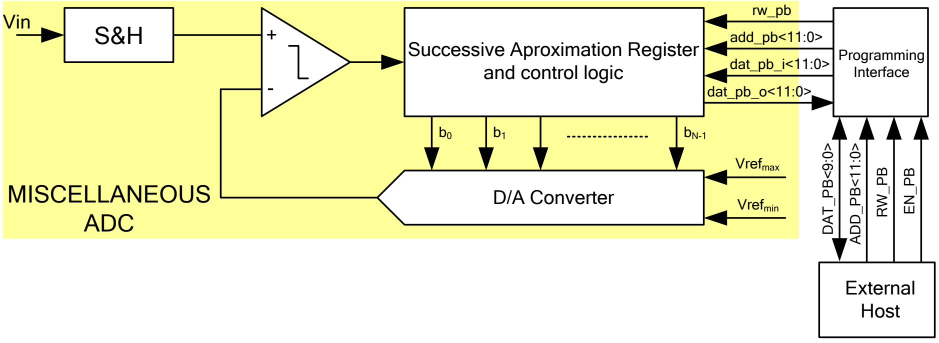


Figure 27. **MISCELLANEOUS ADC ARCHITECTURE.**

4.1.3.6. TEMPERATURE SENSOR

It is based in the generation of *Proportional To Absolute Temperature* (PTAT) voltage, whose mechanism is described in Figure 29 [Raza01]. If two bipolar transistors operate at unequal current densities, then the difference between their base-emitter voltages is directly proportional to the absolute temperature.

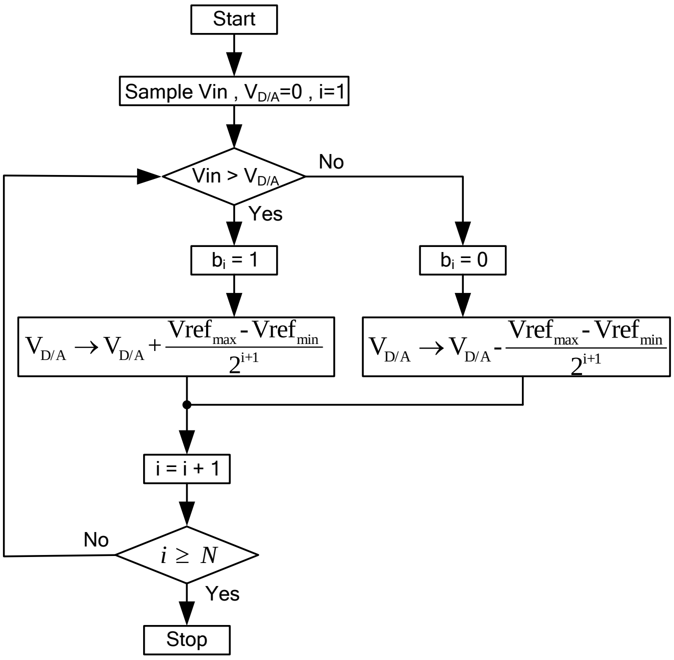

Start
Sample Vin , $V_{D/A}$=0 , i=1
Vin > $V_{D/A}$
No
Yes
b_i = 1
b_i = 0
$V_{D/A} \rightarrow V_{D/A} + \frac{Vref_{max} - Vref_{min}}{2^{i+1}}$
$V_{D/A} \rightarrow V_{D/A} - \frac{Vref_{max} - Vref_{min}}{2^{i+1}}$
i = i + 1
No
$i \geq N$
Yes
Stop

Figure 28. Binary Search Algorithm.

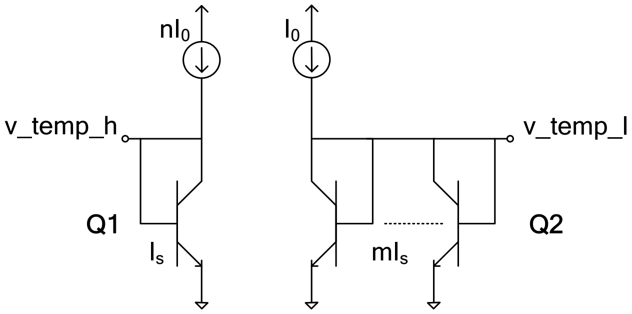


Figure 29. PTAT VOLTAGE GENERATION.

Considering two BJT transistors, Q1 composed by one unitary transistor and Q2 implemented by m unitary transistors, and biased by collector currents nI_0 and I_0 respectively, the difference between their base-emitter voltages is given by:

$$\Delta V_{BE} = v\_temp\_h - vtemp\_l = V_T \cdot \ln \frac{n \cdot I_0}{I_s} - V_T \cdot \ln \frac{I_0}{m \cdot I_s}$$
$$\Delta V_{BE} = V_T \cdot \ln(n \cdot m) = \left(\frac{k}{q} \cdot \ln(n \cdot m)\right) \cdot T = C \cdot T \qquad \text{Eq. 318}$$

where their base currents have been considered negligible. The voltage difference is proportional to the absolute temperature with a positive temperature coefficient:

$$\frac{\partial \Delta V_{BE}}{\partial T} = \frac{k}{q} \cdot \ln(n \cdot m) = C \qquad \text{Eq. 319}$$

k is the Boltzmann constant and q is the electron charge.

The PTAT voltages are driven to a SC circuit (see Figure 30) to adapt the difference between the base-emitter voltages to the input range of Miscellaneous converter, which carries out the analogue-to-digital conversion. The result of this conversion is stored in a configuration register belong to the Programming Interface. Therefore, an external host has access to the internal temperature of Q-Eye system.

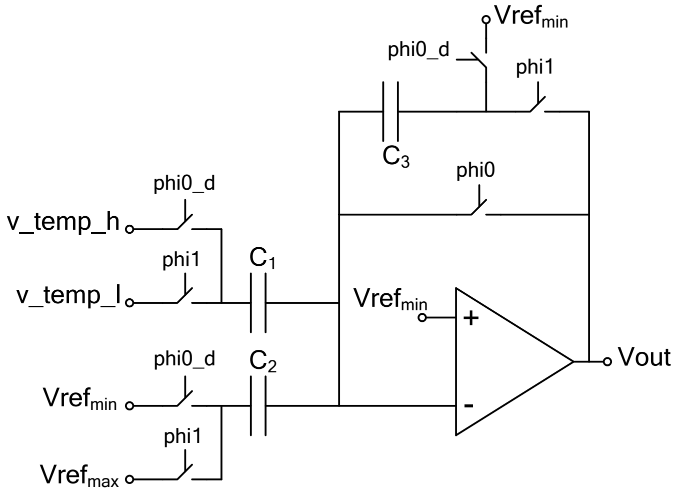


Figure 30. SIGNAL CONDITIONING CIRCUIT.

The behavior of SC circuit which develops the signal conditioning before AD conversion is described by the equation:

$$Vout = \frac{C_1}{C_3}\cdot(v\_temp\_h - v\_temp\_l) - \frac{C_2}{C_3}\cdot(Vref_{\max} - Vref_{\min}) + Vref_{\min} \qquad \textbf{Eq. 320}$$

$$Vout = K_2\cdot(v\_temp\_h - v\_temp\_l) - K_1 + Vref_{\min} \qquad \textbf{Eq. 321}$$

The reference voltages Vref$_{max}$ and Vref$_{min}$ represent the input range of the Miscellaneous AD converter.

The temperature coefficient C is determined through a calibration process. The signal conditioning circuit is controlled through a specific configuration register. The timing diagram of control signals is depicted in Figure 31.

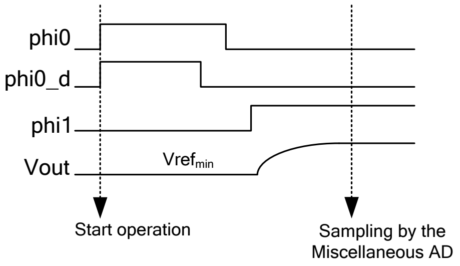


Figure 31. CONTROL SIGNALS.

At the end of phase phi1, the conditioning circuit output is connected to the input of Miscellaneous AD converter and the conversion is launched through the configuration registers associated to the converter.

4.2. THE Q-EYE READ-OUT

The Q-Eye includes circuitry for image *uploading* and *downloading*. Reading and writing operations follow a **row-by-row strategy**. The number of pixels read or written through the data bus of **IMAGE INTERFACE** per cycle of the external master clock (CLK) depends on the kind of image being handled, namely:

- In the case of **binary images**, the number of pixels read or written per master clock cycle is 32 because the width of data bus in the Image Interface is 32 bits, and one pixel in a binary image have two values associated, requiring only one bit for the codification.

- Regarding **gray-scale image**, and since their resolution is 8 bits, the number of pixels read or written per cycle of external master clock is 4.

Therefore, the maximum frame rate is 63100 frames per second (fps) for binary images and 7890 fps for gray images.

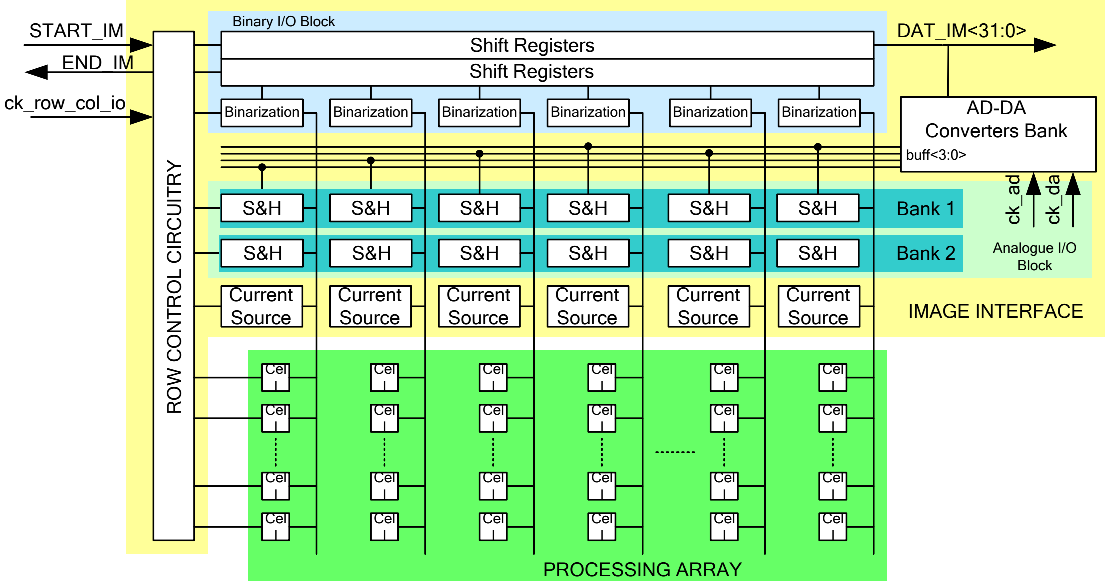


Figure 32. IMAGE INTERFACE ARCHITECTURE.

The **IMAGE INTERFACE,** whose architecture is described in the next subsections, is composed by five main blocks, see Figure 32:

- the **row by row control circuitry**, which controls the Processing Array during a read/write process; it also controls the Binary Input/Output block and the Analogue Input/Output block in the Image Interface.
- the **Binary Input/Output block**, which is the responsible for the read/write process of binary images.
- the **Analogue Input/Output block**, which handles the transmission of gray level images to the AD converters or from the DA converters.
- the **AD converters bank** and
- the **DA converters bank**.

4.2.1. ABOUT THE READOUT CHANNEL ARCHITECTURE

The readout channel of the Q-Eye starts in the processing cell, similar to any image sensor. The processing cell includes a specific functional block called Image I/O block to transmit voltage data associated to gray level and binary images from the Q-Eye pixel to the Image Interface. The Image I/O block drives the signal voltage from the local data node of the processing cell (*ladn*) to the data column node during an image download process. This is similar to any conventional image sensor architecture. However, the Q-Eye being a vision sensor it can also process electrically uploaded images and thus implement feedback in the way from optical images to electrical images. At the electrical level it requires the interface being bi-directional; i.e. provisions must be made to enable driving the signal voltage from the data column node to the local data node of cell during an image load process.

The Image I/O block contains the corresponding driver of the output Source Follower (SF) and selection switches. Figure 33 shows the schematics of this I/O block in the processing cell.

For each column in the Processing Array, there is a common input-output data node called data column node and denoted by *d_col<j>*, the j index represents the column number. These data column nodes constitute the interconnection bus between the Processing Array and the Image

Interface. It has already been mentioned that communication between the Processing Array and the Image interface happen row by row.

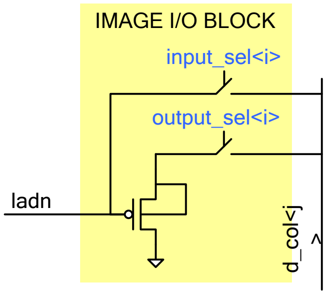


Figure 33. **ARCHITECTURE OF IMAGE I/O BLOCK IN THE PROCESSING CELL.**

READING OPERATION. During a reading operation, the output of SF driver is connected to the data column node. The signal voltage associated to the local analogue data node of processing cell (*ladn*) is fixed in the data column node *d_col* by the SF driver of Image I/O block. The corresponding SF drivers of row i are connected at the same time to the data column through the control signal *output_sel<i>* which is generated by the row control circuitry of Image Interface.

WRITING OPERATION. During a writing operation, the data column is connected directly to the local data node of processing cell through the control signal *input_sel<i>* which is activated row by row by the control circuitry of Image Interface.

It is worth referring to the cell configuration and the local analogue data node (*ladn*), bear in mind that this latter data will be driven by a memory during image reading and that a memory will be upload through the local data node during writing. Hence, the control of analogue or digital memories will be taken by the row control circuitry of Image Interface when the Q-Eye system is configured for a load/download process. The control of the Processing Array is developed by the Programmable Control Unit through the FPP instructions except during the load/download processes when the row control circuitry of Image Interface takes the control of the memories.

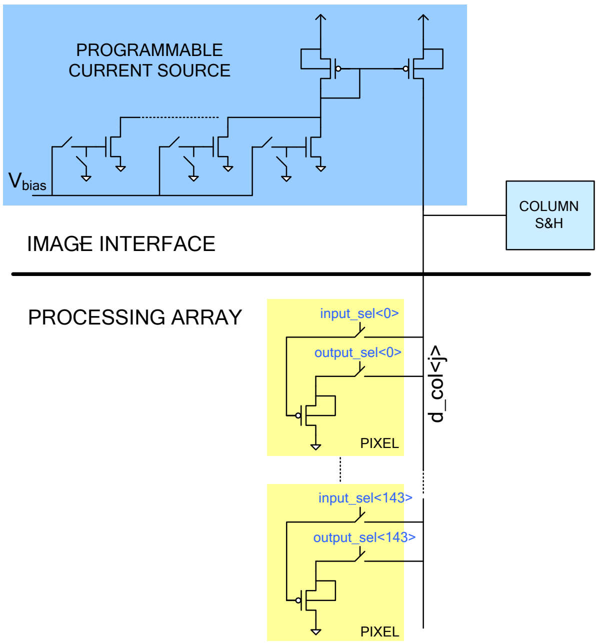


Figure 34. **OUTPUT COLUMN IN THE PROCESSING ARRAY.**

PROGRAMMABLE CURRENT SOURCE. As Figure 34 illustrates, each data column is driven at the image interface by a programmable current source. This current source together with the driver located in the processing cell constitutes the output Source Follower. The biasing current of output Source Follower in the processing cell is programmed through a configuration register.

SAMPLE-AND-HOLD AND BINARIZATION BLOCK. For each column, the data column node is connected to a Sample & Hold (S&H) block and a Binarization block - see Figure 32. The group of S&H blocks corresponding to all columns constitutes the Analogue I/O block of Image Interface. Equally, the group of Binarization blocks for all columns composes the Binary I/O block.

To relax the speed specifications during the data transfer between one row of Processing Array and the Image Interface, there are two S&H blocks for each column constituting two banks of S&H, see Figure 32. During a download process, one bank of S&H is reading one row from the Processing Array in parallel while the other bank is transmitting the data of previous row to the Converters Bank in serial manner. When the transmission from the Array to the S&H blocks finishes, the functionality is interchanged between banks. During a load process, the operation is analogous. While one S&H bank is loaded by the DA converters bank serially, the other S&H bank is transferring the data to one row of Processing Array in parallel. Note that there are four AD converters and four DA converters in the Converter Bank. Therefore, four Column Sampled and Hold are written or read per cycle of external master clock CLK.

A similar strategy has been implemented for the load/download process of binary images. Two banks of digital registers have been implemented, one bank is read or written serially by the External Interface while the other bank is written or read in parallel by the Processing Array through the Binarization blocks.

4.2.1.1. COLUMN SAMPLE & HOLD

Figure 35 shows the schematics of the Column Sample & Hold. It is a SC circuit similar to the MAC circuit implemented into the processing cell. In this case, the multiplicative factor $\frac{C_{in}}{C_{out}}$ is constant and equal to the unit ($C_{in} = C_{out}$).

Figure 36 shows the two operation modes of the S6H. During image uploading, it works as an analogue memory of the processing cell (Gregorian's S&H). During image downloading, it works as a Correlated Double Sampling (CDS) circuit.

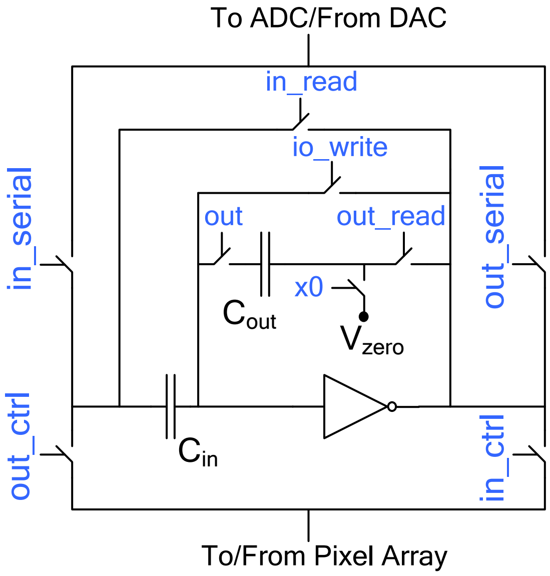


Figure 35. COLUMN SAMPLED & HOLD ARCHITECTURE.

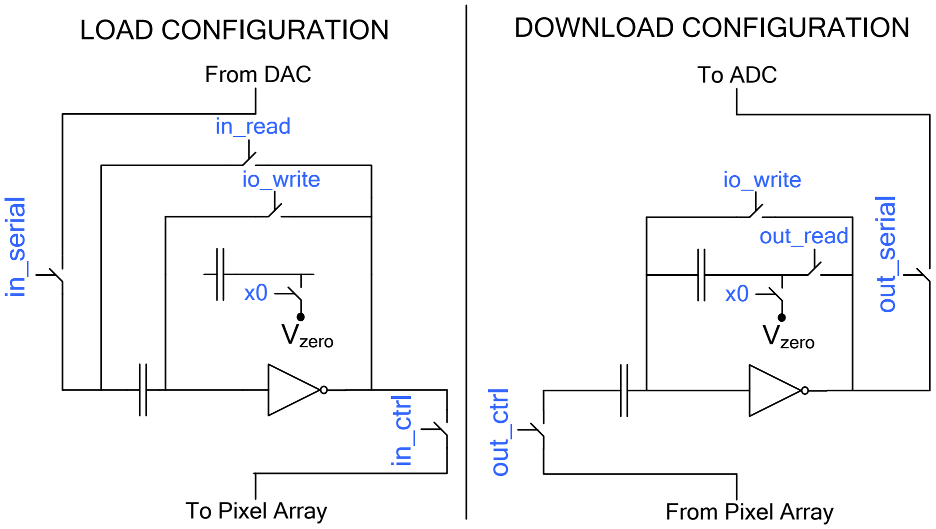


Figure 36. **OPERATION CONFIGURATIONS FOR COLUMN S&H.**

There are two banks of Column S&H. Therefore, during an image load process, one bank is taking the analogue data from DA converters in serial manner while the other bank is writing the analogue data into one row of pixel array in parallel manner. Figure 37 depicts the timing diagram of control signals implied in the load process. In a similar way, during an image download process, one bank is writing the analogue data into the AD converters in a serial way while the second bank is taking the data from a pixel row in parallel way. Figure 38 shows the timing diagram of control signals active in this download process.

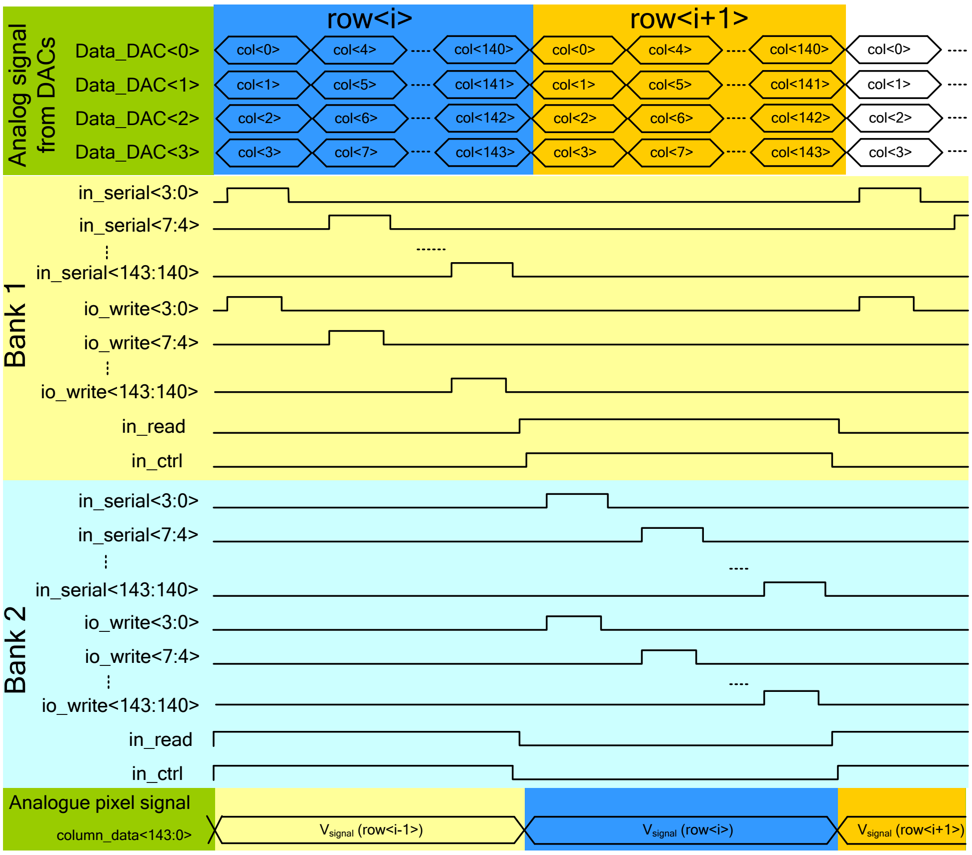


Figure 37. **CONTROL SIGNALS OF COLUMN S&H LOADING AN GRAY IMAGE.**

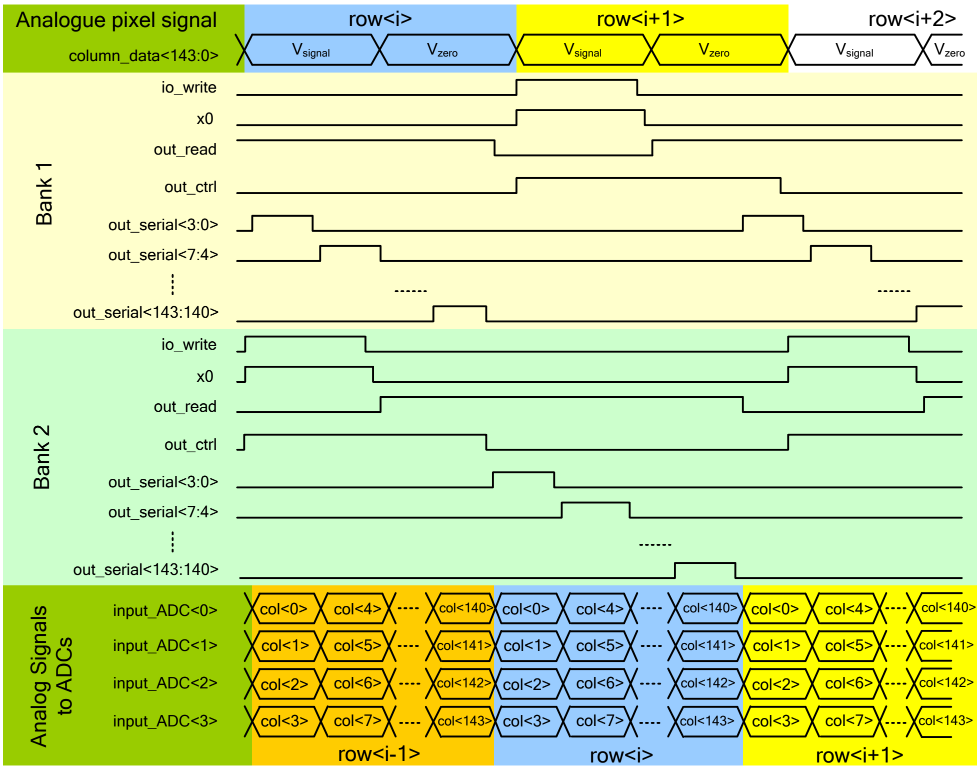


Figure 38. CONTROL SIGNALS OF COLUMN S&H DOWN LOADING AN GRAY IMAGE.

NONIDEALTIES IN THE S&H OPERATION. The impact of circuit non-idealities on the operation of S&Hs can be studied by using procedures similar to those in Chapter 3 for the SC circuits which constitute the analogue memories and MAC operator in the processing cell, see respectively sections 3.3.1 and 3.3.2.

The most important errors provoked by these non-idealities are:

- Temporal settling error.
- Charge injection and clock feedthrough.
- Offset and finite gain of amplifier.
- Mismatch between capacitors.
- Random temporal noise.
- Leakage in hold configuration.

These errors generate deviations respect to the ideal behavior of the Input/Output characteristic associated to the Column S&H circuit. These deviations impact the following S&H behavioral parameters:

- offset,
- gain error,
- non-linearity,

which in their turn may produce Fixed Pattern Noise per column.

From a design point of view, the errors introduced by the circuitry of image interface should be neglected respect to the errors generated by the sensing and processing circuitry at pixel level. The images resulting from the analog processing developed by the pixel circuitry and downloaded by the image interface should not show Fixed Pattern Noise (FPN) due to the non-idealities of readout circuitry. Since roughly speaking enlarging the area reduces the errors and taking into account that area restrictions are larger in the processing cell than in the column readout circuit, the impact of processing cell non-idealities on the image quality dominates versus that of interface circuit non-idealities.

The static errors associated to the processing cells generate 2-Dimensional spatial random noise in the image. But, the static errors associated to the column readout circuitry generate

VFPN. This VFP noise will not be detected if its value is ten times below the random noise either spatial or temporal [Snoe06].

The random temporal noise of Column S&H contributes to the noise floor of complete readout channel (Column S&H and ADCs). Therefore, in general, the readout circuitry must be designed with a temporal noise floor such that this contribution together with temporal noise associated to the pixel circuitry generate processed images with the accuracy required by the processing algorithms to be applied in next steps. This means that the dynamic range and signal-to-noise ratio of images resulting from analog processing and downloading process are enough in order to implement the following operations of processing algorithms.

In the particular case of Q-Eye system, take into account that the main objective of sensor is to download processed and compressed binary images containing the useful characteristics for a higher level of processing, the Analog I/O block has been implemented with debugging purposes. Then, the performances required by the readout channels in these conditions are less demanding than specifications for readout channels implemented in a Standard CMOS Image Sensors.

4.2.1.2. BINARIZATION BLOCK

The Binarization block is the circuitry per column responsible for loading and downloading of binary images. Figure 39 depicts its architecture. During a download process of binary images from the pixel array, the analogue signal from the processing cell is binarized in function to a threshold by a comparator and the result is taken by the shift register, see Figure 32. While the binarization operation is performing in a parallel way and the result stored in one shift register, the other shift register is reading through the image interface.

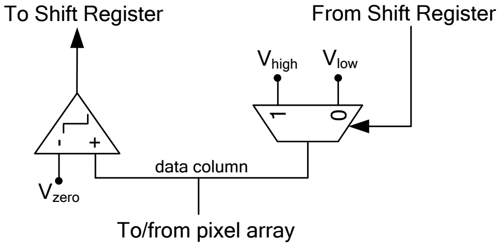


Figure 39. ARCHITECTURE OF COLUMN BINARIZATION BLOCK.

During binary image uploading, the data bit is converted to an analogue level by an analogue multiplexer and transmitted to the data column. The conversion to analogue domain is required in order to introduce the binary data into the local analogue data node (*ladn*) in the processing cell for further storage into a digital memory.

While the row stored in one shift registers is being written into the processing cell in parallel, the other shift register is loaded by the following row through the image interface.

4.2.1.3. AD/DA CONVERTERS BANK

Let us have a look again on Figure 32. It is seen that the Image Interface has an AD/DA converters bank composed by four AD converters to download four pixels of gray level image per external master clock cycle from the Column S&H bank, and four DA converters to load four pixels of gray level image per external master clock cycle into the Column S&H bank. Figure 40 shows the architecture of the AD/DA converters bank.

The control signals of Column S&H and the processing array are generated in basis to the internal clock *ck_row_col_io*. The conversion clocks are *ck_da* for the DA converters and *ck_ad* for the AD converters. In order to synchronize the conversion operation and writing/reading array process, the delay between ck_row_col_io and ck_da or ck_ad is programmable through the configuration registers of CGU (Clock Generation Unit), see section 4.1.3.2 The operation frequency of converters is the same as the external master clock CLK, nominally 50MHz.

In a load/download operation of a gray level image, four pixels are written or read from the S&H banks through the data bus buff<3:0> by either the DA converters (upload) or the AD converters (download). The AD/DA converters bank are configured by the user for a load or download process through the configuration registers associated to the Image Interface which generate the configuration signals for the converters bank.

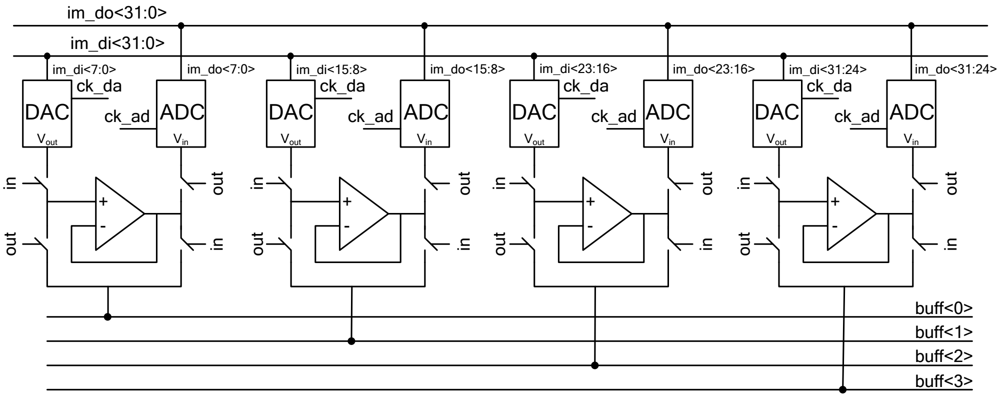


Figure 40. CONVERTERS BANK ARCHITECTURE.

4.2.1.3.1. AD converters

The ADCs in the Image Interface has a *flash architecture* [Gend91] [Plas94] [Raza95] – see Figure 41. These flash converters have a word length of 8 bits. Then, the input signal is fed to 256 comparators in parallel. Each comparator is also connected to a corresponding node of a resistor string which generates the 256 analogue references. Thus, an output code word expressed in thermometer code is generated. This thermometer code is converted to a one-hot code by the Bubble Corrector block and codified to 8-bits binary word by the output encoder. The **Bubble Corrector** block has two functions,

- one is the conversion from thermometer code to one-hot codification and
- the other is to remove the bubble errors which will occur near the transition point of the thermometer code.

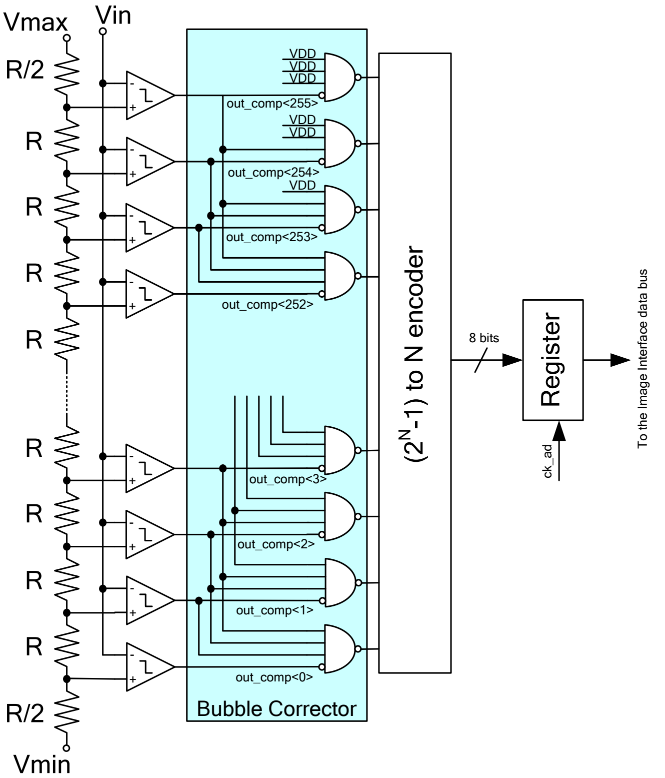


Figure 41. FLASH CONVERTER ARCHITECTURE.

The bubble error is provoked when sometimes a lone 1 occurs within the string of 0s (or a 0 within the string of 1s) due to comparator metastability, comparator offset, noise, cross talk, limited bandwidth, timing mismatch among comparators, etc. These bubbles occur near the transition point and can be removed with extra complexity using NAND gates with more than two inputs, [Gend91] [Stey93]. For instance, with the approach implemented in the Q-Eye Image Interface, which is shown in Figure 41 bubble errors that occur within four places of the transition point do not cause any large errors.

4.2.1.3.2. Description of main circuit errors

All non-idealities associated to the behavior of AD converter limit the image quality downloaded through the readout channel composed by the column S&H and ADC. Then, in the following points, several critical issues to take into account in the design phase of a flash converter are treated, [Raza95]:

- The **INPUT CAPACITANCE** of the converter: The large number of comparators connected to the input node results in a large parasitic load at node V_{in}. Such large capacitive load limits the speed of flash converter and usually requires a strong and power-hungry buffer to drive the input node V_{in}. In the case of Q-Eye system, the AD/DA converters bank has four Input/Output configurable buffers (one per converter) which drive the input of the corresponding AD converter, see Figure 40.

  While the total input capacitance of flash ADC limits the analogue input bandwidth considering the signal source output impedance, nonlinearities in this input capacitance introduce harmonic distortion in the sampled signal. This means that the input/output characteristic of converter presents a nonlinear behavior. For a large number of comparators, this nonlinearity becomes significant, and the low-pass filter formed by the signal source resistance and the ADC input capacitance yields an input signal-dependent delay.
- **CLOCK AND SIGNAL DISPERSION** (skew of the clock and input signal at different places on comparators string): The distributed nature of sampling, the analogue input signal and clock must travel long distances on a large ADC, implies that this signal and clock experience different delays due different loading. Furthermore, even with identical loading, the clock waveform (ideally square wave) is dispersed as its transitions are slowed down by the distributed resistance and capacitance of interconnects. Thus, the exact time difference between the analogue signal and the clock edge varies along the comparators string, causing non-linearity or harmonic distortion in the sampled waveform.
- **CLOCK JITTER:** All sampling circuits suffer from SNR degradation as the jitter of the sampling command increase. To arrive at a simple relation between maximum tolerable jitter and the ADC speed and resolution, it can be concluded that the jitter has negligible effect on the overall SNR if the analogue input varies by less than 1 LSB (Least Significant Bit) during jitter-induced time deviation of the sampling point.
- **LIMITED RISE AND/OR FALL TIME OF THE SAMPLING CLOCK:** Small rise or fall time of the sampling clock avoids additional jitter caused by (white) noise of the clock buffer circuits.
- **SUBSTRATE AND POWER SUPPLY NOISE:** On an integrated circuit having a clock signal in tens of MHz, it is difficult to keep power-supply noise below a few tenths of a volt. This power-supply noise can easily couple through the circuitry or substrate resulting errors. The effects of this noise can be minimized following several strategies. For instance, the clocks must be shielded from the substrate and from the analogue circuitry. Also, running differential clocks closely together will help prevent the signals being coupled into the substrate or through the air. Also, analogue power supplies should be separated from digital power including having analogue power to the comparator preamps while using digital power to the latch stages.
- Errors associated to the **GENERATION OF ANALOGUE REFERENCES** by the resistor string:
  - An important aspect of resistor string is the nonlinearity it introduces. This error results from mismatches in the resistors comprising the string, see section 4.2.1.3.4.

- In CMOS flash converters wherein comparators usually employ switched capacitors at their input, a transient or AC bowing affects to the resistor string. CMOS comparators often incorporate offset cancellation for resolutions of 8 bits and above, similar to the comparator topology used in the flash converters of the Q-Eye, depicted in Figure 42. It can be observed that the equivalent capacitance, resulting from the serial connection of sampling capacitance C_1 and input capacitance of inverter C_p, switches between input voltage and analogue reference during the operation of comparator corresponding to the sampling and comparison phases respectively. This is equivalent to a resistive path between the input voltage V_{in} and reference voltage, introducing an AC bowing along the string. This bowing generates integral nonlinearity in the input/output characteristic of converter.

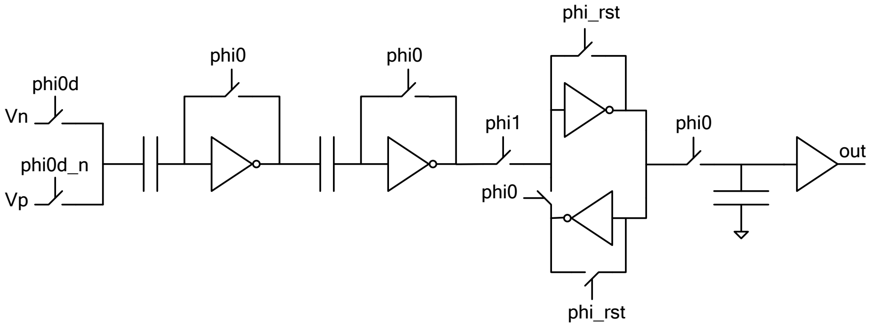


Figure 42. SCHEMATIC OF COMPARATOR IN THE ADC.

- Another error in the reference voltages is the transient error due to the capacitive feedthrough and charge injection towards the resistor nodes. This transient error depends on the temporal response of ladder. It can be reduced by lowering the unit resistance of string, but at the cost of higher power dissipation.
- The thermal noise associated to the resistors introduces random deviations in the reference voltages.

- **COMPARATOR ERRORS**, [Rodr02]: A comparator can work in continuous time or with sampled data. Figure 43 describes a general architecture of a sampled data comparator to which the comparator actually employed in the Q-Eye (Figure 42) belongs. It is the cascade connection of three blocks: a sample and hold, an amplifier and latch.

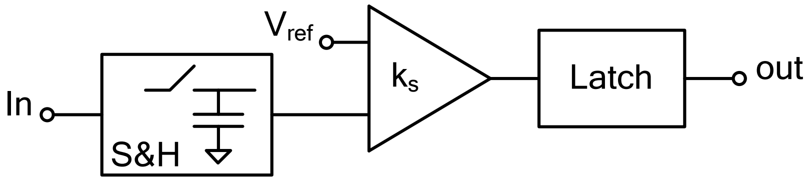


Figure 43. SAMPLED DATA AND LATCHED COMPARATOR.

- Below, the most important features of comparators are described.
  - **OFFSET:** It is the voltage or current that must be applied to the input to obtain the crossing point between low and high logic level. The lack of symmetry of comparator circuit results in an equivalent input offset. This offset sets a limitation on the minimum achievable comparator resolution or accuracy. Input signals whose amplitude is smaller than the input offset will not be properly detected by the comparator.

    The offset has two different components. Deterministic offset is due to asymmetries of the comparator circuit structure itself. On the other hand, random offset contemplates asymmetries caused by random deviations of transistor sizes and technological parameters, and it is observed in both symmetrical and asymmetrical circuit topologies. These deviations make nominally identical transistors become mismatched, the amount of mismatch being inversely proportional to the device area and directly proportional to the distance among them.

The offset parameter E_{OS} together with the static gain k_s associated to the amplifier defines the static resolution:

$$\xi_s \approx |E_{OS}| + \frac{E_{OH}}{k_s} \qquad \textbf{Eq. 322}$$

where $\pm E_{OH}$ are the levels which correspond to the logic one and zero.

- **SENSITIVITY**: It is the minimum input voltage or current that produces a consistent output signal within the expected comparison time.

Let us assume that the comparator is perfectly balanced for zero input signal *x=0*; this means that the output is in the quiescent point of comparator with *y=0*. Therefore, the input offset is ignored in this definition. Let us then consider that a step stimulus of value $x = \Delta_d$ is applied at *t=0*. By using small-signal models around the quiescent point, the output waveform can be calculated as:

$$y(t) = \Delta_d \cdot k_d(t) \qquad \textbf{Eq. 323}$$

where $k_d(t)$, called dynamic gain, depends upon the comparator structure characterized by high-level parameters, such us static gain, unitary time constant, etc. For each structure, $k_d(t)$ reflects the time needed to build a given amount of gain. Particularly, for a given Δ_d it is interesting to know the time needed to build the necessary gain so that the output reaches the restoring logic level E_{OH}. This time, called quiescent comparison time T_c, is calculated from:

$$E_{OH} = \Delta_d \cdot k(T_c) \qquad \textbf{Eq. 324}$$

This equation defines a resolution-speed tradeoff which is different for each comparator architecture. In general, the larger T_c (the slower the comparator) the smaller Δ_d (the more accurate the comparator).

- **METASTABILITY**: It occurs when the difference at the input of comparator is small, making the circuit take a long time to produce a well-defined logic output. If the instantaneous value of the input signal to a flash converter is close to the reference voltage of one of the comparators, that comparator will have an indeterminate output for a long time, possibly causing an erroneous digital output for a particular conversion. Note that when metastability occurs, the final logical value of comparator output is not critical, since the difference between the analog input and the converter reference is very small. However, the undefined output logic value produces substantial errors.

This time that takes a comparator latch to come from the latch mode to the track mode when a small input of the opposite polarity from the previous period is present can be minimized in several ways, by keeping the time constants of the internal nodes of the latch as small as possible or shorting the differential internal nodes together temporarily just after latch time. This last method has been used for the comparator of flash AD converter in the Q-Eye, depicted in Figure 42.

- **HYSTERESIS**: The comparison threshold for input signals changing from low to high can be different from the threshold for signals changing in the opposite direction. The difference between these two thresholds is defined as hysteresis. This effect can be a limit or a benefit depending on application. For instance, the comparator hysteresis is a limitation in data converters. Because it generates different output codes depending on the sign of the input signal derivate. In others applications, for example, for continuous time applications the crossing of a noisy input signal through the reference may produce many transitions of the output signal. A hysteresis larger than the noise level avoids the effect and results are beneficial.

  - **KICKBACK NOISE**: This phenomenon is present in latched comparators. When clocked comparators are switched from track to latch mode, or vice-versa, a charge glitch at the input of the latch is produced. If there is no preamplifier, this glitch will cause major errors due to the unmatched impedances at the comparator inputs.
  - **POWER SUPPLY REJECTION**: Spur signals affecting the power supply lines can modify the input sample and produce wrong outputs.
  - **TEMPORAL NOISE**: The electronic noise affects mainly during the sampling phase of comparator, because during the amplification phase the input equivalent noise is negligible due to the open loop gain of comparator.

- Errors associated to the Input/Output buffers in the Converter Bank (Figure 40):
  - Offset
  - Gain error
  - Distortion in the input amplifier
  - Settling error
  - Temporal noise

4.2.1.3.3. On the design of the comparator

The comparator is an essential building block in any ADC architecture. Therefore, in this section the comparator implemented for ADCs included in the image interface of Q-Eye system is described in detail.

Referring the comparator in Figure 42 it is composed by two pre-amplifier stages and a Discrete Time (DT) regenerative output stage (latch stage). The regenerative stage works in discrete time to avoid the hysteresis phenomenon. The regenerative stage is commonly built by cross-coupling a pair of inverters to form a latch. As a consequence of the circuit memory, an inertia appears. If the initial stable equilibrium point corresponds to a high output, it means that large enough negative (positive) values must be applied to counterbalance the circuit tendency to remain in the high state and, thereby, force its evolution toward the low state. The inertia is eliminated by employing switches to place the latch circuit at an unstable equilibrium point before comparison or regeneration phase. This operation, realized during the reset phase (see Figure 44), is equivalent to erasing the memory of the circuit and it is the key to guarantee that hysteresis will not appear.

The effect of kickback noise due to the latch is minimized by the two amplifier stage topology.

The amplification stages include offset compensation based in the auto-zero technique.

To build with a single stage the gain needed for comparison operation results in a disadvantageous resolution-speed tradeoff. To relax this tradeoff, multistep structures like the topology used in the comparators of Q-Eye AD converters are employed to achieve the gain through the multiplication of several gain factors ($g_m \cdot r_o$). Usually, the static resolution of multistep architectures is basically limited by the offset, and the influence of the static gain becomes negligible.

In general, for a multistep comparator with N stages, considering for illustration purpose all stages identical, ignoring the input offset of amplification stages and assuming that all capacitors are initially discharge, when an input step valued is applied at the initial instant *t=0*, following behavior is obtained applying Laplace-domain analysis:

$$Y(s) = \left(\frac{k_s}{1 + s \cdot \tau_o} \right)^N \cdot \frac{\Delta_d}{s} \qquad \textbf{Eq. 325}$$

Where a first-order model has been considered for the amplification stages. $k_s = g_m \cdot r_o$ is the static gain associated to the stage and $\tau_o = C_o \cdot r_o$ is the time constant.

Assuming $\Delta_d \cdot k_s^N >> E_{OH}$ results in $T_C << \tau_o$, and the following approximation can be developed:

$$Y(s) \cong \frac{\Delta_d}{\left(s^{N+1} \cdot \tau_u^N\right)} \Rightarrow y(t) \cong \Delta_d \cdot \frac{1}{N!} \cdot \left(\frac{t}{\tau_u}\right)^N \qquad \textbf{Eq. 326}$$

Where $\tau_u = {C_o}/{g_m}$ is the unitary time constant of amplifier.

The dynamic gain is expressed by the equation:

$$k_d(t) = \frac{1}{N!} \cdot \left(\frac{t}{\tau_u}\right)^N \qquad \textbf{Eq. 327}$$

And the resolution-speed tradeoff becomes:

$$\Delta_d \cdot \left(\frac{T_c}{\tau_u}\right)^N \approx N! \cdot E_{OH} \qquad \textbf{Eq. 328}$$

Given the resolution and conversion time, the design parameters for the amplifier stages can be obtained in basis to Eq. 328. The timing diagram corresponding to the control signals of clocked comparator are shown in Figure 44.

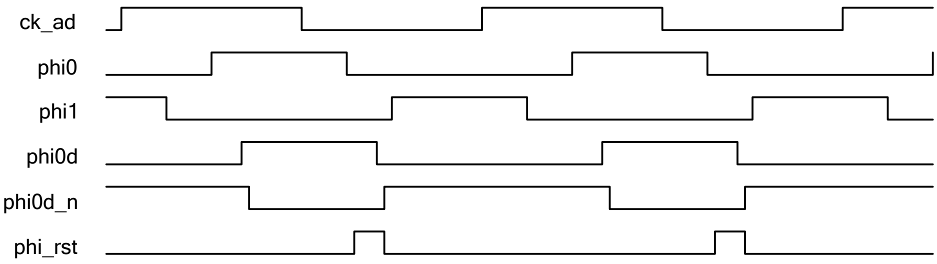


Figure 44. CONTROL SIGNALS OF ADC COMPARATOR.

The control signals are generated by a Clock Phase Generator. Two non-overlapped control signals or phases are obtained from an external clock *ck_ad*. The schematic is shown in Figure 45.

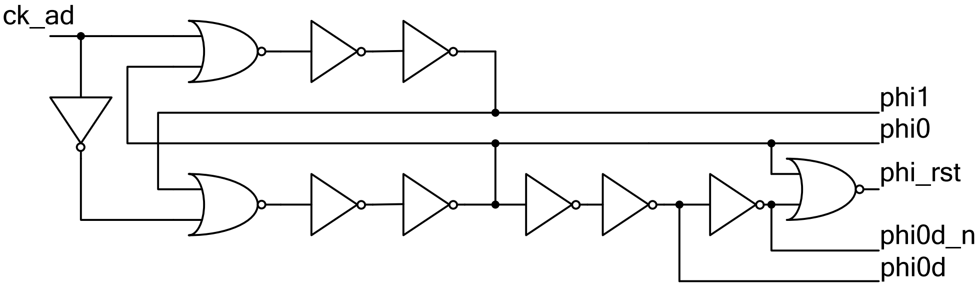


Figure 45. CLOCK PHASE GENERATOR.

In order to avoid timing errors along the comparators string, a careful distribution and buffering of clock and control signals is required taking into account the loading. In the case of Q-Eye converters, a tree distribution of clock *ck_ad* controlling the delay and slew has been implemented. In basis to this clock the control signals are generated locally for each comparator using a Clock Phase Generator.

4.2.1.3.4. DA converters

IMAGE INTERFACE DAC ARCHITECTURE. These DACs are decoder-based converters. The decoder-based approach for realizing a N-bit DA converter generates 2^N reference signals and pass the appropriate value to the output, depending on the digital input word. The digital input word is captured by the Q-Eye input register which is controlled by the internal *ck_da* clock. Regarding the references, they are generated by a resistor string and the analogue value is transmitted by a switch network connected in a tree-like decoder. In our case, 256 analogue levels are generated for 8-bit DACs – see Figure 46.

As Figure 40, included at the beginning of Section 4.2.1.3, shows, ADCs and DACs are grouped by pairs. This figure also shows that the output voltage of DA converter is driven towards the Column S&H in the Image Interface through the Input/Output configurable buffers depicted in Figure 40. Assuming the offset of these buffers does not depend on their input voltages, the DAC is inherently *monotonic* because any tap on the resistor string must have a voltage lower than that of its upper neighbor tap.

Also, the accuracy of this D/A depends on the matching of resistors R in the string. The mismatch characteristic determines the Integral Non-Linearity (INL) error and Differential Non-linearity (DNL) error of converter, parameters which characterize the non-linear behavior, in following paragraphs of this section the INL and DNL parameters associated to the string of resistors is obtained.

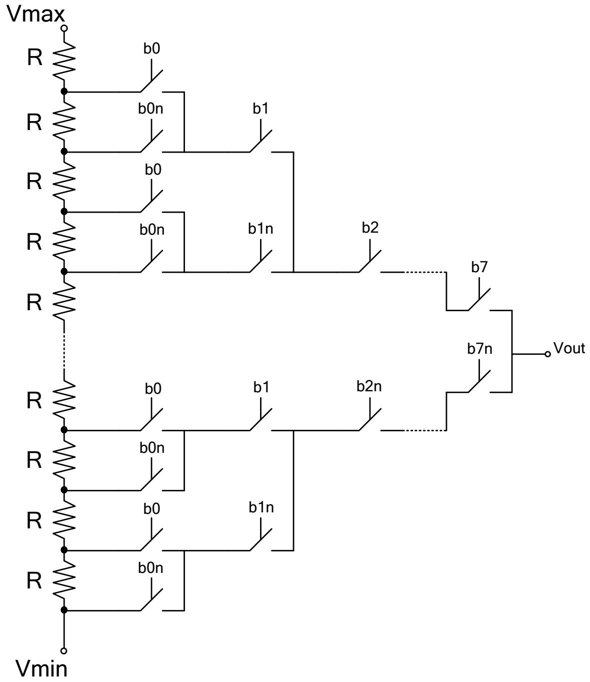


Figure 46. RESISTOR STRING CONVERTER ARCHITECTURE.

The level of random noise associated to the DA converter is given by the thermal noise generated by the resistors and the electronic noise (white and 1/f noises) introduced by the output buffer. This noise floor limits the Dynamic Range (DR) and Signal to Noise Ratio (SNR) of input image. The delay through the switch network is another source of errors in the characteristic of DA converter (introducing offset, gain error and non-linearity) and the main limitation on speed for the converter operation.

Regarding the performance of DA converters, it is important to consider the non-idealities of Input/Output buffers implemented for Q-Eye Image Interface. These buffers drive the input image to the Column S&H bank. Therefore, the non-idealities associated to the buffers affects to the input image. The main non-idealities are listed below:

- Temporal noise
- Offset

- Gain error
- Distortion (Non-linearity)
- Settling error
- Rejection to power supplies noise.

DESIGN CONSIDERATIONS FOR THE RESISTOR STRING. As mentioned above, ADCs and DACs are grouped into pairs (Figure 40); each pair consisting of one ADC and one DAC which share a single resistor string. Bear in mind that the interface includes four ADC-DAC pairs and hence four different physical strings. Because the strings do not carry signals but references, these four strings are interconnected in parallel to reduce the differences between converters and hence the Column Fixed Pattern Noise. Figure 47 illustrates this parallel connection.

Therefore, there is a resistor string common for each ADC and DAC to generate the reference levels required by the flash AD converter and decoder-based DA converter. The performances of AD and DA converters come determined by the resistor strings accuracy and characteristics.

The DACs and ADCs are designed to implement the Input/Output correspondence displayed in Figure 48. This is slightly different from the standard coding because the zero code disappears. The range of converters is defined by the analogue references V_{xmax} and V_{xmin}, being the range equal to:

$$V_{REF} = V_{\max} - V_{\min} \quad \textbf{Eq. 329}$$

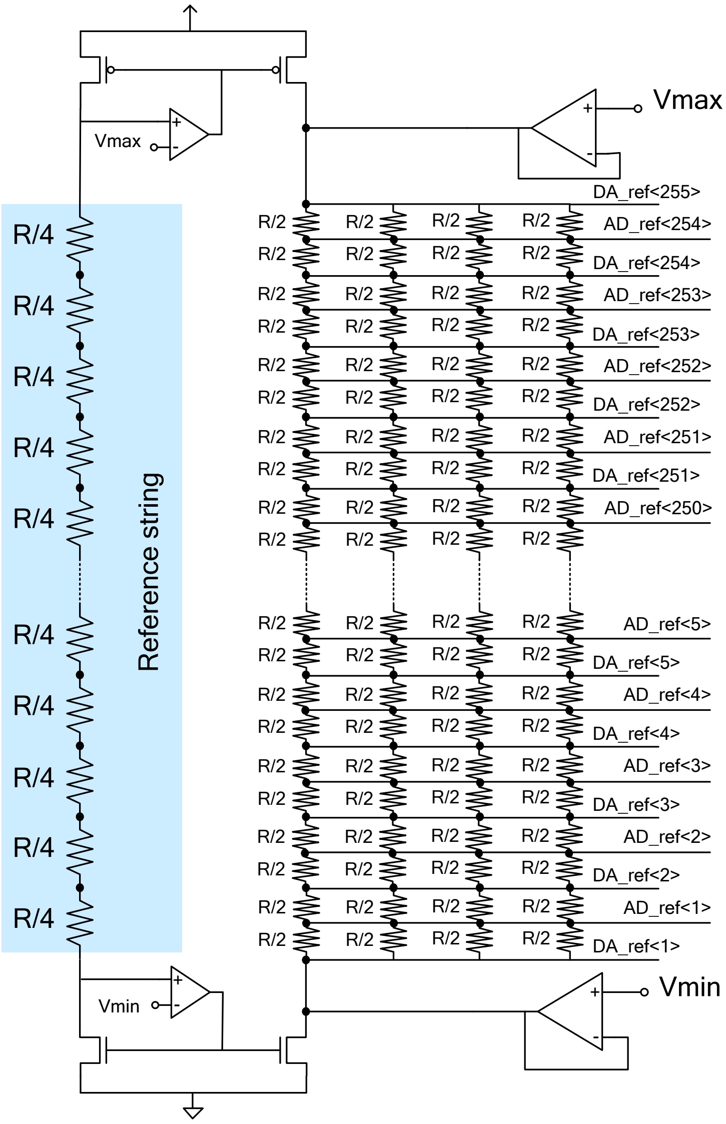


Figure 47. RESISTOR STRINGS FOR CONVERTERS.

Also, the zero level V_{zero} in the analogue processing domain corresponds to the decimal codification 128. Therefore, the following relation is obtained:

$$V_{zero} = \frac{V_{\max} - V_{\min}}{2} = \frac{V_{ref}}{2} \qquad \textbf{Eq. 330}$$

Note that Figure 47 includes a reference string, which has been introduced to avoid that the reference voltages V_{xmax} and V_{xmin} change depending on the real resistance value of implemented resistors chip to chip. The resistances in converters have been implemented with metal resistors to obtain the required speed for the conversion operation. In order to reach the conversion speed, the resistance value must be enough low to obtain an appropriate constant time for the resistance ladder, but this strategy go against the power consumption. There is a trade-off between speed and power consumption.

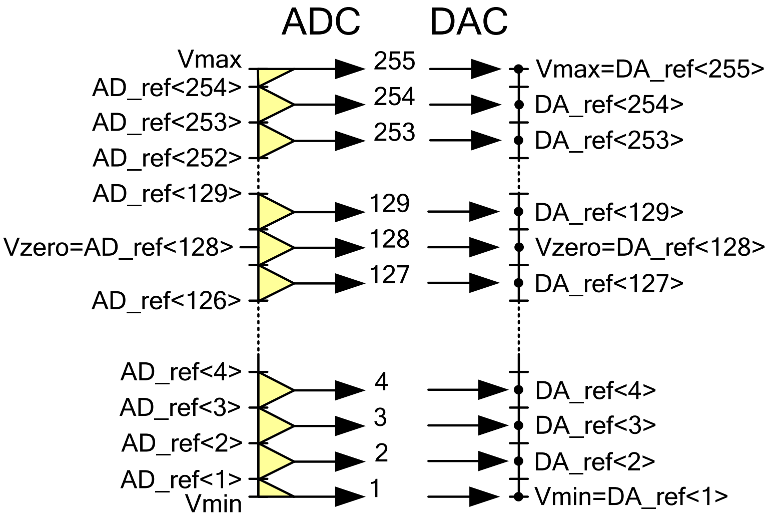


Figure 48. AD AND DA CONVERSION CORRESPONDENCE.

The resistance associated to the metal resistor change chip to chip significantly. Inside the chip, other parameter which affect appreciably to the resistance value is the temperature. Therefore, to avoid that the changes in the current flowing through the resistor string affect to the reference values in the extreme of strings due to the finite output resistance associated to the reference buffers, the Reference string has been introduced to generate the current demanded by the resistors strings.

This error in the range of converters (Vmax-Vmin) implies a gain error in the conversion process. The amplifiers which drive the analogue references include an automatic offset calibration mechanism similar to the offset calibration algorithm of Programmable Buffers, see section 4.1.3.1, in order to minimize the error provoked by the changes in the converter range.

The errors due to the non ideal behavior of resistors in the string provoke errors directly in the reference levels of converters and therefore in the characteristic and accuracy of conversions (analogue to digital and digital to analogue).

The resistor mismatch introduces differential and integral non-linearity (DNL and INL respectively) [Raza95]. This mismatch has two contributions: a systematic component due to gradients of several technological parameters, for instance, variation in doping, geometrical or physical parameters, etc. And a random component originated primarily from uncertainties in geometry definition during physical integration, as well as random variations in contact resistance, etc.

Considering a simple case where the string exhibits a linear gradient as it is described in Figure 49 . The voltage at the jth tap of this ladder is:

$$V_j = \frac{j\cdot R + \frac{j\cdot(j-1)}{2}\Delta R}{N\cdot R + \frac{N\cdot(N-1)}{2}\Delta R}\cdot(V_{\max}-V_{\min}) = \frac{j\cdot R + \frac{j\cdot(j-1)}{2}\Delta R}{N\cdot R + \frac{N\cdot(N-1)}{2}\Delta R}\cdot V_{ref} \qquad \textbf{Eq. 331}$$

The INL parameter is given by the difference between the real voltage value and the ideal tap voltage:

$$INL_j = \frac{j}{N}\cdot V_{ref} - \frac{j\cdot R + \frac{j\cdot(j-1)}{2}\Delta R}{N\cdot R + \frac{N\cdot(N-1)}{2}\Delta R}\cdot V_{ref} = \frac{j\cdot(N-j)}{R + \frac{N-1}{2}\Delta R}\cdot\frac{\Delta R}{2N}\cdot V_{ref} \qquad \textbf{Eq. 332}$$

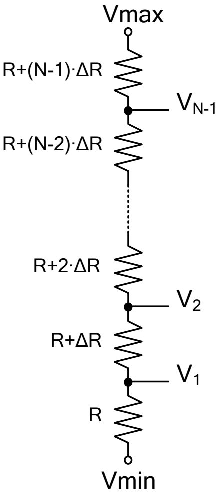


Figure 49. LINEAR GRADIENT IN RESISTOR STRING.

Assuming $R >> (N-1)\cdot \Delta R/2$, the INL$_j$ parameter reaches a maximum at $j = N/2$:

$$INL_{\max} = \frac{N}{8}\cdot\frac{\Delta R}{R}\cdot V_{ref} \qquad \textbf{Eq. 333}$$

The DNL parameter associated to the ladder is obtained by calculating the deviation of voltage step between consecutive taps $(V_{j+1} - V_j)$ from the ideal value of 1 LSB:

$$DNL_j = V_{j+1} - V_j - \frac{V_{ref}}{N} \approx \left(j - \frac{N-1}{2}\right)\cdot\frac{\Delta R}{R}\cdot\frac{V_{ref}}{N} \qquad \textbf{Eq. 334}$$

Where it is assumed $R >> (N-1)\cdot \Delta R/2$. For large N, this parameter reaches a maximum at $j=1$ and $j = N-1$:

$$DNL_{\max} = \frac{1}{2}\cdot\frac{\Delta R}{R}\cdot V_{ref} \qquad \textbf{Eq. 335}$$

To analyze the nonlinearity resulting from random resistor mismatch a Probability Density Function (PDF) for the value of each resistor of string is considered. Since random mismatch arises from a significant number of technological parameters which can be considered uncorrelated events, a Gaussian PDF for the resistance values in the string can be assumed, characterized by a mean value R and a standard deviation σR.

Under these considerations, the tap voltages of resistor ladder follow a Gaussian distribution too, [Kubo82], with a mean equal to:

$$V_j = \frac{j}{N} \cdot V_{ref} \qquad \textbf{Eq. 336}$$

And a standard deviation expressed by the equation:

$$\sigma V_j = \sqrt{\frac{j}{N^2} \cdot \left(1 - \frac{j}{N}\right)} \cdot \frac{\sigma R}{R} \cdot V_{ref} \qquad \textbf{Eq. 337}$$

Therefore, the INL parameter is given by the standard deviation:

$$INL_j = \sqrt{\frac{j}{N^2} \cdot \left(1 - \frac{j}{N}\right)} \cdot \frac{\sigma R}{R} \cdot V_{ref} \qquad \textbf{Eq. 338}$$

The integral non-linearity reaches a maximum at $j = N/2$:

$$INL_{\max} = \frac{1}{\sqrt{4 \cdot N}} \cdot \frac{\sigma R}{R} \cdot V_{ref} \qquad \textbf{Eq. 339}$$

Developing a similar probabilistic study, the PDF associated to the voltage between consecutive taps $\left(V_{j+1} - V_j\right)$ can be obtained. In this case, the resulting mean value associated to the PFD is determined by the equation:

$$\left(V_{j+1} - V_j\right) = \frac{V_{ref}}{N} \qquad \textbf{Eq. 340}$$

And the standard deviation:

$$\sigma\left(V_{j+1} - V_j\right) = \sqrt{\frac{N-1}{N} \cdot \frac{\sigma R}{R} \cdot \frac{V_{ref}}{N}} \qquad \textbf{Eq. 341}$$

Being the differential non-linearity parameter described by:

$$DNL = \sqrt{\frac{N-1}{N} \cdot \frac{\sigma R}{R} \cdot \frac{V_{ref}}{N}} \qquad \textbf{Eq. 342}$$

When the number of resistors is large $\left(N \gg\right)$, the DNL figure is approximated to:

$$DNL \approx \frac{\sigma R}{R} \cdot \frac{V_{ref}}{N} \qquad \textbf{Eq. 343}$$

Resistor strings with a large number of segments yield lower absolute non-linearity than those with a small number of segments. The reason is because random error in the value of resistors tends to average out when many segments are connected in series.

4.3. ADDRESS-EVENT READ-OUT

4.3.1. ASYNCHRONOUS DOWNLOADING OF IMAGES. ADDRESS-EVENT READING

Besides the load/download processes of binary and gray level images the Q-Eye system includes a special protocol to download the coordinates of active pixels in a sparse binary image, called Address Event protocol. The download of coordinates using the Address Event protocol is performed through the Programming Interface instead of using the Image Interface.

Given a binary image, an active pixel is defined as that pixel with a "1" binary value associated (white pixel). Consequently, an inactive pixel is that pixel with a "0" binary value associated (black pixel). The binary data associated to each pixel is stored in one of the digital memories of pixel or processing cell.

The processing cell or Q-Eye pixel includes a specific block, the Active Pixel I/O block, to carry out the Address Event download process, see section 3.1.1 in Chapter 3. The sparse binary image which will be process by the Address Event circuitry must be stored in a specific digital memory of processing cell, specifically in the mask memory (see section 3.2.2.2 in Chapter 3).

The binary data stored in the mask memory defines if the corresponding pixel is active or not, configuring the Active Pixel I/O block, see Figure 50.

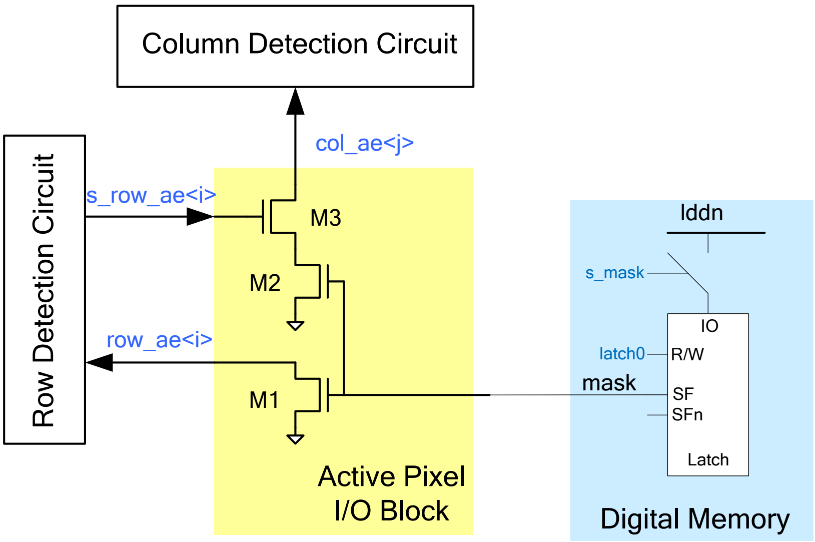


Figure 50. ACTIVE PIXEL DETECTION IN PROCESSING CELL.

The *row_ae* line is common to all pixels belonging to the same row. And the *col_ae* line is the same for all pixels of a determined column. Therefore, when one or more pixels in a row have stored an active value "1" in the mask memory the *row_ae* is set to zero, because the transistor M1 is turned on. In a similar way, when at least one pixel is active in a determined column the transistor M2 is turned on and the *col_ae* line will be set to zero activating the control signal *s_row_ae*. In Figure 50, the index i represents the number of row and index j the number of column.

An active row is defined as a row containing at least one active pixel. Similarly an active column is defined.

When an Address Event download process is launched, the digital Row Detection Circuit looks for the first row with active pixels detecting which *row_ae* line is set to zero in an asynchronous manner with combinational logic. Once detected the active row, the Address Event control determines the coordinate of row, sends this coordinate through the Programming Interface and launches the search of active columns setting to one the corresponding *s_row_ae* line. The Column Detection Circuit looks for the active columns or pixels for the selected row in a sequential and asynchronous way, detecting the columns with the *col_ae* line set to zero. When the Column Detection Circuit detects an active column, the Address Event control determines the column coordinate, sends it through the Programming Interface and orders to the Column Detection Circuit to carry on with the search.

Once the Column Detection Circuit has finished with one row, the Address Event control goes on with the search of following active rows through the Row Detection Circuit.

The Address Event download process is launched reading the Programming Interface Address ADD_PB<11:0>=*h'5F8*, see Table 5. The Address Event process will be interrupted, if this special address is not read by the Programming Interface.

ADD_PB<11:0>	RW_PB	Description
h’5F8	1	Launching of Address Event download process.

Table 5. Address Event download instruction.

The resulting coordinates of applying the Address Event algorithm can be read through the Programming Interface data bus DAT_PB<9:0>. The data bits DAT_PB<9:8> codify if the data DAT_PB<7:0> is a row coordinate or a column coordinate. Moreover, there are two special codes of data bus DAT_PB<9:0>, one of them indicates if the corresponding clock cycle is a processing cycle for the Address Event control machine, the another code informs when the Address Event process has finished, see Table 6.

DAT_PB<9:8>	DAT_PB<7:0>	Description
b’00	d’i	The decimal number i corresponds to a row coordinate.
b’01	d’j	The decimal number j corresponds to a column coordinate.
b’10	h’00	This code indicates a processing cycle for the Address Event control machine.
b’11	h’FF	This code indicates the end of Address Event process.

Table 6. Address Event information in data bus DAT_PB.

Both Row Detection Circuit and Column Detection Circuit require time to detect an active row and column respectively. Then, the Address Event control machine generates a valid coordinates every determine number of cycles. This number or time interval can be configured by the user through a configuration register associated to the Address Event process, its default value is four cycles. These clock cycles, during which the control machine is calculating the coordinate value, are defined as processing cycles by the Programming Interface.

There is a special operation mode of Address Event download process denoted by Calculation of Active Region. When this operation mode is activated through the configuration register of Address Event process, the Address Event Circuit determines and gives by the Programming Interface the coordinates of pixels which limit the active region containing all active pixels. The Address Event, configured in this mode, download first the coordinates of upper left corner of region and then the coordinates of lower right corner. This operation downloads only the first and last coordinates of a standard Address Event download process.

4.3.2. RANDOM ACCESS TO LOAD DATA IN ARRAY. ADDRESS-EVENT WRITING

Inversely, the user can launch a load process where is developed a random writing access into the pixels of processing array. In this writing operation, the pixel corresponding to the coordinates defined by the Programming Interface data bus DAT_PB is connected to the data column d_col<j>, where the Address Event control has set the binary data to introduce in the local node *ladn* of pixel or processing cell, see Figure 51. This binary data is defined by the analogue references V_{max} (associated to “1”) and V_{min} (associated to “0”).

The user defines if the value to introduce is V_{max} or V_{min} with the configuration register of Address Event process.

The Address Event load process is launch through the Programming Interface, writing the Programming Interface address ADD_PB<11:0>=h’5F0, see Table 7. The Address Event load process will be interrupted, if this special address is not written by the Programming Interface.

Once the Address Event load process has been launched, the coordinates of pixel which will be written will be defined by the Programming Interface data bus DAT_PB<9:0>, see Table 8.

ADD_PB<11:0>	RW_PB	Description
h'5F0	0	Launching of Address Event load process.

Table 7. Address Event load instruction.

DAT_PB<9:8>	DAT_PB<7:0>	Description
b'00	d'i	Select the row<i> to be accessed.
b'01	d'j	Select the column<j> to be accessed.
b'10	h'00	Deselect all rows and columns.
b'11	h'FF	Select all rows and columns.

Table 8. Address Event coordinates.

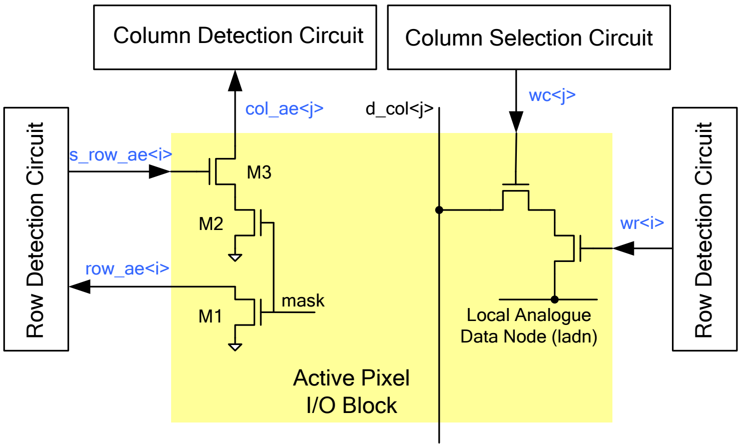


Figure 51. Active Pixel I/O Block architecture.

In order to define the access coordinates, first the row coordinate is indicated and then the column coordinate is defined in consecutive master clock cycles (CLK). Defining the row and column in the processing array, the corresponding *local analogue data node* (*ladn*) is connected to the column data node which is set to the analogue reference defined by the configuration register of Address Event. In the pixel, any analogue memory or digital memory can be connected to the *ladn* in writing configuration to store the data.

4.3.3. Global Mean of Images

Using the special instruction of Address Event load process DAT_PB<9:0>=h'3FF, which selects all rows and columns, an algorithm to calculate the global mean of a determined image can be implemented in the Q-Eye system.

In order to calculate the mean value of an image stored in an analogue or digital memory. Locally in every pixel or processing cell, the corresponding local memory must be connected in reading configuration to the *ladn* node. Once the *ladn* nodes have been set to the voltage which represents the pixel value, these voltages remain stored in the parasitic capacitance associated to the local node *ladn* when the memory is disconnected from it.

All these local nodes will be shorted in a global node to obtain the mean voltage. The shorting is implemented by the Address Event load instruction DAT_PB<9:0>=h'3FF, with this instruction the local nodes corresponding to the same column are shorted. Once this instruction has been applied, all rows and columns remain selected (wc<j>=1 and wr<i>=1) without the necessity of applying the instruction. This permits accessing to the configuration registers or memory instruction through the Programming Interface. This configuration in the array remains until the Address Event load instruction DAT_PB<9:0>=h'200 is applied. In this last case, all rows and columns will be deselected.

After shorting all pixels corresponding to each column, the user can short all columns of the array to obtain the mean voltage in a global node activating through the Programming Interface a specific bit of Focal Plane Processor Instruction.

Once the mean voltage has been obtained in the global node resulting from shorting all columns, this global node can be connected at the input of Miscellaneous ADC and launch the corresponding conversion and read the result using the configuration registers of Miscellaneous ADC with the Programming Interface. The resulting value can be used by the user or external host to act on the Q-Eye operation. For instance, the external host can measure the mean level of images for a determined exposure time using this algorithm and modified several parameters of Q-Eye for the following expositions.

4.4. SUPPLY AND BIASING IN THE Q-EYE

The Q-Eye employs analogue techniques for image pre-processing and is hence constrained by accuracy of data representations. However, an 8-bit equivalent resolution is enough for image processing in many real applications, as established by widely accepted video coding standards [CCIR90]. Moreover, early vision tasks realized by biologically inspired artificial vision chips can be accomplished with an equivalent resolution of 6-7 bits [Koch95].

Therefore, since the resolution required for processing and conversions in this system is moderate (8 bits), the power supply has been designed common for both domains digital and analogue, in order to simplify hardware implementation and reduce production costs. The complete system has been designed in a 1.8V Standard CMOS Technology. At system level, there are important issues to be considered in the implementation of mixed-signal Integrated Circuits:

- Power supply and packaging
- Substrate noise
- Reference distribution

covered in the following sections.

4.4.1. POWER SUPPLY AND PACKAGING

Power consumption is one of the main characteristics of IC to be into account. Two main contributions are defined, one associated to the consumption of analogue circuits and another corresponding to the digital circuits.

$$P_{total} = P_{ana\log} + P_{digital} \qquad \textbf{Eq. 344}$$

In relation to the digital power consumption, there are three primary components of power dissipation in digital CMOS circuits:

$$P_{digital} = P_{dynamic} + P_{sc} + P_{leakage} \qquad \textbf{Eq. 345}$$

The dynamic power $P_{dynamic}$ accounts for the energy dissipated in charging and discharging the nodal capacitances:

$$P_{dynamic} = C \cdot V_{DD}^2 \cdot f_{switch} \qquad \textbf{Eq. 346}$$

C is the capacitor which is charged and discharged, V_{DD} is the digital power supply and f_{switch} is the average frequency of charge/discharge cycle.

The short-circuit power P_{sc} is produced by short-circuit current which flows in a static CMOS gate when a conductive path exits from the power rail to the ground rail. This path exits when a signal at one of the gate inputs transitions, passing through intermediate levels [Hira96]. Within a determined input voltage range, both of the pull-up and pull-down networks conduct current. This current has a maximum value during the transition called I_{peak}. The period of time when this conductive path exits is denoted as t_{base}. A rough expression to estimate the short-circuit power is given by:

$$P_{sc} = \frac{1}{2} \cdot I_{peak} \cdot t_{base} \cdot V_{DD} \cdot f_{switch} \qquad \textbf{Eq. 347}$$

An analytical expression that characterizes with more accuracy the P_{sc} parameter, [Adle97], and exhibits 15% accuracy for a wide variety of RC loads is given by:

$$P_{sc} = \left| \ln\left(\frac{V_{Tn}}{V_{DD} + V_{Tp}} \right) \right| \cdot \left(R_{tr} \cdot C + R \cdot C \right) \cdot I_{peak} \cdot V_{DD} \cdot f_{switch} \qquad \textbf{Eq. 348}$$

This expression is based on the Sakurai alpha-power law model [Saku90]. The parameter R_{tr} is alpha-power model effective transistor resistance in the linear region, V_{Tn} and V_{Tp} are the threshold voltages of NMOS and PMOS transistors respectively, and f_{switch} is frequency for transitions.

The leakage current $I_{leakage}$ in a transistor is the current that flows between the power rails in the absence of switching, giving rise to a leakage power $P_{leakage}$:

$$P_{leakage} = I_{leakage} \cdot V_{DD} \qquad \textbf{Eq. 349}$$

Typically, the dynamic power is the dominant power component, contributing with 70-90% or more of the total power dissipation. Consequently, the most effective strategy for reducing the total power consumption is to reduce the dynamic dissipation.

The power consumption of analogue circuits is determined by the total bias current I_{total} of all analogue building blocks:

$$P_{ana\log} = I_{total} \cdot V_{DDA} \qquad \textbf{Eq. 350}$$

Being V_{DDA} the voltage supply associated to the analogue domain.

The total power consumption defines in the pad ring the number of pads dedicated to the power supply and ground supply during the design phase of top level floorplan. Depending on whether the power and ground supplies of digital and analogue domains are common or not, the pad ring will content independent analogue supplies and digital supplies sections. Usually in mixed-signal systems with a high accuracy specification, a separate power and ground supplies are implemented for analogue and digital circuits to avoid corrupting the analogue section by the large transient noise produce by the digital section. In the case of Q-Eye, the digital and analogue domains have common supplies.

Also the placement of blocks in the floorplan and the power consumption map determines the power pads distribution.

The design of pad ring in the Integrated System is closely related with the package definition.

The parasitics associated with the package and connections to the chip introduce many difficulties in the evaluation of the performance associated to the circuits at high speeds and high accuracies.

The die is mounted in the center of package cavity and bonded from the chip pads to the pads on the perimeter of the cavity. These pads are the tip of each trace that ends in each package pin. Such structure exhibits parasitic which are listed below:

- Bond wire self-inductance.
- Bond wire mutual inductance.
- Trace self-inductance.

- Trace to ground capacitance.
- Trace-to-trace mutual inductance.
- Trace-to-trace capacitance.

The package parasitics must be taken into account in the design of Integrated Circuits, preferably from the very beginning. Thus, simulations must include a circuit model of package, Figure 52, and the design and layout must take many precautions to minimize the effect of these package parasitics.

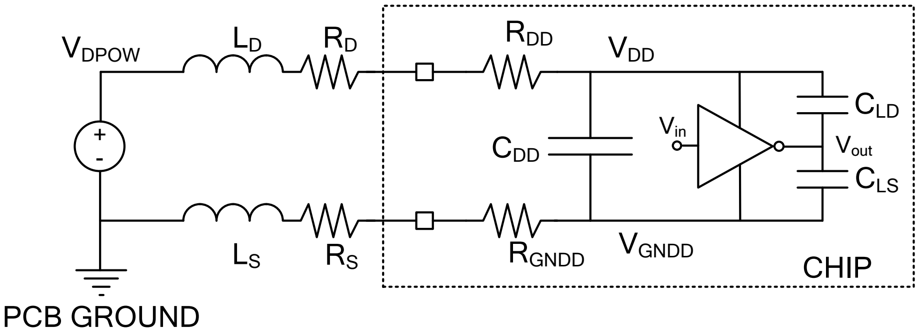


Figure 52. CIRCUIT MODEL OF PACKAGE.

Each bond wire and its corresponding package trace have a finite self-inductance depending on the wire length and type of package (2nH-20nH). The switching of digital circuits demands a dynamic current which is drawn from the voltage supply V_{DD} and V_{GNDD} (particularly from digital supplies) producing a voltage drop across L_D and L_G, see Figure 53, determined by the equation:

$$V_{DD}(t) = V_{DPOW} - L_D \frac{dI(t)}{dt} - (R_D + R_{DD}) \cdot I(t) \qquad \textbf{Eq. 351}$$

$$I(t) = L_G \frac{dV_{DD}(t)}{dt} \qquad \textbf{Eq. 352}$$

The inductance of bond wire and capacitances associated to the wire and die together with the metal resistances constitutes a LC serial resonator. Thus, the behavior of voltage supplies can be modeled, in first approximation, by a second order circuit.

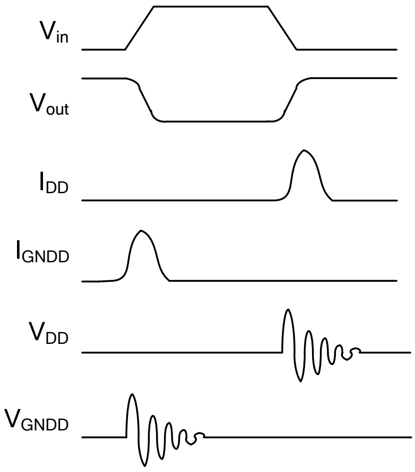


Figure 53. CMOS INVERTER DRIVING A LOAD CAPACITANCE.

This effect is called supply and ground bounce or noise. In mixed-signal systems, the effect of this transient noise over the analogue circuit performances must be studied. For high accuracy applications, mixed-signal systems employ separate supplies for the analogue and digital domains.

Separating power rails into analogue and digital sections is not always straightforward. In general, three types of functional domains can be defined in mixed-signal systems: A pure digital domain which contains the control, digital processing blocks and communication interfaces, a mixed digital domain which transmits and apply the corresponding control and configuration signals over the analogue circuits and finally the analogue domain which contains the analogue processing blocks.

Taking into account that the digital circuitry belonging to the mixed domain generates control signals which are connected to the gate of switch transistors in the analogue circuits. Should the mixed digital domain be supplied from analogue or digital power line? Consider an inverter of mixed digital domain which transmits a control signal to a switch transistor included in a sampling circuit, see Figure 54.

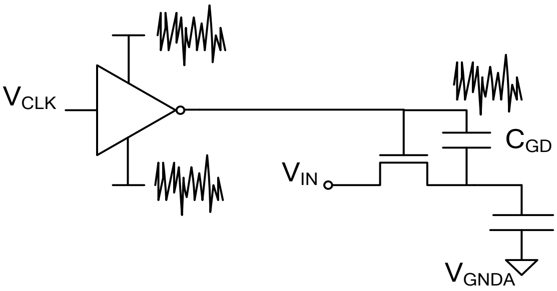


Figure 54. NOISE COUPLING BETWEEN DIGITAL AND ANALOGUE DOMAINS.

If the inverter is connected to the digital supply, then noise couples through the gate-drain overlap capacitance of switch transistor, corrupting the sampling data when the transistor is off. On the other hand, if many of this type of digital circuits are supply from the analogue supply, they draw large transient currents collectively corrupting the supply voltage. Therefore, a third type of power line may be required, so that it remains less noisy than the digital supplies.

In order to reduce the power and ground bounce, multiple pads bond wire and package pins can be used reducing the equivalent inductance.

Other strategies must be considered when high transient currents drawn from the supplies make it difficult to maintain a small bounce on the power and ground supplies individually. In these cases, a large on-chip capacitor is used to stabilize the difference between power and ground supplies (V_{DD},V_{GNDD}) or (V_{DDA},V_{GNDA}) . If the coupling capacitor is enough large the voltage supplies bounce in unison. The coupling capacitor must be chosen carefully to avoid the resonance effect with the package inductance at the operating frequency of system, phenomenon that amplifies the supplies noise. For this reason, some resistance is added in series with the coupling capacitor (usually, a MOS capacitor is sized such that its channel resistance dampens the resonance). It is important to consider that using a large number of MOSFETs in parallel to implement the coupling capacitor could affect to the yield of system.

While dedicating separate power lines to analogue and digital sections reduces the noise on the analogue supply, due to the mutual inductance of bond wires and package trace some noise may still to couple to sensitive analogue signals or analogue supplies. Both are susceptible to noise or transitions on digital supplies, clock signals or output buffers.

With an arbitrary pad configuration, even differential signaling cannot eliminate this effect because the noisy lines may not surround the sensitive lines symmetrically. Therefore, the design of pad ring plays a critical role in the performance that can be achieved.

The power consumption of digital and analogue circuits, the bounce noise in the power supplies due to the self-inductance, coupling effects due to the mutual inductance, etc define the number and distribution of supply pads in the floor-plan.

4.4.2. SUBSTRATE NOISE

Most modern CMOS technologies use a heavily-doped p^+ substrate to minimize latch-up susceptibility. However, the low resistivity of the substrate (on the order of 0.1 Ω·cm) generates unwanted paths among devices in the mixed-signal circuits. Therefore, taking into account that digital circuits introduce noise in the chip substrate by coupling and transient current demanded to voltages supplies; this noise, called substrate coupling or substrate noise, affects to the behavior and performance of analogue circuits. This effect has become an important issue in current mixed-signal ICs.

Figure 55 represents a simple example which explains this phenomenon. This figure shows that large voltage excursions at the output of inverter are coupled to the substrate through the drain junction capacitance, disturbing the substrate voltage because the finite impedance of L_D. In this case, the noise of substrate affects to the analogue buffer through the body effect, varying the threshold voltage of driver transistor with the substrate voltage. Therefore, the clock transitions disturb the analogue output of buffer.

In case that substrate is connected to the digital ground, the transient current demanded by the digital circuit to the ground supply provokes a bounce in the ground voltage which introduces noise in the substrate affecting to the analogue circuit too.

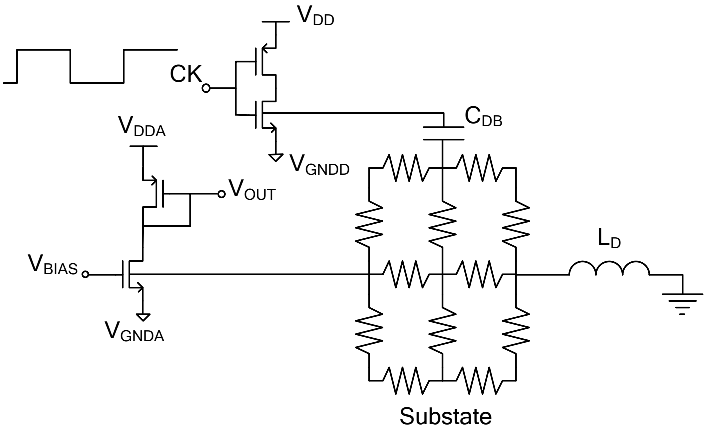


Figure 55. **EFFECT OF SUBSTRATE COUPLING.**

With large transient currents injected by the devices into the substrate, a low impedance connection is necessary to minimize the substrate bounce. Some modern packages contain a metal ground plane to which the die can be attached by conductive epoxy. The plane ends in several package pins that are tied to the board ground, avoiding bond wires and long, narrow traces in the substrate connection, such packages substantially reduce the substrate noise with no additional assembly cost.

Depending on doped level of substrate, techniques to isolate the sensitive analogue circuits from the substrate noise such as using guard rings may be ineffective. For resistive or lightly doped substrates the guard ring are effective. But, in the case of conductive or high doped substrates, the substrate operates as a low-resistance plane distributing a relatively uniform potential across the chip regardless of the position of noise generators.

In order to minimize the effect of substrate noise over the analogue circuits, several strategies can be applied:

- Differential operation should be used in circuits, making the analogue domain less sensitive to common-mode noise.
- Digital signals clocks should be distributed in complementary form, thereby reducing the net amount of the couple noise.
- Analogue operations, like sampling or charge transference between capacitances, should be performed after clock transitions such that the substrate voltage settles.
- OTAs using a PMOS differential input are preferred for signal handling.
- Use NMOS transistors for implementation of DC current sources.
- Use shielding for non-substrate related signals.
- Connect the analogue ground VSSA to the dirty substrate: In large mixed-signal systems, it may not be possible to avoid substrate bounce with respect to the external ground because of the high transient currents drawn by the devices and the finite impedance of the bond wire connected to the substrate. If the analogue ground and

substrate are connected, then the NMOS transistors experience no noise because their analogue ground voltage and substrate potential vary in unison. See Figure 56.

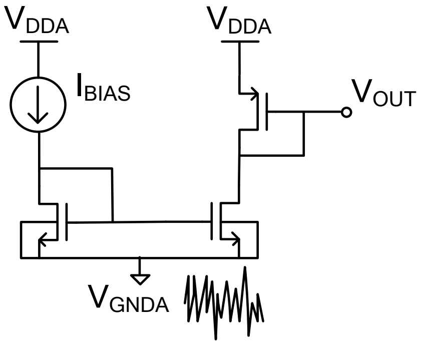


Figure 56. **SHORTING ANALOGUE GROUND AND SUBSTRATE.**

In the case of Q-Eye system, the analogue and the digital ground have been connected to the substrate owing to the moderate accuracy required (8 bits). Also, guard rings have been useful to isolate the sensitive analogue sections from the substrate noise. A guard ring may be simply a continuous ring made of substrates contacts that surround the circuit, see Figure 57. This structure represents a low impedance path to ground for the charge carriers produced in the substrate.

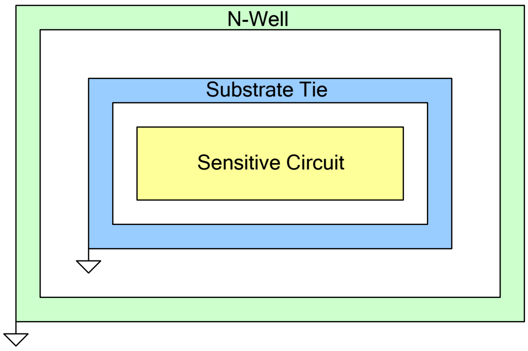


Figure 57. **GUARD RING STRUCTURE.**

CIS Technologies (Pinned Photodiode Technology) have a heavily doped substrate. In these cases, the analogue ground is connected to the substrate because the pixel data is referred to this substrate. Usually, in these highly doped technologies, there is an interesting and useful structure or layer denominated Deep N-Well, which is described in Figure 58.

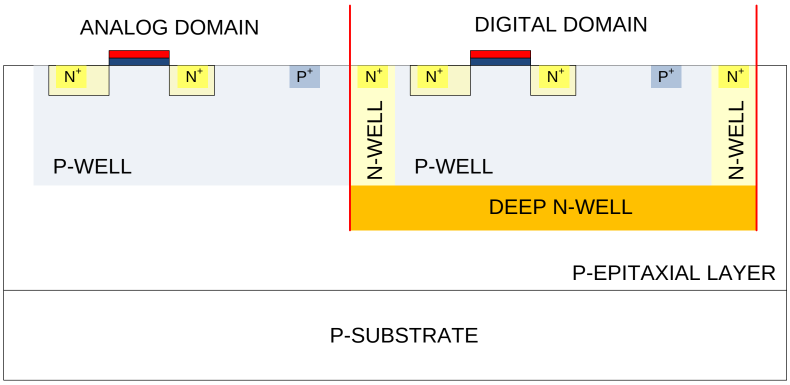


Figure 58. **DEEP N-WELL STRUCTURE.**

This structure allows implement an isolated p-substrate. For instance, in the case of Smart Image Sensors using the Pinned Photodiode Technology, a Deep N-Well structure is used to minimize the coupling noise from the digital domain to the analogue sensor domain. The digital circuitry is isolated including this one in a Deep N-Well structure.

4.4.3. POWER AND REFERENCE DISTRIBUTION

4.4.3.1. POWER SUPPLIES DISTRIBUTION

In Section 4.4.1 the power consumptions associated to each domain, digital and analogue, have been introduced. Directly related with this current demand, there is an important issue, the power supplies distribution along the complete system inside the chip, apart from packaging effects. There are several phenomena which must be taken into account during the design phase of power supply distribution.

ELECTROMIGRATION. It is a slow wear phenomenon caused by extremely high current densities. The impact of moving carriers with stationary metal atoms causes a gradual displacement of the metal. In aluminum, electromigration only becomes concern when current densities approach $5{\cdot}10^5$ A/cm^2. Although this may seem a tremendous current density, a minimum-width lead in a submicron process can experience electromigration at currents of only a few milliamps.

VOLTAGE DROP DUE TO THE RESISTIVE PATHS. The voltage supplies are distributed along the system by metal leads. These metal leads have associated a characteristic resistivity ρ. Thus, a rectangular metal lead characterized by a length L, width W and thickness t has a resistance R given by the expression:

$$R=\rho\cdot\frac{1}{t}\cdot\frac{L}{W}=R_s\cdot\frac{L}{W} \qquad \textbf{Eq. 353}$$

The parameter R_s is called sheet resistance.

The current demanded by digital and analogue circuits provokes a voltage drop which depends on resistance associated to the supply leads. The effect of this drop must be taken into account especially in the analogue blocks, as the accuracy of these circuits can degrade.

During the design phase of Q-Eye system, analogue blocks where the power distribution had to be studied in detail was the Processing Array, Analogue I/O block and AD converters bank in the Image Interface.

A common characteristic in these blocks is the existence of row composed by analogue operators which require a determined bias current. This current is provided by the power lines. Due to the resistance associated to the power line, the voltage experiences a drop along this line. Figure 59 represents a resistive model of line considering the current consumption of each analogue operator.

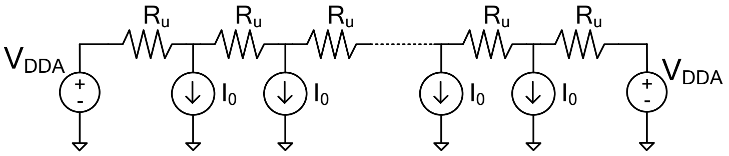


Figure 59. RESISTIVE MODEL OF POWER LINE.

In this simple model, the line has been divided in segments with the same length as the analogue operator to study voltage distribution along the power line. Each analogue operator consumes a static current represented by an independent current source I_0. The drops due to the resistive path from pads to the distribution line in the analogue block have not been considered in this model. The maximum decrement of voltage is in the middle of the row and its value is given by the expression:

$$\Delta V = I_0 \cdot \frac{\frac{N}{2}\cdot\left(\frac{N}{2}-1\right)}{2}\cdot R_u$$ **Eq. 354**

Where N is the number of analogue operators that constitute the row.

To consider the drops existing from pads to the analogue blocks, simply the total bias current required by the corresponding analogue block and the resistance path R_{path} have to be considered, see Figure 60.

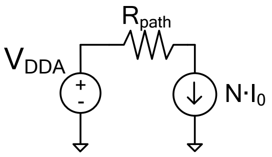


Figure 60. VOLTAGE DROP FROM PADS.

Thus, the voltage drop from pads is determined by the equation:

$$\Delta V = N \cdot I_0 \cdot R_{path}$$ **Eq. 355**

SETTLING TIME. Usually, in a complex mixed-signal system, the current demand associated to the operation varies with the time. This temporal change of current implies an alteration in the power voltage. The current disturbance generates a temporal evolution of power voltage due to the parasitic resistances and capacitances.

4.4.3.2. REFERENCE DISTRIBUTION

The analogue building blocks require references and biasing for their operation. The distribution of these voltages and currents is a key issue in the design of large mixed-signal circuits like the Q-Eye. The bias currents and voltages of building blocks are derived from one or more band-gap reference generators. The biasing references can be distributed in the *current domain* or in the *voltage domain*. These two distribution schemes are described in Figure 61and Figure 62**,** respectively.

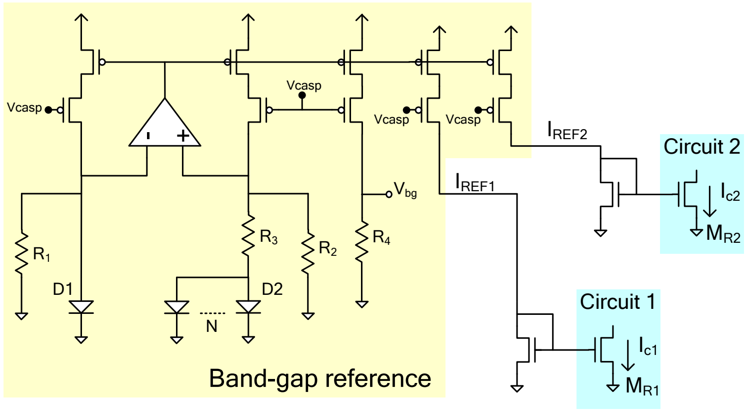


Figure 61. BIASING DISTRIBUTION IN THE CURRENT DOMAIN.

Regarding the distribution in voltage domain, depending on matching level between I_{REF} and I_{c1}, I_{c2} required by the performance of analogue building blocks, the voltage drop along the ground line must be taken into account. In fact, for a large number of circuits connected to the same ground line, the systematic mismatch between the current sources I_{c1}, I_{c2} and I_{REF} may be unacceptable. If the reference is distributed in the current domain, this difficulty disappears. The idea is to route the reference current to the vicinity of the building blocks and perform the current mirror operation locally. However, in this case, the mismatch between I_{REF1} and I_{REF2} and M_{R1} and M_{R2} introduce systematic error.

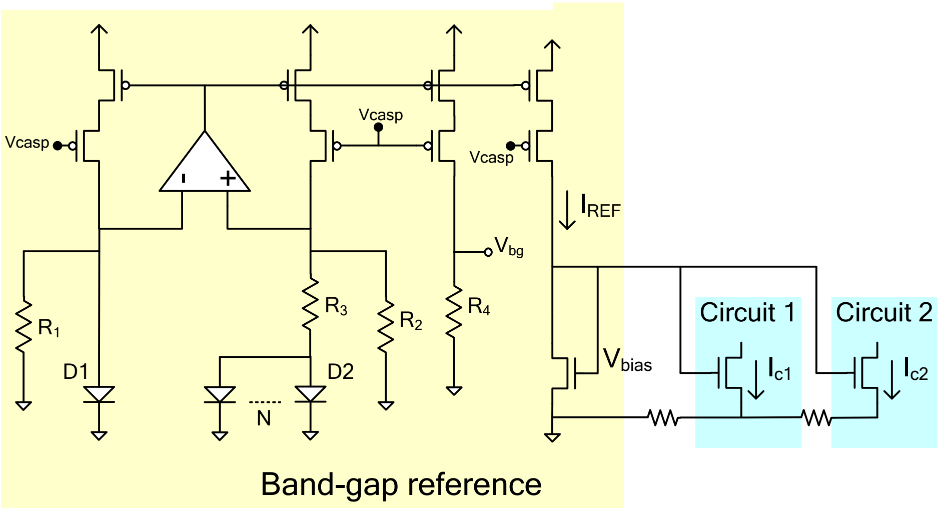


Figure 62. **BIASING DISTRIBUTION IN THE VOLTAGE DOMAIN.**

In the particular case of Q-Eye, to implement a biasing distribution in the current domain for each processing cell is complicated. Therefore, the distribution of biasing references is developed in the voltage domain but following a strategic different to the scheme described in Figure 62. The bias current is generated locally in each processing cell by the circuit shown in Figure 63 using two reference voltages (V_{bn} and V_{bp}).

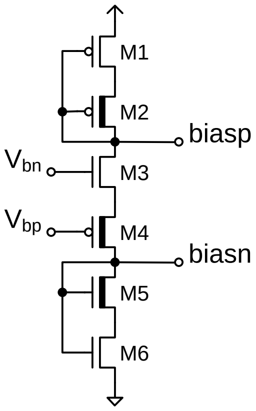


Figure 63. **LOCAL BIASING IN PROCESSING CELL.**

The structure composed by transistors M3 and M4 generates a current independent of power supplies in first approximation. The value of generated current is defined by the difference between the references V_{bn} and V_{bp}. In the Q-Eye system, these references voltages are defined by their corresponding programmable buffers. This means that the bias current of processing cells is programmable. The local biasing voltages biasn and biasp in the cell are generated by transistors M5-M6 and M1-M2 respectively. M2, M3 and M4 are special transistors with low threshold voltage (around 0.2V) in order to obtain the maximum operation range.

The power supplies in the processing array drops due to the current consumption of processing cells which depends on the operation of processing cells and varies along the time. Then, the voltage through the power lines changes with the time and location in this array. It must be taken into account that the building blocks responsible for processing in the analogue domain are basis on single input and single ended circuits. For example, the OTAs implemented during the design of Local Analogue Memories (LAMs) and Multiplier-Accumulator (MAC) are folded inverters in order to optimize in area and power consumption. The signals stored and processed respectively in these circuits are referenced to the quiescent point of inverter, which is defined by the unitary feedback configuration shown in Figure 64**.**

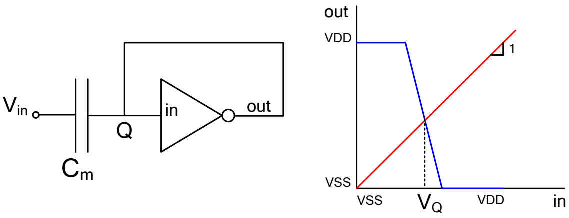


Figure 64. **UNITARY FEEDBACK CONFIGURATION OF INVERTER.**

The voltage associated to the quiescent point depends on the bias current of inverter. Therefore, if this current of inverter changes between the sample instant and hold phase or during hold phase, an error is introduced in the data. The changes of bias current from pixel to pixel imply an increment in the spatial and temporal noise of processed images. Depending on the biasing error distribution over the array and its characteristic, different kinds of artifacts can be observed in the images. For example, different offsets frame to frame can be detected if a global change of bias current in whole array is produced during the processing time. Under these conditions, a biasing strategic in the voltage domain similar to that described in Figure 62 must not be implemented, because the ground line varies with the time and space in the array. This fact provokes that the bias current defined by the reference voltage V_{bias} and ground is different along the cells of array generating artifacts in the image. A constant bias current is obtained locally in each processing cell with the circuit proposed and described in Figure 63.

The general bias scheme implemented for the processing array of Q-Eye system is described in Figure 65**.** The band-gap generates two references, the biasing voltage Vbiasg used to bias the output buffers of DAC and the reference voltage VREF for DACs. The biasing voltages Vbp and Vbn are generated with the DACs and distributed to the processing array and other building blocks of system.

The current generates by the references V_{bn} and V_{bp} depends on the process parameters. Therefore, in order to avoid important variations in the consumptions of smart sensor with the process parameters, a special block has been design whose circuit is depicted in Figure 66.

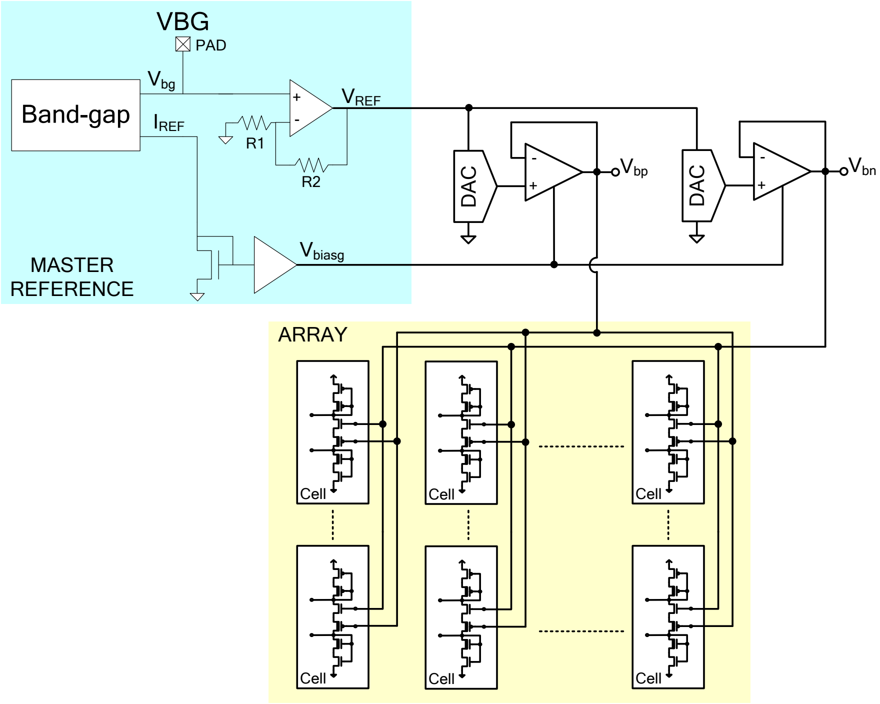


Figure 65. **BIASING DISTRIBUTION IN THE PROCESSING ARRAY.**

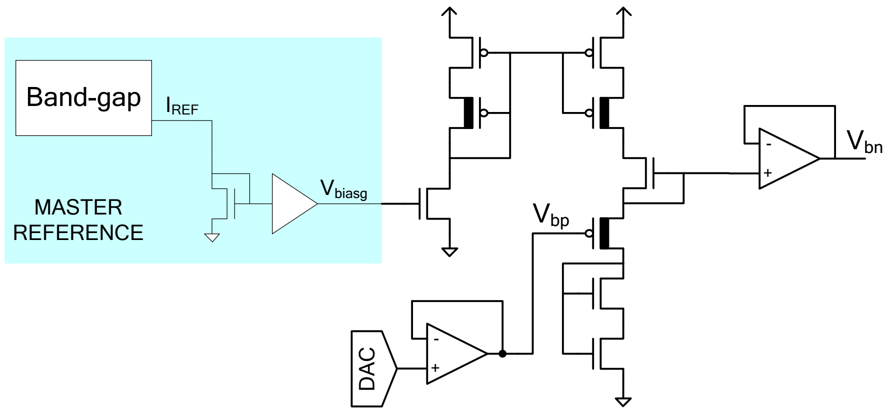


Figure 66. CIRCUIT FOR ADJUSTMENT TO THE PROCESS PARAMETER CHANGES.

This adaptation circuit generates the biasing voltage V_{bn} once the other biasing voltage V_{bp} has been programmed, taking as reference the band-gap current I_{REF}.

The analogue references required for the operation of Q-Eye system, for example, Processing Array, Input/Output Converters, etc., are generated by the programmable buffers, which are composed by a DA converter and an output buffer. For large mixed-signal systems, like the Q-Eye, the distribution of this analogue references throughout the chip is a critical point to take into account during the design phase.

Even though reference generators are low frequency circuits, they may affect to the speed of circuits which they feed. Furthermore, various building blocks may experience crosstalk through reference lines. These issues arise because of the finite output impedance of reference voltage generators. The scheme depicted in Figure 67shows an example. The voltage at node V_1 is heavily disturbed by the Circuit 1, and this disturbance passes through the coupling capacitance to the reference node V_R.

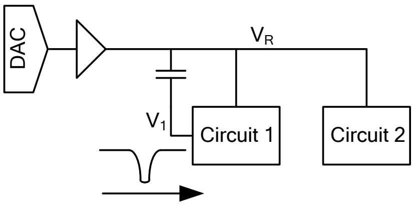


Figure 67. CROSSTALK THROUGH THE REFERENCE LINES.

For fast changes in V_1 the buffer or operational amplifier cannot maintain the reference V_R constant experiencing large transient changes. Also, the duration of this transient may be quite long if the buffer has a slow response. For this reason, many applications may require a high-speed operational amplifier in the reference distribution. For large mixed-signal systems, even the own RC constant associated to the reference line limits the transient respond. In this case, a tree distribution scheme can be implemented. This new structure of distribution is represented in Figure 68.

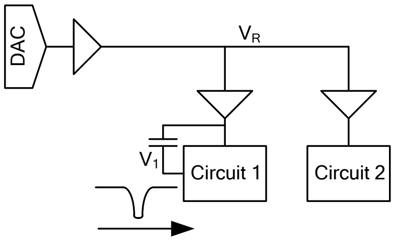


Figure 68. TREE DISTRIBUTION STRUCTURE.

In systems where the power consumed by the reference circuit must be small, the use of a high-speed operational amplifier may not be feasible. Another technique is to connect the reference node V_R to ground node through a large capacitor C_B in order to suppress the effect of external disturbances, see Figure 69. This approach implies two problems. One issue is related with the stability of operational amplifier. This stability must not degrade with the addition of the capacitor. The other issue is related with the transient response. The addition of large capacitor slow down the dynamic response associated to the output buffer, its value must be much greater than the capacitance that couples the disturbance to the reference node V_R in order to minimize the amplitude of disturbance, such that its value is negligible. If the decoupling capacitor C_B is not enough large, then the reference voltage experiences a change and takes a long time to return to its original value. Therefore, there is a tradeoff between the speed of output buffer and the decoupling capacitance.

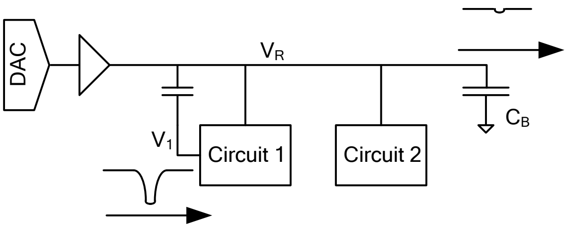


Figure 69. **DECOUPLING CAPACITOR TECHNIQUE.**

During the design phase of output buffer belong to the reference generator is critical to determine the current demanded by the circuits fed by the corresponding reference voltage, both static and dynamic current. For example, in the case that the reference node V_R is capacitive, the static current demand is null, but there is a dynamic current whose magnitude depends on the frequency operation of circuits connected to the reference and the capacitances implied in the operation.

Another important point to consider in the reference distribution is the integrity of signals. When the mixed-signal system is very large the routing of references must be developed avoiding coupling with noisy nodes, for example, digital signals operating in high frequency.

Regard image quality, the errors and artifacts in sensed and processed images depend on accuracy of:

- Circuitry at pixel level
- Readout circuitry and data converters belong to the image interface
- Generation and distribution of analog references
- Power supplies distribution

Therefore, the design, distribution strategy and physical implementation of global analog references required by the pixel circuitry and readout circuitry is critical in order to avoid artifacts in the images (shadows, vertical and horizontal fixed patterns, etc.). Usually, due to the resolution of sensing array, the dimensions of image sensors are significant. In these cases, the distribution strategy and points treated in this last section play a fundamental role to obtain images with the highest quality. For instance, considering a global reference for all AD converters in a one readout channel per column approach, a high level of flicker noise associated to this global analog reference could generate random line noise in acquired images.

In a similar way, an inefficient power supplies distribution could provoke artifacts in the acquired images. For this reason, it is very important to design the system with the maximum power supply rejection. The data acquired and digitalized by the image sensor or data resulting from a sequence of operations should be as independent as possible of variations in the power supplies.

CHAPTER 5 - THE EYE-RIS VISION SYSTEM

Vision requires sensor data be transmitted and processed through a *hierarchy* of processing levels involving different tasks which can be roughly grouped in three groups, namely:

- *Low-level* tasks,
- *Medium-level* tasks,
- *High-level* tasks.

Throughout this chapter low-level tasks will be called *pre-processing* or *early-vision* tasks, and medium-level and high-level tasks will be called *post-processing* tasks. Early-processing comprises tasks related with image enhancement, restoration, feature extraction and segmentation. Post-processing is composed by representation, description and recognition algorithms. As data progress through the hierarchy, the *level of abstraction* of data structures increase and the *dimensionality* of the data themselves decrease. Thus, while the set of sensory data includes all *spatial samples* acquired by the front-end imager, each encoded by several bits, the outcome of a vision function may consist of a single bit to just codifying whether an object is either present or absent in a scene [Gonz92] [Toma06]. Of course, this is an extreme situation, but in general the dimensionality of output data structures is much smaller than that of input ones.

Let us use **F** to denote the amount of data used to encode input images, **f** for the data obtained after early processing and ***f*** for the data obtained after mid-level processing. In conventional vision system architectures, whose front-ends are conventional imagers (either CIS or SCIS for CMOS-VLSI devices) without embedded processing capabilities, the whole set **F** must be transmitted and stored for ulterior processing. This is an inefficient process which results into systems with large *latency* and large SWaP factors. On the contrary, systems employing CVISs like Q-Eye at the sensory front-end, deliver the set **f** <<**F** for further processing, thus yielding increased speed and reduced SWaP factors. Following the presentation of Q-Eye pixels and chips in previous chapters, this chapter describes the architectures of the vision systems built by using these chips, the so called Eye-RIS vision systems. Two different versions of Eye-RIS are presented in this chapter, namely:

- Eye-RIS_v1, which employs an *off-chip* digital processor.
- Eye-RIS_v2, where the digital processor is *on-chip*, embedded on the same silicon substrate than the Q-Eye.

While both systems share architectural concepts, Eye-RIS_v1 has larger functional power, featured by its off-chip components, while Eye-RIS_v2 trades functional power by system compactness.

5.1. EYE-RIS SYSTEMS ARCHITECTURES

5.1.1. GENERALITIES ABOUT THE PROCESSING CHAIN OF VISION

Vision applications involve a variety of processing tasks with different complexity and dissimilar computational demands. As illustrated by Figure 5 of Chapter 1 and in Section 3.1.2 of Chapter 3, the processing chain of vision involves a hierarchy of abstraction levels. Each level has input data which are processed to produce output data with new structure. This is usually accompanied by an increase of the abstraction level of the data. Also, the amount of information gets reduced as processing progress through the chain.

Figure 1 [Gonz92] [Gonz15] includes at the top-left those levels in the hierarchy that are characterized by *regular* data structures. Hence, computational flow at this stage consists of a set of predefined operations that are performed repeatedly onto the data, despite the particular value represented by these data. In contrast, levels at the bottom-right in the figure involve *irregular* computational flow characterized by data-dependent decisions and unpredictability. Obviously, the regularity of the computational flow, the data size and number, the simplicity of the data structure and the level of abstraction all play a significant role in the complexity of vision processing algorithms, and must be carefully taken into account to design efficient vision system architecture [Pirs98].

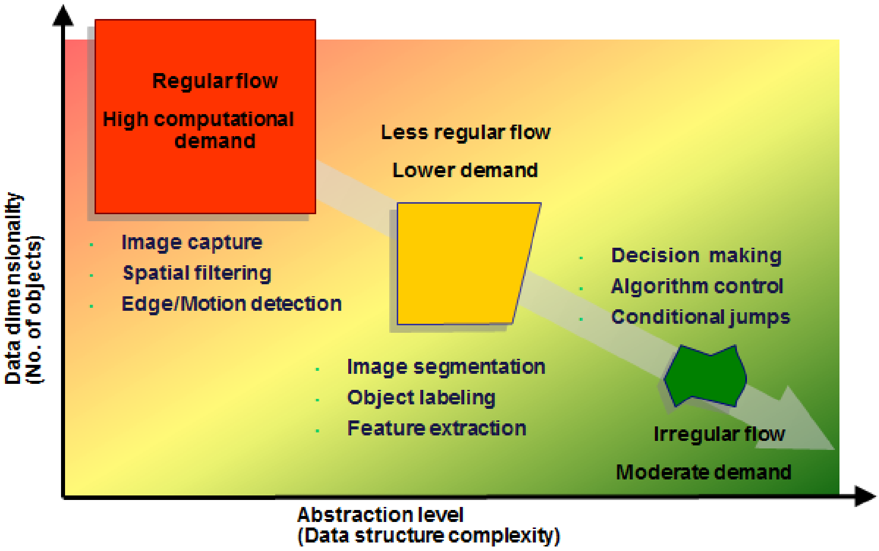


Figure 1. PROCESSING HIERARCHY, FROM LEFT-TOP TO BOTTOM-RIGHT, IN VISION.

Left-top corresponds to input data retrieved from the sensor and the bottom-right corresponds to data on the basis of which decision-making is made. The dimensionality of the data increases from left-top to right-bottom, while the abstraction level of the data increases in the same direction.

The first stage of the vision processing chain is usually devoted to **image enhancement and restoration**. During this stage, non-idealities of the sensing process (for instance, temporal and spatial noise) are compensated and the quality of captured images is improved in relation to some image features. This is achieved by applying several filters (convolution masks, diffusion process, etc.) and by performing point-to-point transformations. The output data provided by enhancement and restoration tasks is still a matrix of real numbers, which are the input of second stage composed of **feature extraction tasks**. Feature detection is also a low-level image processing operation (belonging to pre-processing or early-processing stage). Usually, feature extraction operations examine every pixel to verify if there is a feature present at that pixel considering its neighborhood. Interesting characteristics of images for subsequent image processing algorithms are *edges*, *corners* or *interest points*, *blobs* or *region of interest*, *ridges*, etc. Examples of some typical feature detectors among all that have been developed are indicated in table Table.1,[Gonz15] [Cann86].

FEATURE DETECTOR	DETECTED CHARACTERISTIC
Sobel operator	edge
Prewitt	edge
Roberts	edge
Canny operator	edge
Laplacian of Gaussian	Corner and blob
Difference of Gassusians	Corner and blob
Grey level blobs	blob

Table.1. Feature detectors.

Data obtained after feature detection are images containing the characteristics interesting for the segmentation process. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions. Based on these features, the *segmentation* algorithms extract from an image the objects or other entities of interest for subsequent processing such as *description* and *recognition*. More precisely, **image segmentation** is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristic. The goal of segmentation is to simplify the representation of an image into a data structure that is easier to analyze.

At this point, the simplified representation of an image based on regions is the input to the next abstraction level which comprises the **representation and description tasks**. The description of regions or objects is a previous stage required for the recognition tasks. In this abstraction level, the characteristics associated to each object are determined in order to recognize them and develop a determined classification. The representation of regions is based on two types of characteristics:

- *External* characteristics consisting basically of object shapes described by *boundary descriptors*: length of contour, perimeter, curvature, major and minor axis, shape numbers, eccentricity, Fourier descriptors, moments, etc.
- *Internal* characteristics focused on reflectivity properties and described by regional descriptors: area of a region, compactness, topological descriptors, textures, moments, etc.

Segmentation techniques yield raw data in the form of pixels along a boundary, or pixels contained in a region. A region is a connected component, and the boundary of a region is the set of pixels in the region that have one or more neighbors that are not in the region. Points not on a boundary or region are called *background* points. The simplest representation consists of a binary image where region or boundary points are represented by “1” logic value and background points by “0”. Although these data are sometimes used directly to obtain descriptors, it is a standard practice to use schemes that compact the data into representations that are considerably more useful in the computation of descriptors. Some examples of these representation approaches are: *chain codes*, *polygonal approximations*, *signatures*, *boundary segments*, *skeleton of a region*, etc. On the one hand, it is clear that data obtained at the outcome of description level have smaller dimensionality than those data available at the input of enhancement tasks. On the other hand, the former data are organized into more complex structures which are appropriate for computation of properties.

The representation and descriptors resulting from description stage are the input of **recognition tasks,** which are aimed to identify each object segmented in a scene. Approaches to computerized pattern recognition can be divided into two areas:

- *Decision-theoretic* approaches deal with patterns described using quantitative descriptors, such as length, area, textures, etc. Methods of decision-theoretic are Minimum-Distance Classifiers, Matching by Correlation, Optimum Statistical Classifiers, Bayesian Classifier, Artificial Neural Networks, etc.
- *Structural* approaches work with patterns best represented by symbolic information, such as strings, and described by the properties and relationships between those symbols.

The data structure is based on pattern concept, which is an arrangement of descriptors. A pattern class is a family of patterns that share a set of common properties. Pattern recognition by vision machines involves techniques for assigning patterns to their respective classes. The two principal pattern arrangements used in practice are vectors for quantitative descriptions and strings for structural descriptions.

A basic concept in recognition algorithms, especially in **decision-theoretic approaches**, is the **pattern matching**, which measure the similarity between objects, ranging from the trivial case of comparing two pixels to the highly complex problem of determining in some meaningful way how similar two or more objects are. Some of the descriptors presented previously can be used as the basis of comparison between two image regions or objects. Techniques to measure the level of similarity are distance measures, correlation, boundary matching, etc.

Decision-theoretic approaches to recognition are based on the use of decision functions which are used to assign objects to classes. These functions $d\left(\vec{x}\right)$ are characterized by a weight vector $\vec{w}=\left(w_1,w_2,...,w_n,w_{n+1}\right)$ or decision function parameters:

$$d\left(\vec{x}\right)=f\left(\vec{x},\vec{w}\right) \qquad \textbf{Eq. 356}$$

Given a n-dimensional pattern vector represented by $\vec{x}=\left(x_1,x_2,...,x_n\right)$ and W pattern classes denoted by $\left(\omega_1,\omega_2,...,\omega_W\right)$, the basic problem in decision-theoretic pattern recognition is to find W decision functions $\left(d_1\left(\vec{x}\right), d_2\left(\vec{x}\right), ..., d_W\left(\vec{x}\right)\right)$ with the property that if a pattern $\vec{x}$ belongs to class ω_i, then:

$$d_i\left(\vec{x}\right) > d_j\left(\vec{x}\right) \quad j = 1, 2, ..., W; j \neq i$$ **Eq. 357**

In other words, an unknown pattern $\vec{x}$ is said to belong to the *ith* pattern class if, upon substitution of $\vec{x}$ into all decision functions, $d_i\left(\vec{x}\right)$ yields the largest numerical value. Finding decision functions entails estimating parameters from patterns that are representative of the classes of interest. Patterns used for parameter estimation are called training patterns or training sets.

Structural recognition techniques are based on representing objects as strings, trees or graphs and then defining descriptors and recognition rules based on those representations. The key difference between decision-theoretic and structural methods is that the former considers quantitative descriptors expressed by numeric vectors. And the other hand, the structural techniques deal with symbolic information, in a way that exploits any structural relationships that may exist among objects contained in an image. For this purpose, techniques for relating components of an image based on grammatical concepts is being considered as an effective approach to these structural-description problems. The objects contained in an image are represented by connected string of symbols. These strings are handed by the formal language theory, field which deals with the study of mathematical models used for the generation, translation; or other process involving strings of an artificial language. In addition to the formal language techniques, there is an additional method based on matching symbols or strings to perform classification.

Another fundamental concept in the recognition process is the **learning algorithm**. The training patterns of each class are used to compute the parameters of the decision function corresponding to that class. After the parameters have been estimated, the structure of classifier is fixed, and its eventual performance will depend on how well the actual pattern populations satisfy the underlying statistical assumptions made in the derivation of the classification method being used. The minimum-distance classifier is specified completely by the mean vector of each class. Similarly, the Bayes classifier for Gaussian populations is specified completely by the mean vector and covariance matrix of each class of parameters. But when the pattern classes are not characterized by Gaussian probability density functions, the methods described previously are not valid. Designing a statistical classifier becomes a much more difficult task because estimating multi-variate probability density functions is not a trivial endeavor. In practice, such decision-theoretic issues are best handled by methods that yield the required decision functions directly via training. Therefore, the principal approach in use today for this type of classification is based on artificial neural networks, which are designed to be trained using an inductive learning algorithm. After the network is initialized, it can be modified to improve its performance on input/output pairs. To the extent that the learning algorithms can be made general and efficient, the artificial neural networks become useful tools for creating a wide variety of high-performance applications.

5.1.2. PROGRESSIVE PROCESSING, CVIS-BASED ARCHITECTURAL CONCEPT

The top drawing in Figure 2 shows basic architectural concepts for a conventional vision system and the bottom one for a system employing a CVIS like the Q-Eye at the front-end (bottom drawing). The main difference between both concepts is highlighted at the figures. On the one hand, in conventional architectures there is a clear separation between the sensor and the processor. All vision tasks (namely pre- and post-processing ones) are completed by the processor and it means that a large set of *raw* data [11], represented by **F**, must be delivered to

[11] Actually, as described in Chapter 1, modern smart imagers, SCIS, may deliver corrected digital data, free from artefacts, and hence ready for processing. The term raw data is usually employed to refer to the signals measured at the pixel sensors, prior to correction. Hence, the usage of the term raw is not absolutely accurate in this context. However, it is employed to emphasize that any and all data acquired by the sensor must be delivered to the processor.

the processor stages. Because **F** is huge in the case of images, computational and memory resources of the processing section must be huge as well. On the other hand, the architecture at the bottom delivers pre-processed data whose dimensionality, represented by **f**, is significantly smaller, thus relaxing computational and memory demands of the processing stage.

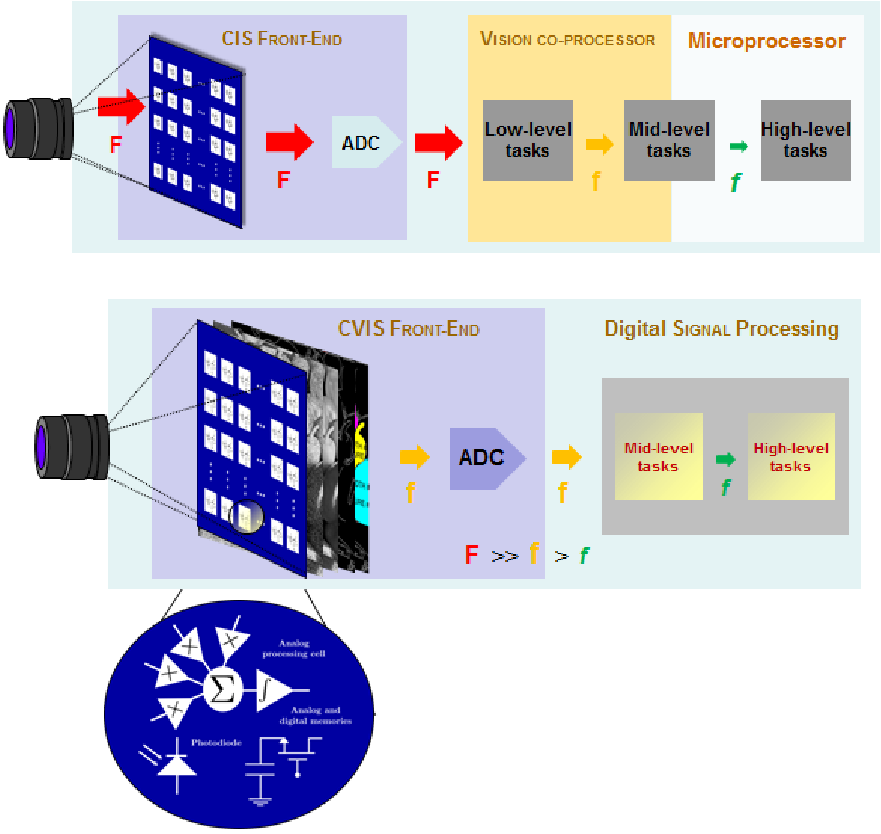
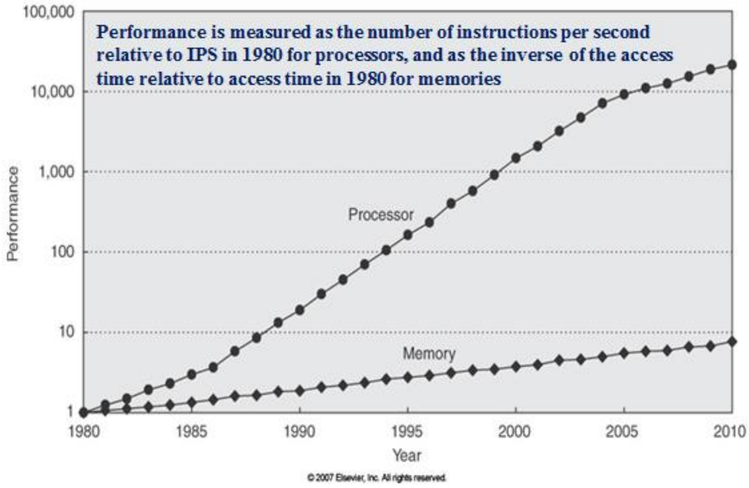


Figure 2. Concept of Conventional Vision System Architecture (top); Concept of Vision System Architecture with Processing Embedded at the Sensory Front-End (Bottom)

Conventional architectures make an early, clear separation between *analog* and *digital* signal domains. Analog and mixed-signal techniques are employed just for acquiring, conditioning and encoding signals. All processing is realized in digital domain, using digital processors. This approach benefits form the increased robustness, flexibility and scalability of digital circuits as compared to analog ones [ITRS13]. It is argued that advances in VLSI technologies and *multi-core* processor architectures will confront any processing demand; in other words that processing capabilities featured by deep submicron VLSI processors are beyond the requirements posed by today´s applications. An obvious counter-argument is that new applications demands will emerge to exceed any processor performance expectation. In any case, as illustrated by the drawing at the right inset, processor performance has prevalently increased over the years [Henn06]. However, for imaging applications, memory demands are so important as processing demands and, memory technologies do not evolve so quickly as



processing technologies. This is the so-called *memory wall,* also illustrated at the figure inset. Improved vision system architectures are hence advisable.

In the case of the CVIS-based architecture of the bottom of Figure 2, the separation between analogue and digital domains is not made so early; namely part of the processing is completed in mixed-signal domain, close to the sensors. Thus, processing is *distributed* between analog and digital domains, is made in more *progressive* way and much less data, represented by **f** <<**F** need to be transmitted and stored for processing in the digital processors. Another rationale for this architectural strategy takes into account the nature of data and algorithms involved in pre-processing and post-processing respectively. The algorithms belonging to early-processing are characterized by a regular computational flow involving identical operations on large number samples. Both input and output data consist of simple data structure, such as pixels in an image. The computing power required by these tasks is high due to the number of data and speed requirements of real time applications. The early-processing tasks are also denoted as data-intensive tasks. These tasks are highly regular and offer a large potential of data parallelism, strategy that relaxes the performance requirements associated to the processors. This justifies the usage of a dedicated piece of hardware consisting of an array of processors, one per data, all executing the same operation simultaneously according to the paradigm of Single Instruction Multiple Data (SIMD) machines [Rosk01]. Regarding post-processing tasks, they deal with smaller number of symbols and objects. Data structures are complex and of variable size. The computational flow is irregular and cannot be predicted in advance. Performance requirements for these high-level tasks, denoted also by math-intensive tasks, are less demanding, but a high degree of flexibility is mandatory. This justifies the usage of flexible digital processors for these tasks. In this worth mentioning that this splitting between pre- and post-processing and types of architecture is not so evident during the implementation phase of a vision system. An automated partitioning of applications into hardware and software portions is an old research challenges in the field of hardware-software co-design [DeMi97] [Gupt95] and remains essentially unsolved.

Besides following the architectural concept at Figure 2 bottom with the Q-Eye as front-end, Eye-RIS systems are *embedded systems* in the sense that the digital signal processing stage responsible of post-processing is part of the same physical system as the front-end. In fact, this processor lies on the same silicon substrate than Q-Eye in the case of Eye-RIS_v2. Overall system (acquisition + early-processing + late processing) is conceived by following a *software-hardware co-design* approach. Usually, embedded vision systems are conceived for *specific* tasks. However, Eye-RIS systems are conceived as programmable, "general-purpose" vision processors. Programmability is achieved by employing software reconfigurable circuit structures at the CVIS front-end, and by employing general-purpose processors for post-processing.

5.1.3. EYE-RIS ARCHITECTURE OUTLINES

The drawing at the top in Figure 3 shows the block diagram of Eye-RIS_v1 system [Anaf1v3]. While that at the bottom corresponds to Eye-RIS_v2. Both architectures employ the same CVIS at the front-end, namely the Q-Eye. The Q-Eye is a SIMD processor consisting of an array of interconnected mixed-signal processors, one per pixel, that operate in parallel. Its architecture and parameters are conceived for efficient completion of pre-processing vision tasks. The implementation of regular algorithms in hardware involves mapping of operations onto dedicated processing elements and representation of data dependencies by hardware interconnections or intermediate memories. For regular algorithms of image processing, array processors are typically derived as appropriated hardware structures. Favorable properties of array structures are the incorporation of parallel processing and pipelining and the locality of connections between processing elements. Thus, high performance and throughput are obtained at moderate hardware expense.

Like any vision system, the Eye-RIS systems include a sensor (in this case the Q-Eye sensor-processor), a Digital Image Processor (DIP), a microprocessor, memories and I/O and communication ports. Since the Q-Eye is software-controllable, Eye-RIS systems must include a dedicated microprocessor to control and configure the operation of this CVIS. The Q-Eye controller includes a simple microprocessor responsible for implementing the FPP microinstructions of Q-Eye system. Users can defined a particular algorithm or sequence of operations through the NIOS microprocessor, and the microprocessor of Q-Eye controller sends the microinstructions to Q-Eye through the control interface.

In the case of Eye-RIS_v1, the NIOS-II microprocessor together with the Digital image Processor and Q-Eye controller are implemented on FPGA. However, in the Eye-RIS_v2, the front-end CVIS is not the Q-Eye, but the so-called Eye-RIS_VSoC, which embeds the Q-Eye, the DIP, the Q-Eye controller and the NIOS-II on a common silicon substrate. Thus the Eye-RIS_v2 is composed by this CVIS and some external memories. Therefore, the compactness level achieved by the Eye-RIS_v2 system is larger than level obtained with the Eye-RIS v1. The implementation of an External Serial Flash Interface in the Eye-RIS VSoC allows the Eye-RIS v2 to work in stand-alone mode without requiring any external host to develop a particular application.

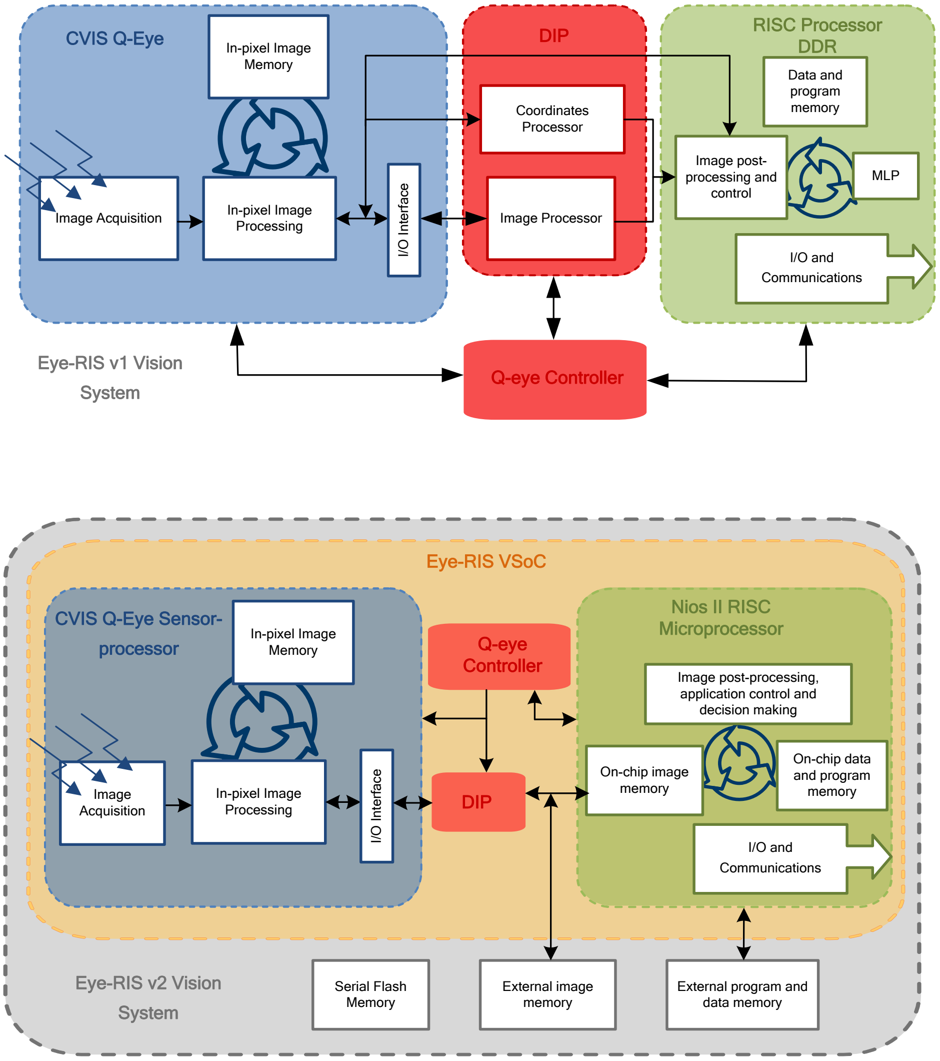


Figure 3. **EYE-RIS VISION SYSTEMS BLOCK DIAGRAMS**

The top corresponds to Eye-RIS_v1 and the bottom corresponds to Eye-RIS_v2

5.2. ILLUSTRATING PROGRESSIVE, DISTRIBUTED PROCESSING: PART-FINDING APPLICATION

This section describes a typical application in the field of Machine Vision (part-finding application), in order to present an example for the concepts introduced in previous sections and show the operations and mainly capabilities required by the building blocks of a vision system to develop a vision application in general.

The aim of part-finding application is to detect and classify those parts which meet certain similarity conditions with a pattern object among a set of parts with different shapes and sizes.

This classification must be performed independently of rotation with which the objects are presented in the captured images.

The vision system must resolve two main issues developing this application. The first is the detection of a possible part and the second its classification after analyzing it, determining if a part is correct by the comparison with the pattern. The detection is carried out by the segmentation, which can be implemented using thresholding techniques. Before segmentation, the captured images are processed to improve the image quality and extract features for the segmentation stage.

For the classification, the comparison between the part under studying and the pattern is based on a determined descriptors obtained in the representation and description phase. In this example, the characteristics or descriptors considered for the analysis are blob properties and boundary parameters.

Blob properties obtained for the description of objects in the scene are indicated below:

- Area: Defined as the size of blob expressed in number of pixels.
- Major axis: Length (expressed in number of pixels) associated to the major axis of an object, which is calculated through the longest axis that can be traced between two contact points of blob.
- Perimeter: The number of pixels which compose the blob boundary.
- Mean value: The mean value of intensity level associated to a particular blob gives a measurement of brightness for a possible part.
- Standard deviation: It is a measurement of dispersion respect to the mean value, indicating the object contrast.
- Number of holes: Holes included in a blob. The minimum size should have a hole to be considered must be defined.
- Histogram: Indicates the percentage of pixels belong a determined blob with a certain gray level. Because their calculation is very laborious, several gray level intervals are defined, thus, the blob histogram is the percentage of pixels located in each interval or band.

In relation to the area descriptors presented in the previous list, the comparison of descriptors in the classification stage is based on the percentage of error, which is given by the equation:

$$\frac{\left|Feature - Feature\_pattern\right|}{Feature\_pattern}\cdot 100 \geq Tol(\%) \Rightarrow Error \qquad \textbf{Eq. 358}$$

In the case of hole number, the comparison is based on the absolute error:

$$\left|Number\_holes - Number\_holes\_pattern\right| \geq Tol \Rightarrow Error \qquad \textbf{Eq. 359}$$

And for histogram comparison, the process is more complex. In a first step, the histogram intervals are compared independently using the error percentage Eq. 358. In a second step, the number of intervals where the comparison faults is determined and compared with the permitted tolerance.

On the other hand, the boundary descriptors considered for the classification algorithm are described below:

SIGNATURE: The object boundary is represented as a one-dimensional polar function. For this, a characteristic point of object (for instance, the centroid) is selected, and the distance of each pixel belonging to the perimeter to the characteristic point (ρ) is represented in function of angle defined for both points (θ), see Figure 4.

In this case, the two functions corresponding to the object under analysis and the pattern are compared during the classification stage calculating the error between them and verifying if the error is larger than the tolerated error for each θ value.

A problem associated to this technique is that the representation depends on which is the starting point to explore the contour. For this reason, a comparison method of signatures based on correspondence between points determined by the minimum distance error must be implemented.

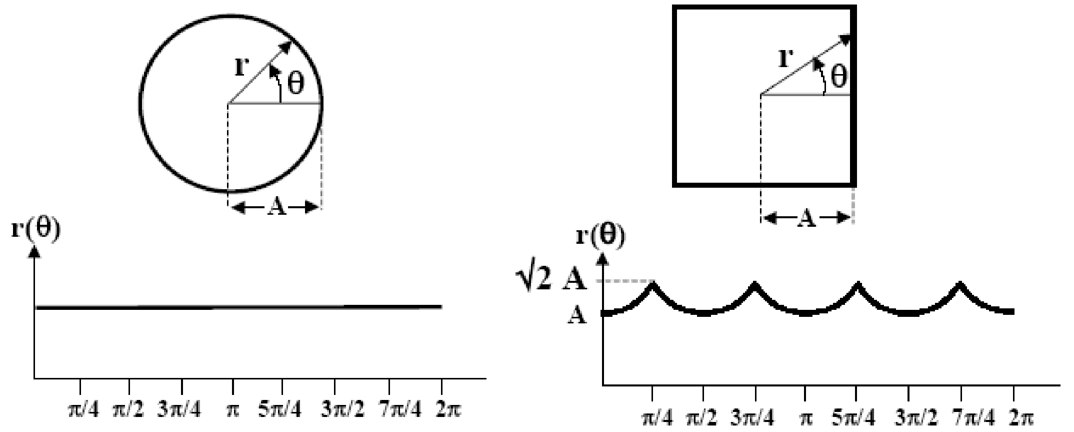

r
θ
A
r(θ)
π/4 π/2 3π/4 π 5π/4 3π/2 7π/4 2π
r
θ
A
r(θ)
√2 A
A
π/4 π/2 3π/4 π 5π/4 3π/2 7π/4 2π

Figure 4. SIGNATURE REPRESENTATION.

HOUGH TRANSFORM: The objective of this technique is to find object instances within a certain class of shapes by a voting process. This voting procedure is developed in a parameter space, from which object candidates are obtained as local maxima in accumulator cells defined in the parameter space.

The classical Hough transform has been concerned with the identification of lines in the image [Duda72] [Shap01]. The general equation of a straight line is:

$$y_i = a \cdot x_i + b \qquad \text{Eq. 360}$$

However, if this equation is written as:

$$b = -x_i \cdot a + y_i \qquad \text{Eq. 361}$$

And the ab plane is considered, which is denoted by parameter space, the coordinates (a,b) represents a single line in the xy plane which contains the point (x_i, y_i). Considering a second point (x_j, y_j) which also has a line in parameter space associated with it, when this line intersects the line associated to (x_i, y_i) at (a', b'), this point in the parameter space represents the line which contains in the xy plane both points (x_i, y_i) and (x_j, y_j).

The Hough transform subdivides the parameter space into so-called accumulator cells. The dimension of accumulator array is equal to the number of unknown parameters associated to the Hough transform problem. The cell at coordinates (i, j), with accumulator value $A(i, j)$, corresponds to the square associated with parameter space coordinates (a_i, b_j). Initially these cells are set to zero. Then, for every point (x_k, y_k) in the image plane, the parameter a is set to each of the allowed subdivision values on the a axis and solve for the corresponding b using the equation Eq.361. The resulting b values are then rounded off to the nearest allowed value in the b axis. If a choice of a_p results in solution b_p then:

$$A(p,q) = A(p,q) + 1 \qquad \text{Eq. 362}$$

At the end of this procedure, a value M is $A(i, j)$ corresponds to M points in the xy plane lying on the line described by the equation Eq.360. The accuracy of the collinearity of these points is established by the number of subdivisions in the ab plane.

A problem with using the representation given by equation Eq.360 is that both the slope a and intercept b approach infinity as the line approaches a vertical position. For this reason, it is better to use a different pair of parameters, considering the normal representation of a line, given by the equation:

$$x \cdot cos\beta + y \cdot \sin\beta = \rho \qquad \textbf{Eq. 363}$$

The meaning of parameters ρ and β is described in Figure 5.

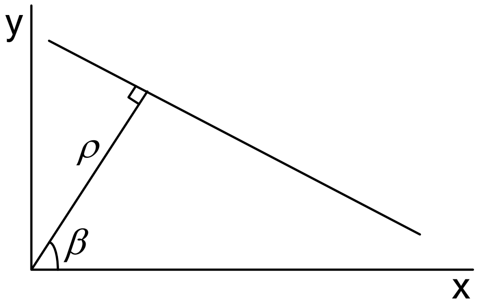


Figure 5. NORMAL REPRESENTATION.

Although the example has been focused on straight lines, the Hough transform is applicable to any function of the form:

$$g\left(\vec{x}, \vec{c}\right) = 0 \qquad \textbf{Eq. 364}$$

Where $\vec{x}$ is a vector of coordinates and $\vec{c}$ is the vector of coefficients or parameters.

Although the version of the transform presented above applies only to find object contours described with an analytic equation, there is a modification of Hough transformation based on the principle of template matching, [Ball81]. This modification called Generalized Hough Transform can be used to detect an arbitrary object described with its model. The problem of finding the object (described with a model) in an image can be solved by finding the model's position in the image. The problem of finding the model's position is transformed to a problem of finding the transformation's parameter that maps the model into the image. As long as the values of transformation parameters are known, the position of the model in the image can be determined [Kass99].

In part-finding applications, any curve can be modeled using the parameters ρ_i (distance from a curve point $\left(x_i, y_i\right)$ to a reference point $\left(x_{ref}, y_{ref}\right)$) β_i (the direction defined by the curve point $\left(x_i, y_i\right)$ and reference point $\left(x_{ref}, y_{ref}\right)$) and the coordinates associated to the reference point $\left(x_{ref}, y_{ref}\right)$, see Figure 6 Each point belonging to the object contour is described by the equation:

$$\begin{aligned} x_{ref} &= x_i + \rho_i \cdot \cos\beta_i \\ y_{ref} &= y_i + \rho_i \cdot \sin\beta_i \end{aligned} \qquad \textbf{Eq. 365}$$

In the following paragraphs, the algorithm carried out during the detection and classification phases based on Generalized Hough Transform will be presented. For this purpose, a general case where the rotation of object contour respect to pattern object is unknown will be considered. This rotation is determined by a angle parameter denoted by θ. In a first stage the pattern object is characterized developing the following steps:

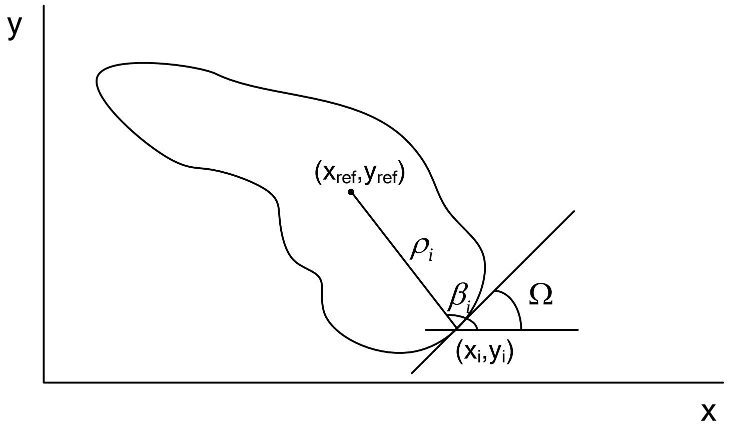


Figure 6. CURVE REPRESENTATION..

- Extraction of boundary associated to the patter object.
- Definition of a reference point inside the pattern contour, $\left(x_{ref}, y_{ref}\right)$.
- Generation of a LUT (lookup table) where each point of pattern contour $\left(x_i, y_i\right)$ is represented by:

$$\left(\rho_i, \beta_i\right) = f\left(\Omega\right) \qquad \textbf{Eq. 366}$$

  Where Ω is the orientation, defined by the gradient at the boundary point $\left(x_i, y_i\right)$, see Figure 6

- Generation of accumulator array with the same size as the image under analysis, indexed by coordinates and initialized to zero. Since the rotation of object respect to the pattern is unknown, the number of accumulators created is equal to the possible rotation values. Each accumulator cell is defined by $\left(x'_{ref}, y'_{ref}, \theta\right)$.

After pattern characterization, the vision system can begin to develop the application which is composed by the following steps:

- Image acquisition.
- Image enhancement.
- Boundary detection.
  - Calculation of the orientation Ω associated to each point $\left(x_i, y_i\right)$ of boundary.
  - For each pair $\left(\rho_i, \beta_i\right)$ indexed by Ω, the coordinates of reference point are estimated:

$$\begin{aligned} x'_{ref} &= x_i + \rho_i \cdot \cos\left(\beta_i + \theta\right) \\ y'_{ref} &= y_i + \rho_i \cdot \sin\left(\beta_i + \theta\right) \end{aligned} \qquad \textbf{Eq. 367}$$

  - This operation must be carried out for each possible rotation value θ.
  - The accumulator corresponding to the rotation consider must be incremented, $\left(x'_{ref}, y'_{ref}, \theta\right)$.
  - The maximum among accumulators gives the contour and its rotation in the image.

The boundary descriptors presented previously (signature and generalized Hough transform) require a microprocessor with high performance of processing. In the case of Eye-RIS system, the NIOS-II has a limited processing speed. Therefore, for high speed inspection applications the blob descriptors has been considered during classification phase. Moreover, due to the characteristics of the APAP operations included in the Eye-RIS system, the Q-Eye can operate as co-processor helping to NIOS-II in the calculation of blob descriptors (area, major axis, perimeter, mean, number of holes) in order to obtain them in an efficient manner in power and speed.

The operation flow performed by the vision system during the part-finding application is described in Figure 7, once the characterization of pattern object has been developed and their results stored.

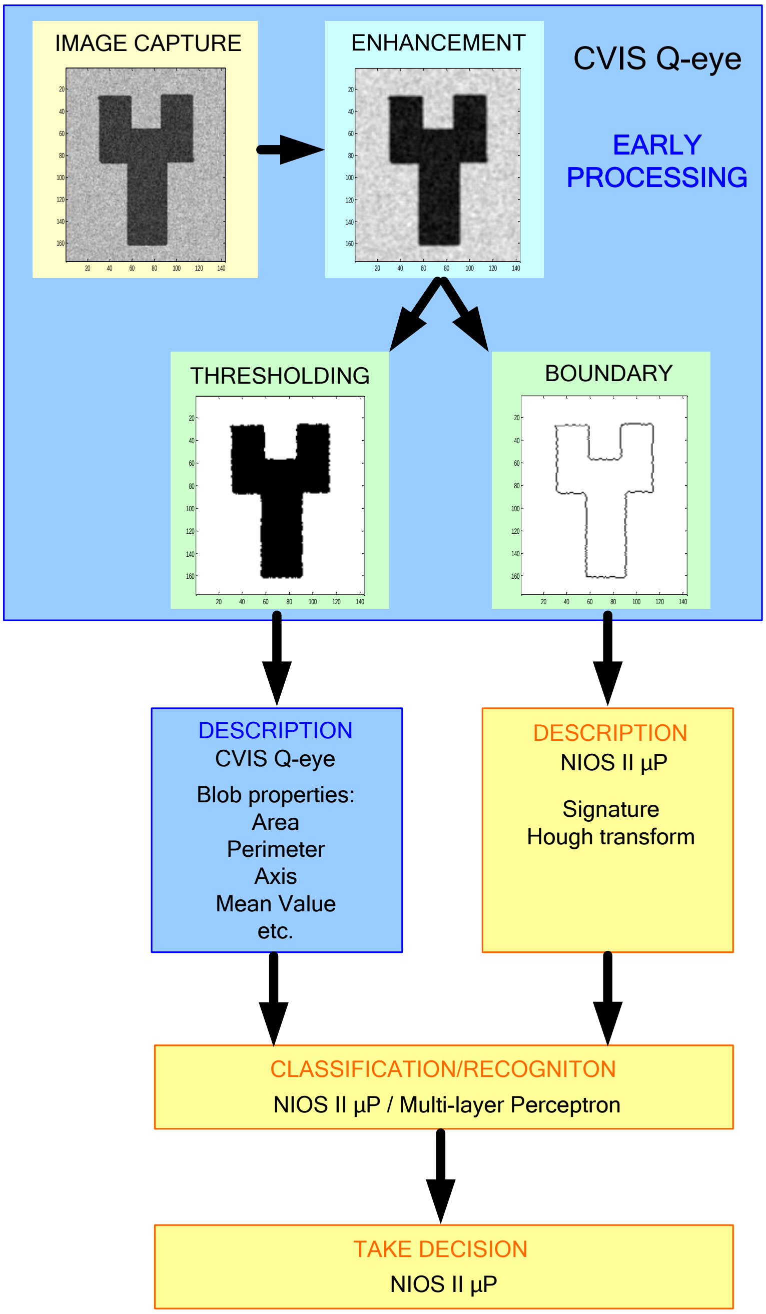


Figure 7. HIERARCHICAL PROCESSING.

In the particular case of Eye-RIS system, the Q-Eye develops the image acquisition. Thanks to the processing circuitry included at pixel level, a High Dynamic Range (HDR) sensing is implemented in the smart sensor. The captured image is processed by the Q-Eye analogue processing array developing a low pass filter in order to reduce the high frequency spatial noise (image enhancement stage). This filtered image is the input for the segmentation phase, which is carried out by the Q-Eye system applying a simple thresholding operation, followed by removal of isolated points and blobs in contact with the image borders. Each detected blob is reconstructed from a seed determined by the first point downloaded from Address Event and applying a morphological function of filling by flood.

The images resulting from segmentation are binary images for each detected blob or object. These images are the input to the description phase where the blob descriptors are calculated by the Q-Eye and NIOSII microprocessor. In this case, the Q-Eye is a co-processor developing for each characteristic a particular FPP (Focal Plane Processor) function to obtain the value of corresponding characteristic. Once the feature value is obtained, the Q-Eye controller sends an interruption to the NIOS-II microprocessor in order to transmit this value. NIOS-II microprocessor receives the value developing several operations and storing the result in the corresponding field of data structure. After this, the microprocessor performed a comparison between the extracted characteristic and the object reference, storing the similarity percentage and classifying the part as useful or useless. If the comparison result is useless then the part under analysis is discarded and new one is considered to obtain the features. After comparison, when the part is mark as useful, the Q-eye calculates the value of following characteristic, repeating the process until finish with all candidates. Therefore, after obtaining the descriptor value by the Q-Eye, the NIOS-II microprocessor performed the classification and recognition task. In this case, the blob descriptors have could be calculated efficiently by the Q-Eye by the nature of operators required for these tasks. However, in the case of using the boundary descriptors, the description and classification phase must be performed by the microprocessor completely, because the operations involved in these algorithms are not compatible with the architecture of massively parallel analog processor. Thus, the applications will be performed by the same vision system but requiring a more processing time and power consumption. The characteristics of algorithms to implement by the vision system determine their performances.

5.3. THE EYE-RIS_v1

Let us copy the block diagram of Figure 3-top for easier reading (see Figure 8). The Eye-RIS_v1 system is a compact vision system which captures images, enhances the quality of these images, processes the image flow in real time, interprets the information contained in the image flow and takes decisions based on the outcome of such interpretation. The Eye-RIS_v1 includes a CVIS sensor (the Q-Eye) at the front-end, a Digital Image Processor, a microprocessor, memories and I/O and communication ports.

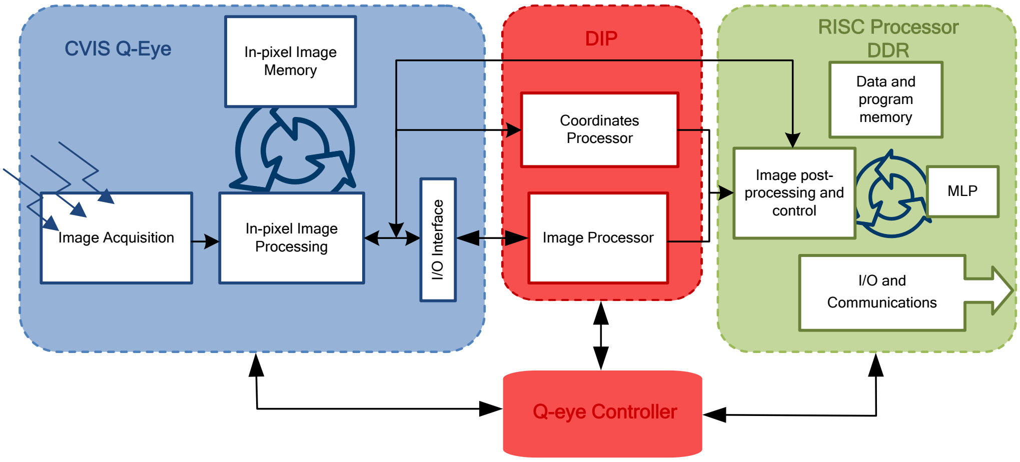


Figure 8. EYE-RIS VISION SYSTEM FUNCTIONAL DIAGRAM.

The CVIS at the front-end is described in Chapters 3 and 4 and is responsible for vision pre-processing. The Digital Image Processor (DIP) carries out linear and non-linear point-to-point image operations, geometric image transformations and momentum calculations. The Eye-RIS vision system includes a Multi-Layer Perceptron (MLP) block developed in software domain and used for classifying patterns.

The RISC processor is the ALTERA NIOS-II processor. It controls the operation of the whole vision system. Moreover, it analyses the information output by the Q-Eye, DIP and MLP and performs decision-making tasks required for actuation.

I/O and communication ports consists of a variety of digital input and output ports such as UART, PWM ports, GPIOs and Gigabit Ethernet.

Besides these blocks, a controller block is required to translate the high level instructions of NIOS-II microprocessor to the Q-Eye micro instruction format. It is called Q-Eye controller. It is also responsible for the generation of control signals associated to the Q-Eye Image Interface in a load/download image process (image data protocol).

The main blocks of Eye-RIS systems are described briefly in the following sections.

5.3.1. NIOS-II PROCESSOR

NIOS-II is a FPGA-synthesizable Digital Microprocessor. It is a configurable soft-core processor. “Soft-core” means the CPU core is offered in “soft” design form (not fixed in silicon), and can be targeted to any Altera FPGA family or system on-chip. The design of Eye-RIS has involved the definition and implementation of a methodology for synthesizing NIOS-II microprocessors on chips starting from the reference VHDL code generated by the ALTERA NIOS-II IDE software.

The main features of the Altera NIOS-II processor are summarized below:

- General-purpose RISC processor core.
- Harvard architecture.
- Full 32-bit instruction set, data path, and address space.
- 32 general-purpose registers.
- 32 external interrupt sources.
- Single-instruction 32 x 32 multiplies and divides producing a 32-bit result.
- Single-instruction barrel shifter.
- Access to a variety of on-chip peripherals, and interfaces to off-chip memories and peripherals through Avalon Bus.
- Hardware-assisted JTAG debug module enabling processor start, stop, step and trace.
- Instruction and Data Cache Memory.
- JTAG Debug Module.
- Exception and interrupt controller.

5.3.2. MEMORY MAP

The memory map of the Eye-RIS vision system (described in Figure 9) is organized into the following sections, namely:

- DDR2 Memory: Two sections of 64 Mbytes of DDR2 (RAM) memory are included within Eye-RIS. One section is devoted to store the application code and data. The other one is the Eye-RIS Image Memory – described in next section.
- On-chip RAM: One section of 2Kbytes of memory is included inside the FPGA. This area is reserved for storing exception addresses together with internal, critical, low-level routines.
- Reserved Section: Internal, reserved registers are mapped within this section.
- Peripherals: PIOs utilized for reading and/or writing the peripherals available in Eye-RIS.
- Serial Flash memory Interface: This section maps the interface used for controlling a serial flash memory included in the Eye-RIS vision system. Flash memory is non-volatile, and it is mainly used for storing programs that will run immediately after power-up plus user data. Additionally, lower addresses are reserved for storing critical internal information. This Flash memory allows the Eye-RIS vision system to work in stand-alone mode.
- Serial EEPROM Interface: Eye-RIS also includes an EEPROM, which is accessible to the user only in read-mode. This memory contains information like the serial number, the software version, default network configuration, etc.

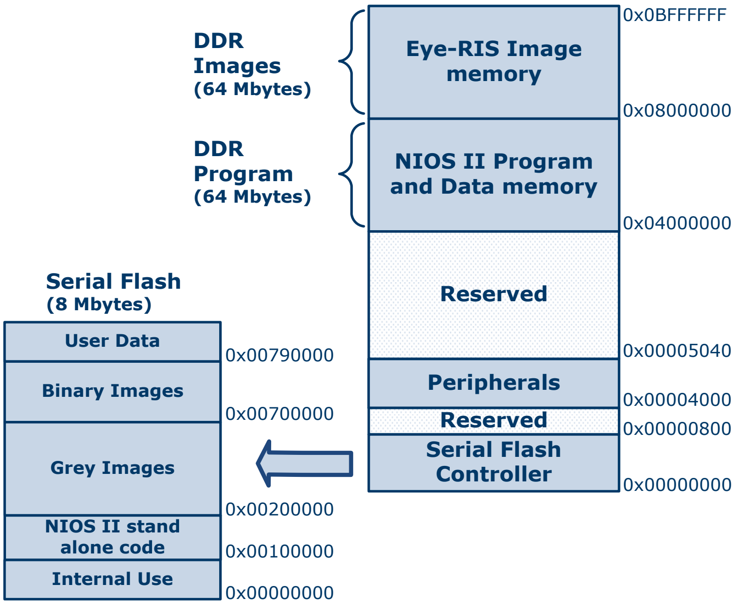


Figure 9. EYE-RIS SYSTEM MEMORY MAP.

5.3.2.1. EYE-RIS IMAGE MEMORY

The Eye-RIS Image Memory is dedicated to storing images. This memory is accessed both by the Q-Eye Controller and the NIOS-II processors, although not simultaneously. It is used for two main functionalities:

- **Long term images storage**: Due to the limited retention time of the LAMs in the Q-Eye, digital non-volatile memory might be needed to store images that do not change (or vary very slowly) during the execution of a certain algorithm.
- **Debugging purposes**: The user can store images coming from the PC in this memory so they can be loaded in real time to the Q-Eye chip through the I/O block. Likewise, the Q-Eye can download images to this memory for displaying them in the screen of your PC or for storing in your hard drive for later revision. Such image load/download is useful during the development of your vision application since it allows for debugging the algorithm with pre-acquired or artistically designed images before processing real-time images.

5.3.2.2. CACHE MEMORY

The NIOS-II architecture supports data and instruction cache memories. The cache memory resides on chip as an integral part of the NIOS-II processor core. The use of cache memory improves the average memory access time for NIOS-II processor as compared to the use of slow off-chip memory such as DDR2 SDRAM for program and data storage. The NIOS-II processor implemented within Eye-RIS contains both data and instruction cache memories, sized 2 Kbytes each.

The instruction and data caches are enabled perpetually at run-time, but methods are provided by software to bypass the data cache so that peripheral accesses do not return cached data. Cache management and cache coherency are handled by software. The NIOS-II instruction set provides instructions for cache management – for further information refers to [Anaf1v3] [NIOSII].

5.3.3. DIGITAL IMAGE PROCESSOR

The Eye-RIS vision system includes a Digital Image Processor (DIP), intended for two main objectives:

- Performing **complex digital image processing** operations not covered by the Q-Eye. For example, image products and divisions, geometric operations such rotation, scaling, etc.
- Accomplishing **iterative operations over grey-level images**, which cannot be realized by the Q-Eye due to its analogue nature.

The DIP block is described through the following sections.

5.3.3.1. FUNCTIONAL DESCRIPTION

The DIP block carries out three main groups of operations, which are described below:

- **Pixel to pixel operations** between grey images:

$$I_o = \alpha \cdot I_1 \circ \beta \cdot I_2 \qquad \textbf{Eq. 368}$$

  where α y β are constants (signed, 8 + 8 fixed point), “◦” is an operator, which can be addition, subtraction, multiplication or division, I1 and I2 are input images, and I0 the resulting image.

- **Second order momentum** calculation: $\sum_i x_i^2$, $\sum_i y_i^2$ and $\sum_{i,j} x_i \cdot y_j$ where x_i, y_j are pairs of coordinates of points of interest present in the images.
- **Projective transformation**, which is a geometrical transformation widely used in image processing. For example, particular cases of projective transformation are rotation, translation, scaling or shearing. This transformation is represented by the following expression:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \textbf{Eq. 369}$$

5.3.3.2. ARCHITECTURE

Figure 10 illustrates the architecture of the DIP. It is conceived to process images coming either from the Q-Eye or from the Eye-RIS Image Memory. Additionally, it comprises two pages of internal memory that may be used as intermediate storage during the execution of an image processing operation. Main features of this block include:

- 56 kbytes of internal memory able to contain up to two full grey-level images. As it was mentioned above, these memories are normally used to save partial results of image operations.
- Four ALUs capable of working concurrently.
- Capability to process images “on-the-fly”, during loads/downloads of images to/from the Q-Eye.
- Easy programming through the Q-Eye control unit.
- Three banks of registers for processing configuration.

5.3.4. DIGITAL INPUT/OUTPUT PORTS

5.3.4.1. GIGABIT ETHERNET PORT

This port conforms the IEEE 802.3 1000BASE-T standard for Gigabit Ethernet (GigE) over copper. GigE port carries two main connections within Eye-RIS:

- NIOS-II JTAG connection: NIOS-II JTAG Interface can be accessed through an Ethernet connection. Thus, programs can be downloaded to the NIOS-II from the PC. Additionally, NIOS-II can be fully debugged through this connection.
- Data connection: user data can be exchanged between the Eye-RIS and the PC

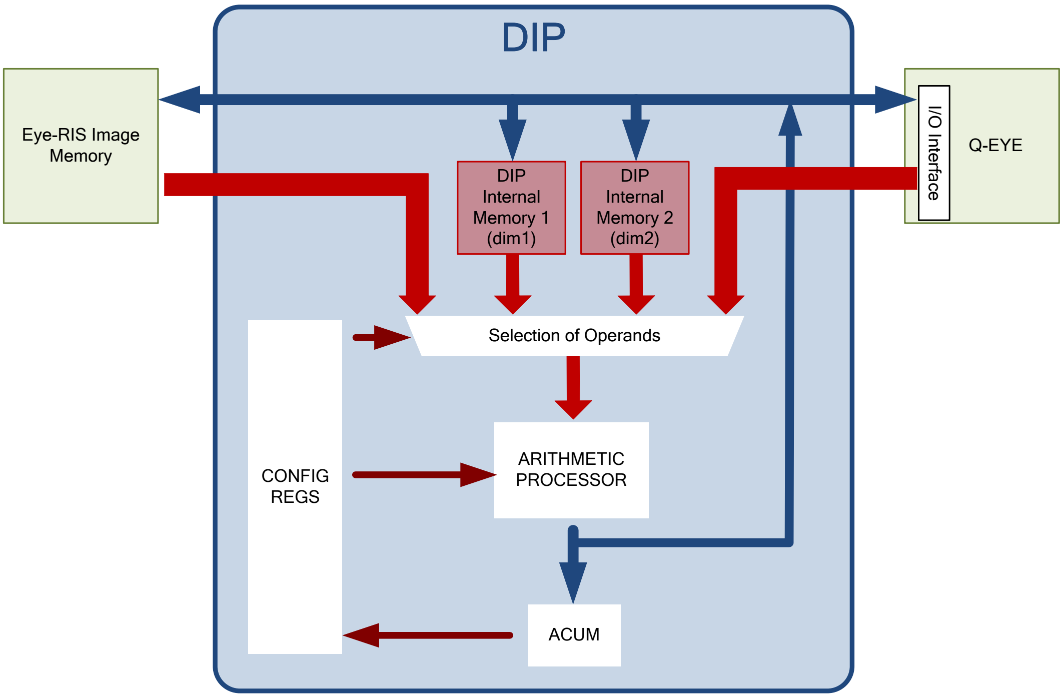


Figure 10. BLOCK DIAGRAM OF DIP.

5.3.4.2. UART

The embedded Universal Asynchronous Receiver/Transmitter peripheral implements a method to communicate serial character streams between the Eye-RIS vision system and an external device. Table.2 summarizes the features of the UART implemented within the Eye-RIS.

UART Features	
Name	UART1
Default Baud Rate	115200 (can be modified by software)
Parity	None
Data Bits	8
Stop Bits	1
Flow Control	None
Output voltage	3.3 LVCMOS

Table.2. Eye-RIS UART features.

The UART core uses a logic 0 for mark, and a logic 1 for space. It provides an active-high interrupt request (IRQ) output that can request an interrupt when new data has been received, or when the core is ready to transmit another character.

5.3.4.3. PWM

The Eye-RIS incorporates two PWM (Pulse-Width Modulation) ports. A PWM component outputs a square wave with a modulated duty cycle. A basic pulse-width waveform is shown in Figure 11.

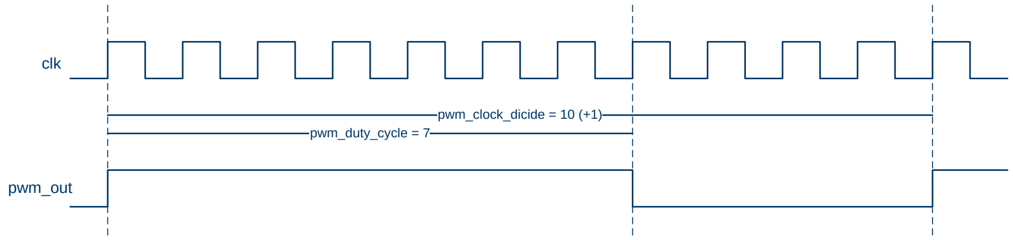


Figure 11. PWM WAVEFORM.

The PWM module is specified as follows:

- The task logic operates synchronously to a single clock.
- The task logic uses 32-bit counters to provide a suitable range of PWM periods and duty cycles.
- The host processor can halt the PWM output by using an enable control bit.
- PWM outputs are standard 3.3V LVCMOS, whose electrical features are shown in Table.3.

IO Current max	VIN min	VIN max	VIL max	VIH min	VOL max	VOH min
24 mA	-0,5 V	4 V	0,8 V	1,7 V	0,45 V	2,4 V

Table.3. Electrical features of the standard 3.3V LVCMOS.

5.3.4.4. GPIO

The Eye-RIS system provides 16 General Purpose Inputs/Outputs (GPIOs) pins. Each one can be configured individually and can also serve as an interrupt input.

The parallel input/output (PIO) core provides a memory-mapped interface between the NIOS-II processor and general-purpose I/O ports.

The PIO core provides easy I/O access to external devices in situations where a "bit banging" approach is sufficient. Some example uses are:

- Controlling LEDs
- Acquiring data from switches
- Controlling display devices
- Configuring and communicating with off-chip devices, such as application-specific standard products (ASSPs)

The core can be configured with inputs only, outputs only, or both inputs and outputs via a direction register.

The hardware logic is separate for reading and writing data register.

Reading the data register returns the value present on all ports, both ports configured as input and output. It is the programmer's responsibility to mask the read value to only read the input ports.

Writing data affects the value driven to the output ports (if any port is configured as output).

From the moment a port is configured as output, that port will deliver the data already present in the output data register, so it is important to correctly initialize the data register before configuring a port as output.

The PIO core interrupt request (IRQ) can assert an interrupt based on input signals. The PIO core can capture edges on its input ports. It can capture high-to-low transitions (falling edge) as well as low-to high transitions (rising edge). The PIO core can generate an IRQ when an input signal performs a high-to-low or a low-to-high transition.

Interrupts are individually maskable for each input port. The interrupt mask determines which input port can generate interrupts.

Like PWM port, GPIO signals are standard 3.3V LVCMOS, whose electrical features are shown in Table.3.

5.3.4.5. JTAG MODULE

The Eye-RIS v2.1 incorporates a JTAG module for loading and debugging programs. The NIOS-II architecture supports a JTAG debug module that provides on-chip emulation features to control the processor remotely from a host PC. PC-based software debugging tools communicate with the JTAG debug module and provide facilities, such as:

- Downloading programs to memory
- Starting and stopping execution
- Setting breakpoints and watch-points
- Analyzing registers and memories
- Collecting real-time execution trace data

The debug module connects through a JTAG standard interface, so that the Nios-II processor can be externally accessed using JTAG probes. On the processor side, the debug module connects to signals inside the processor core. The debug module has non-maskable control over the processor, and does not require a software stub linked into the application under test. All system resources visible to the processor in supervisor mode are available to the debug module. For trace data collection, the debug module stores trace data in memory either on-chip or in the debug probe.

5.3.5. Q-EYE CONTROLLER

The Q-Eye controller translates the high level instructions of NIOS II microprocessor to the *Q-Eye* micro instruction format [Anaf2v1a]. Furthermore, it is the responsible for the generation of control signals associated to the Q-Eye Image Interface in a load/download image process (image data protocol). The Q-Eye controller block consists of two main blocks:

- Set of **control state machines** which are responsible for the programming and image data protocols with Q-Eye.
- **Microprocessor** which carries out the micro-programs of Q-Eye system and communicates with the NIOS microprocessor.

The communication between the control state machines and the microprocessor of controller is carried out through a set of registers denoted by Interface Registers which structure is depicted in Figure 12. See also Figure 17 for the positioning of the interface registers within the control flow of Eye-RIS system. The controller can read and write data in the Q-Eye Programming Interface and apply FPP microinstructions stored in the Program Memory to the Q-Eye system using these Interface Registers, as it was described in Chapter 4.

The writing of data to the Q-Eye through the Programming Interface (EN_PB, RW_PB, ADD_PB<11:0>, DAT_PB<11:0>, ENG) is carried out writing this data in register 2 (DAT_PB_OUT) and then writing the address where the data must be introduced in register 0 (ADD_WR_PB). The writing access to the Q-Eye system is applied automatically by the controller when the register 0 is written.

The reading of data from the Q-Eye through the Programming Interface is launched by the controller automatically when the register 1 (ADD_RD_PB) is written. After several system clock cycles the corresponding data can be read in register 16 (DAT_PB_IN).

A FPP microinstruction is applied to the Q-Eye system by the controller writing in register 0 (ADD_WR_PB) the address corresponding to PBCTRL register or PBMASK register, then the instruction is applied automatically. In this case the content of register 2 is irrelevant.

Registers 3, 4, 5, and 6 (INST_CODE, PAR_1, PAR_2, END) active the state machines which control the load/download processes through the Image Interface of Q-Eye system. The implemented functions which control the communication protocols with Q-Eye system are load-grey, download-grey, load-bin, download-bin, ae-direct and ae-inverse. These functions control the input and output of images in the corresponding formats: grey images, binary images and coordinates.

<table>
<tr><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">ADD_WR_PB[11:0]</td></tr>
<tr><td colspan="12">ADD_RD_PB[11:0]</td></tr>
<tr><td colspan="2">N.U.</td><td colspan="10">DAT_PB_OUT[9:0]</td></tr>
<tr><td colspan="5">N.U.</td><td>BIT3</td><td>BIT2</td><td>BIT1</td><td>START</td><td colspan="3">INST_CODE[2:0]</td></tr>
<tr><td colspan="12">PAR1[11:0]</td></tr>
<tr><td colspan="12">PAR2[11:0]</td></tr>
<tr><td colspan="11">N.U.</td><td>END</td></tr>
<tr><td colspan="12">NUM_POINTS_AED[11:0]</td></tr>
<tr><td colspan="4">N.U.</td><td colspan="8">ROW0_AED[7:0]</td></tr>
<tr><td colspan="4">N.U.</td><td colspan="8">COL0_AED[7:0]</td></tr>
<tr><td colspan="4">N.U.</td><td colspan="8">ROW1_AED[7:0]</td></tr>
<tr><td colspan="4">N.U.</td><td colspan="8">COL1_AED[7:0]</td></tr>
<tr><td colspan="2">N.U.</td><td>COLLR</td><td>ROWUD</td><td colspan="8">ROWINIT[7:0]</td></tr>
<tr><td colspan="4">N.U.</td><td colspan="8">ROWEND[7:0]</td></tr>
<tr><td colspan="12">DAT_PERIPHERAL[11:0]</td></tr>
<tr><td>R/nW</td><td colspan="11">ADD_PERIPHERAL[11:0]</td></tr>
<tr><td colspan="2">N.U.</td><td colspan="10">DAT_PB_IN[9:0]</td></tr>
</table>

Figure 12. **INTERFACE REGISTERS STRUCTURE.**

In register 3, bits INST_CODE<2:0> select the communication function to execute, Table.4 describes the codification.

BIT1 and BIT2 are bits to configure some parameters of data communication functions together with the registers 4 and 5 (PAR_1, PAR_2).

The START bit is a control bit. When the START bit is set to 1 the corresponding control machine of selected function is launched. When the control machine has finished the load/download process, the bit END of register 6 is set to 1 by the controller to indicate the end of process.

COMMUNICATION FUNCTION	INST_CODE[2:0]
LOAD GREY	000
LOAD BIN	001
DOWNLOAD GREY	010
DOWNLOAD BIN	011
ADDRESS EVENT DIRECT	100
ADDRESS EVENT INVERSE	101

Table.4. Communication functions code.

Parameters of Q-Eye Image Interface are stored in registers 12 and 13. Through registers 14 and 15 (DAT_PERIPHERAL, ADD_PERIPHERAL), the Q-Eye controller can access to external peripherals. The Image Communication Functions will be described briefly in the following subsections.

5.3.5.1. LOAD/DOWNLOAD GREY

These functions developed the load and download processes of grey level images. Table.5 presents the configuration parameters.

PARAMETER	DESCRIPTION
PAR_1[11:0]	Number of memory page where the image will be stored or taken
BIT1	External or Internal memory is used in load/download process

BIT2	Inversion of image

Table.5. Parameters of grey image Load/Download processes.

5.3.5.2. LOAD/DOWNLOAD BIN

These functions developed the load and download processes of binary images. Table.6 describes the configuration parameters.

PARAMETER	DESCRIPTION
PAR_1[11:0]	Number of memory page where the image will be stored or taken
BIT1	External or Internal memory is used in load/download process
BIT2	Inversion of image

Table.6. Parameters of binary image Load/Download processes.

5.3.5.3. ADDRESS EVENT DIRECT/INVERSE

These functions control the load/download processes of coordinates associated to active points in the Q-Eye processing array. Table.7 defines the configuration parameters.

PARAMETER	DESCRIPTION
PAR_1[11:0]	Number of memory page where the image will be stored or taken
PAR_2[11:0]	Number of points to load or download (zero means process all points given by the Q-Eye system)
BIT1	External or Internal memory is used in load/download process
BIT2	Set to 1 actives a special download mode

Table.7. Parameters of Address Event processes.

The data structure in the memory is described in Figure 13. Each coordinate is composed by two numbers (row and column). Two points are stored in a word memory of 32 bits (POINT0 is processed first and POINT1 is second).

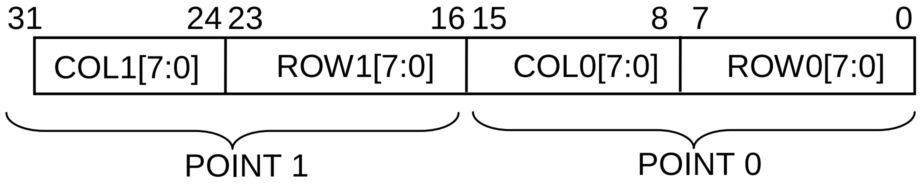


Figure 13. STRUCTURE OF ADDRESS EVENT DATA.

In the special download mode activated by configuration bit BIT2, only two points are downloaded in the registers 8, 9, 10 and 11 (ROW0_AED, COL0_AED, ROW1_AED, COL1_AED).

5.3.6. NIOS-Q-EYE CONTROLLER COMMUNICATION

During the execution of a vision application by the Eye-RIS system, a program is executed in the NIOS microprocessor. This program makes call to the Q-Eye controller in order to execute FPP microinstructions for Q-Eye system. Therefore, a communication channel between both blocks, NIOS and Q-Eye controller, is necessary, for control tasks and data exchange (programs and images) – see Figure 14.

The NIOS-II microprocessor can access in writing and reading mode to several areas of Q-Eye controller related with the execution of FPP microinstruction through the called Configuration Interface. The address map of NIOS has parts associated to the Q-Eye controller:

- Internal registers of Q-Eye system
- Q-Eye controller registers
- Internal memory of Q-Eye controller
- Communication registers

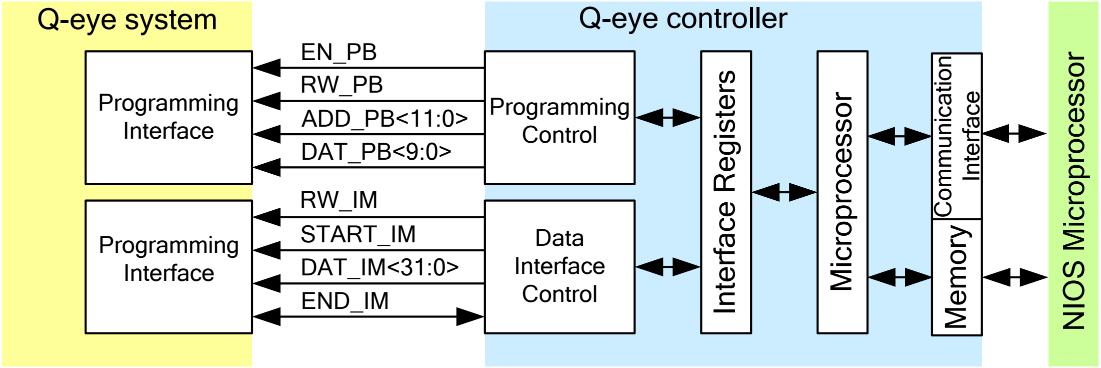


Figure 14. CONTROL FLOW OF EYE-RIS SYSTEM.

The NIOS microprocessor can reset the Q-Eye controller through one of its registers with Input/Output Ports (PIOs). The communication registers allows the NIOS microprocessor to send instructions to the Q-Eye controller and to receive information. Through these registers, the NIOS microprocessor can send a start instruction for the execution of a micro-program, send the internal address memory of controller from the micro-program will start, receive information about the status of micro-program execution, send control signals to the Q-Eye system (ENG, CLK_CTRL, RESET), receive information related with controller FIFOs, etc.

5.3.7. CONTROLLER MEMORY

5.3.7.1. DATA MEMORY

The Q-Eye controller works with an internal data memory consists of 4K words of 12 bits. This data memory contents 32 general purpose registers and the rest of memory corresponds to generic data (global variables, tables, stack, etc.).

The internal memory is used by the controller instructions to take parameters or give a resulting data. This memory can be written and read by the NIOS microprocessor during the time intervals when the controller does not execute programs.

The data memory of controller can be increased using an external memory which is accessed by special instructions that transfer the data between both memories. The operation is carried out using the internal memory.

5.3.7.2. PROGRAM MEMORY

The instructions to execute by the controller are stored in the program memory. This memory can have a variable size, until 4096 pages of 4096 instructions each page. The instructions consists of 24 bits words.

The program memory is read by the controller and written and read by the NIOS when the Q-Eye controller does not execute any micro-program.

The Q-Eye controller works with this program memory through a register of page selection (PROG_PAGE special register in data memory) and a counter program of 12 bits which indexes the selected page.

5.3.8. CONTROLLER MICROPROCESSOR

The microprocessor of Q-Eye controller is responsible for the micro-program execution. This micro-program is composed by the FPP (Focal Plane Processor) instructions which codified the operations of Processing Array. Table.8 describes the main instructions of microprocessor contained in the Q-Eye controller. The Q-Eye controller has 12 generic lines of interruption. Each interruption has associated an interruption vector which indicates the start address of interruption routine, a state bit, enable bit and priority code. Through the registers 14 and 15 (DAT_PERIPHERAL, ADD_PERIPHERAL) contained in the Interface registers the controller configures and controls six timers – see Figure 12.

INSTRUCTION	DESCRIPTION
NOP	No operation
RETI	Interruption routine return
JUMP	Conditional and unconditional jump
CALL	Sub-routine call
END	Program end
RET	Sub-routine return
PUSH	Introduction of parameter into the stack
POP	Extraction of last data in the stack and storing in data memory
NOT	Not operation
SET	Introduction of value into the memory data
INC	Increment operation
DEC	Decrement operation
COMP	Comparison operation
AND	AND operation
OR	OR operation
ADDC	Addition operation
SUBC	Subtraction operation
SHIFTR	Right shifting operation
SHIFTL	Left shifting operation
DELAY	Wait operation
LOAD	Transfer data from external memory to the internal memory
STORE	Transfer data from internal memory to the external memory
WR_INTREG	Write a data into an Interface register
RD_INTREG	Read a data from an Interface register

Table.8. Microprocessor instructions of Q-eye controller.

5.4. THE EYE-RIS_V2

Figure 15 is a copy of the block diagram of Figure 3-bottom that has been also inserted here for easier reading. It shows a block diagram for the system Eye-RIS_v2.

As shown in the figure, the Eye-RIS_v2 consists of the so-called Eye-RIS_VSoC and some external memories. The Eye-RIS_VSoC embeds the Q-Eye, the DIP and the Q-Eye controller on a single silicon substrate. The Eye-RIS_VSoC is hence a much more powerful CVIS than the Q-Eye because it is capable of performing both pre-processing and post-processing tasks. Figure 16 shows the placement of the Eye-RIS_VSoC sub-blocks on a silicon substrate corresponding to a standard CMOS 0.18μm technology with one poly and six metals.

Compactness level achieved by the Eye-RIS_v2 system is larger than the corresponding level of the Eye-RIS_v1. Indeed, the implementation of an external serial flash interface in the Eye-RIS_VSoC allows the Eye-RIS v2 to work in *stand-alone* mode without requiring external hosts. Obviously, the fact that the RISC microprocessor is implemented using the same thick-line technology that the photo-sensors limits the operating frequency. However, this is not particularly critical since post-processing is less data intensive than pre-processing. Also, operation at lower frequencies means smaller substrate noise. Finally, design is easier because transistor lengths are larger than in main-stream VLSI digital technologies.

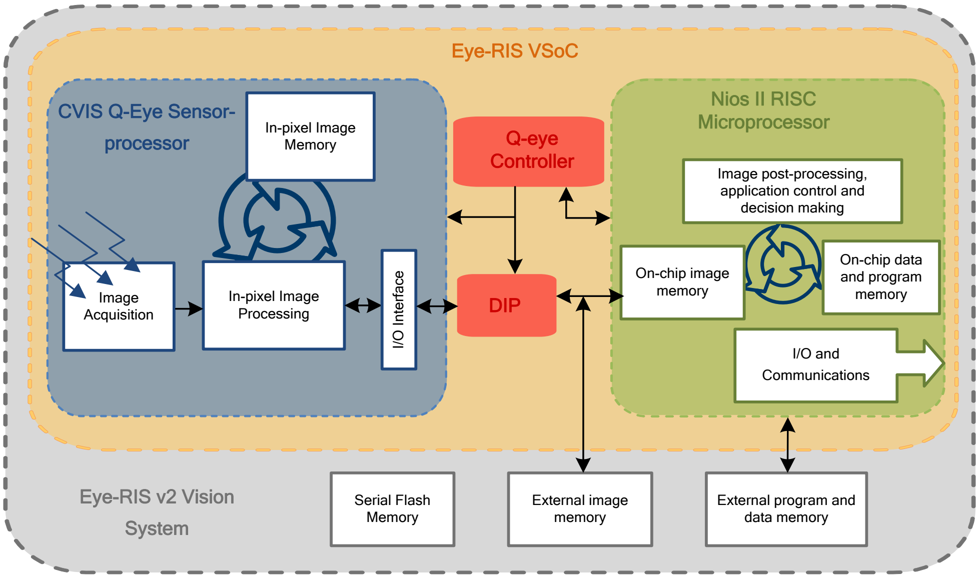

Eye-RIS VSoC
CVIS Q-Eye Sensor-processor
In-pixel Image Memory
Image Acquisition
In-pixel Image Processing
I/O Interface
Q-eye Controller
DIP
Nios II RISC Microprocessor
Image post-processing, application control and decision making
On-chip image memory
On-chip data and program memory
I/O and Communications
Eye-RIS v2 Vision System
Serial Flash Memory
External image memory
External program and data memory

Figure 15. EYE-RIS_V2 BLOCK DIAGRAM.

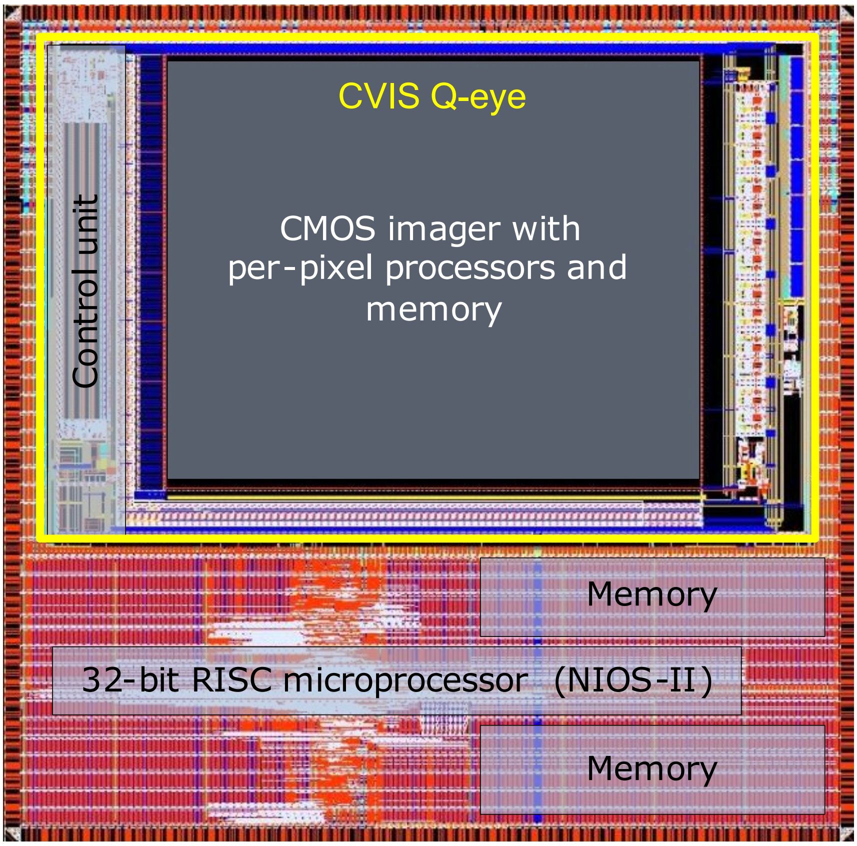

CVIS Q-eye
CMOS imager with per-pixel processors and memory
Control unit
Memory
32-bit RISC microprocessor (NIOS-II)
Memory

Figure 16. EYE-RIS VISION SYSTEM ON CHIP LAYOUT.

5.4.1. ARCHITECTURE AND CONTROL OVERVIEW OF TASK SCHEDULING

Figure 17 provides larger details of the Eye-RIS_VSoC architecture [Anaf2v1b]. Its design and implementation face similar issues than other embedded system. For example, restrictions regarding available silicon area, pose limitations on the resources that will be implemented in hardware and will be available for application developments. Data and program memories are one of these resources, whose dimension is crucial in embedded vision systems. The Eye-RIS VSoC contains on-chip RAMs composed by two sections. One section of 256 Kbytes is devoted to store the program code and data. The other section of 128 Kbytes is used for storing images. Also, the memory map of VSoC is composed by a reserved section where internal registers are mapped and peripheral section dedicated to PIOs used for reading and/or writing the peripherals available in Eye-RIS_v2.

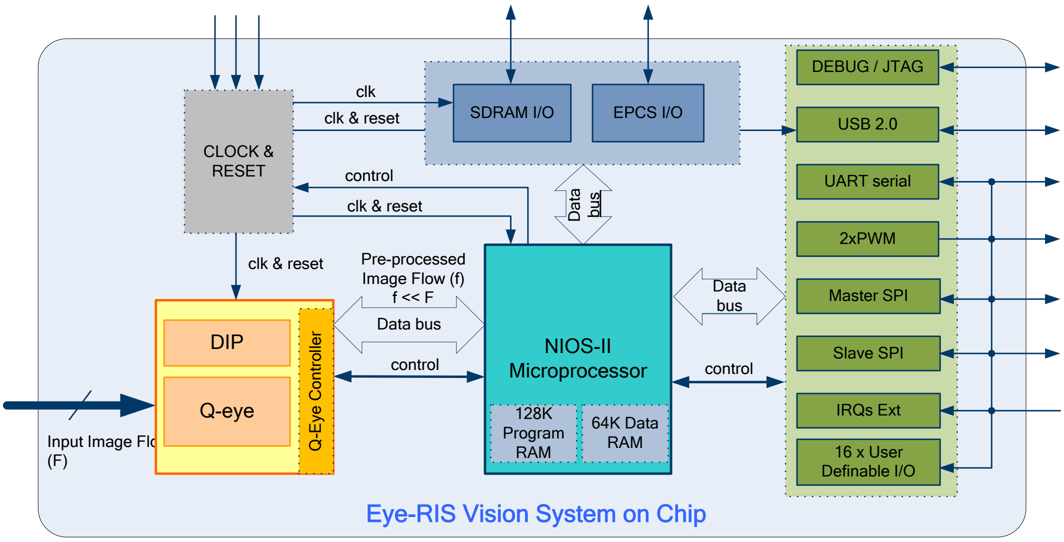


Figure 17. EYE-RIS_VSOC ARCHITECTURE.

Eye-RIS_v2 includes memories outside the VSoC, namely SDRAM memories composed by two sections of 16 Mbytes, one dedicated to store the application code and data and other is the image memory. In addition, a Serial Flash memory is included. So, there is a section in the memory map which maps the interface used for controlling this serial flash memory. The Flash memory is non-volatile, and it is mainly used for storing program that will run immediately after power-up. The memory map is described in Figure 18 the size of on-chip memories can be compared with the size of outside chip memory, being confirmed that the amount of memory inside chip has to be considerably less due to physical size constraints.

The Eye-RIS_VSoC is characterized by a co-processing architecture, where the microprocessor works in conjunction with the dedicated hardware (Q-Eye and DIP) to deliver a specific application. In this case, the application-specific hardware can operate concurrently with software execution performed by NIOS-II microprocessor. In this context, critical points to be treated in the definition of task scheduling during the design phase of a vision system and application developments are:

- The concurrent operation of dedicated processors (Q-Eye and DIP) and the general-purpose microprocessor (NIOS-II).
- Timing constraints which are common in high-speed applications.
- Task execution with limited resources.

As explained in Chapter 4, the Q-Eye is a programmable sensor-processor capable to execute FPP microinstructions. This set of micro instructions give arise to the programming language of Q-Eye, denoted by CFPP code [Anaf2v1a]. From a software point of view, this language has involved the development of a specific compiler. CFPP code is structured according to the two concepts described below:

- **Functions**. A function in CFPP code is the same as a function in any other kind of programming language. It is a set of statements of CFPP code that implements a certain algorithm and can optionally receive arguments and/or return a value as a result of its execution.
- **Sections**. A section in CFPP code is similar to a function. But the main difference is that it can be called by an external host (in this case, the NIOS-II processor). This makes sections the entry points from the host. They can be seen as the points through which the NIOS-II can access the Q-Eye services. Sections cannot call other sections and cannot

be called from CFPP code. Therefore, sections are only called by the NIOS-II microprocessor.

The execution flow in Eye-RIS systems is controlled by the NIOS-II microprocessor. This means that the Q-Eye and DIP operate under direct control of the NIOS-II microprocessor through the Q-Eye controller. The execution flow is defined by the code executed within the NIOS-II microprocessor. It decides which section is being executed at all times, although inside a section the flow is determined by its CFPP code.

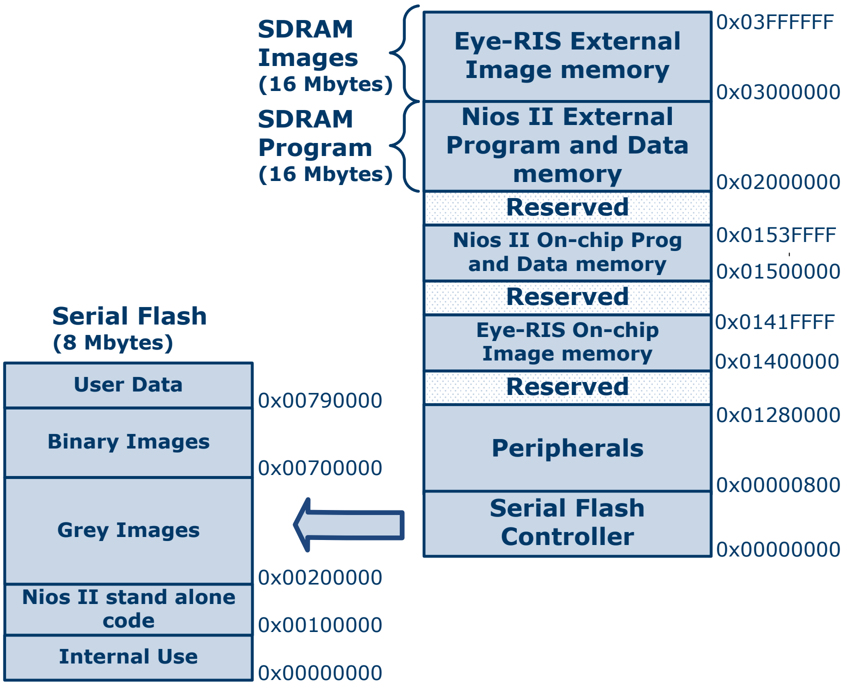


Figure 18. EYE-RIS_V2 MEMORY MAP.

The NIOS-II microprocessor does not have to stall while the Q-Eye is operational in order to achieve the maximum degree of parallelism in the processing for high speed-applications. When the microprocessor does not stall, some mechanism of synchronization is necessary. This synchronization can be implemented through proper interruption procedures. The use of interruptions between NIOS and Q-Eye controller allows synchronizing both processors while working in parallel. For example, a special function called *nioscall* has been developed for FPP code. With this function in the FPP code, the Q-Eye controller can send an interruption to the NIOS-II processor, which interrupts the task in progress to develop a new task required by the Q-Eye controller. During the time required for NIOS to perform this new task, the FPP and its controller remain waiting until NIOS automatically returns the task result to Q-Eye controller and allows the FPP to continue its work. At this point the NIOS processor goes on with the pre-planned tasks. Figure 19 depicts the procedure described previously.

In case interruptions are not employed, parallelism is achieved in different way. Namely, the NIOS launches the execution of a particular task in the FPP Q-Eye using an asynchronous section which allows the NIOS to continue execution of its code. Later, when the NIOS-II has completed its task or estimates that FPP has finished the algorithm, the NIOS takes the result of FPP algorithm. Accordingly, to ensure proper synchronization in this procedure the application developer must have a thorough knowledge of the execution time of the FPP task, in order to know when it has finished, collect the results and launch new tasks. It is very difficult for the application developer to optimize the parallelism level in co-processing with this method.

From user point of view, one of the most important characteristics of a vision system is the **application development software**. The software architecture is closely related to the architecture of the associated hardware. The application development environment is a dedicated piece of software conceived for writing, compiling and debugging programs in order to facilitate the vision application development. Usually, the development environment has a set of libraries including functions which avoid that the user has to work at instruction level with the system processors.

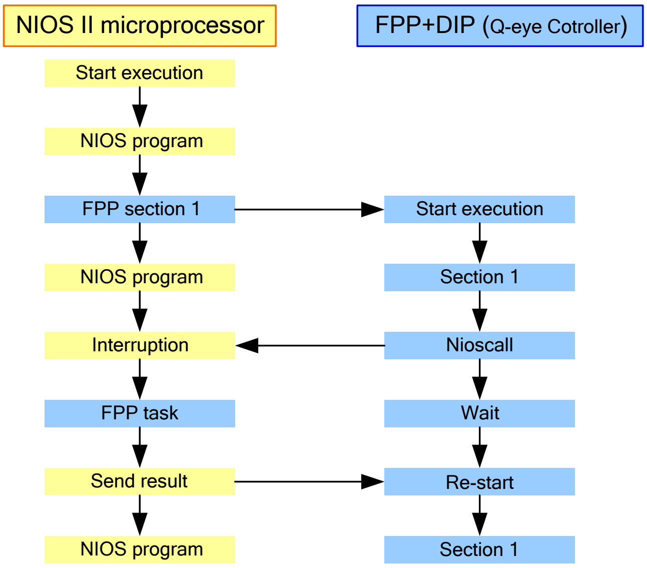


Figure 19. PARALLEL OPERATION BASED ON INTERRUPTIONS MANAGEMENT.

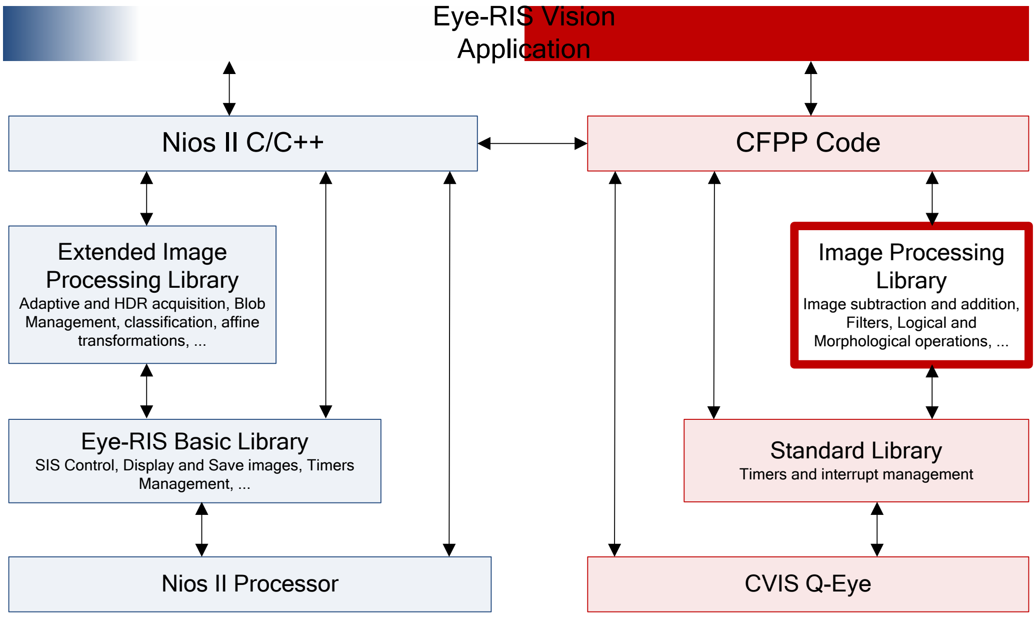

Eye-RIS Vision Application
Nios II C/C++
CFPP Code
Extended Image Processing Library
Adaptive and HDR acquisition, Blob Management, classification, affine transformations, ...
Image Processing Library
Image subtraction and addition, Filters, Logical and Morphological operations, ...
Eye-RIS Basic Library
SIS Control, Display and Save images, Timers Management, ...
Standard Library
Timers and interrupt management
Nios II Processor
CVIS Q-Eye

Figure 20. SOFTWARE ARCHITECTURE OF EYE-RIS SYSTEM.

In the case of Eye-RIS system, there are two processors working in parallel. Therefore, the system is programmed combining two programming languages: C/C++ code for the NIOS microprocessor and CFPP code for Q-Eye. Each operation or task belonging to early-processing has associated a set of Q-Eye microinstructions which constitute a CFPP function. Among the CFPP functions there are some functions or processing operations (reading/writing memories, arithmetic operation, diffusion, combinational operation, etc.) from which image processing functions (filters, morphological operations, adaptive thresholding, etc.) can be developed. Likewise, the post-processing tasks are C/C++ functions which will be developed by the microprocessor.

Also, there are low level tasks (instruction level) for the NIOS microprocessor such as execution control, data transmission from PC, IRQ management, timing functions, communications ports operation, etc. and for the Q-Eye controller microprocessor (IRQ management and timing functions) which are composed by a set of instructions. The set of instructions associated to each low level task has been implemented by a specific function, simplifying the programming for the user.

All these functions that define the image processing tasks are organized into libraries and facilitate the programming of vision system. Although, The Eye-RIS system can be programmed using plain CFPP and C++ codes instead of using this library functions. The library organization is hierarchical, following the abstraction level associated to each task. Figure 20 depicts the programming model of Eye-RIS systems.

5.4.2. PRE-PROCESSING AND POST-PROCESSING IMPLEMENTATION ISSUES

As explained in different parts of this monograph, the architecture of the Q-Eye is specifically conceived to execute pre-processing vision tasks involving regular computational flows consisting of a set of operations that are performed repeatedly. However, the correspondence between dedicated hardware and pre-processing algorithms is not so directly if the algorithm must adapt to the application conditions. Examples of these adaptive algorithms are found in the last section of this chapter. Adaptive algorithms are characterized by irregular flow involving data-dependent decisions which are unpredictable. Handling this unpredictability calls for flexibility and programmability of the processor. However, both the Q-Eye and the DIP are pure *co-processors* that perform only mathematical operations and that cannot make decisions themselves. Therefore, a microprocessor or CPU must be involved in the implementation of such algorithms. In the case of Eye-RIS systems, the NIOS microprocessor is involved in these adaptive algorithms.

Seeking for maximum parallelism in pre-processing and post-processing tasks, which minimizes the processing time for high speed applications, a microprocessor has been introduced in the Q-Eye controller (read corresponding sections of the Eye-RIS_v1 description in this Chapter). This way, the application-specific hardware composed by the Q-Eye, the DIP and the Q-Eye controller can operate in parallel with the NIOS, thus optimizing the processing time.

For very high speed applications, not only the parallelism between pre-processing and post-processing is important, but to get parallelism or concurrency between the sensing process and processing is critical. A pipeline approach, where the optical sensor works in parallel with the circuitry responsible for early-processing and the microprocessor in charge of post-processing tasks, achieves maximum frame rate during vision system operation. Figure 21describes graphically this concept.

Sensing #1	Sensing #2	Sensing #3	Sensing #4
	Pre-processing #1	Pre-processing #2	Pre-processing #3
		Post-processing #1	Post-processing #2

Figure 21. PIPELINE ARCHITECTURE.

In a pipeline structure where several processors are working in parallel with limited resources, the task scheduling becomes a critical topic. Regarding this issue, the IRQ (Interruption ReQuest) management gets to play an important role in synchronizing different process running in parallel. An interesting example of this fact is the parallelization in the smart sensor (Q-Eye) of sensing and pre-processing performed by the circuitry included in the pixel. This functionality is achieved thanks to the IRQ management performed by the microcontroller included in the Q-Eye controller. Several functions in CFPP code have been developed for the Q-Eye controller in order to implement interrupt-based acquisition of images. This set of functions allows parallelizing the acquisition of images with other processing in the Q-Eye. Figure 22 describes the interrupt-based acquisition.

Once the interrupt-based acquisition mode is enabled by a CFPP function, the exposition is configured and launched by another one. At this instant, the sensor starts the acquisition and immediately after that, the execution flow continues with the next instruction. In this way, acquisition and processing can be performed in parallel. Once the specified exposure time has elapsed, the acquisition of image is finish and other CFPP function synchronizes the end of this exposition and the processing operations that are being carried out in parallel. The latency required to attend an interrupt and save the acquired image slightly depends on the CFPP code being executed.

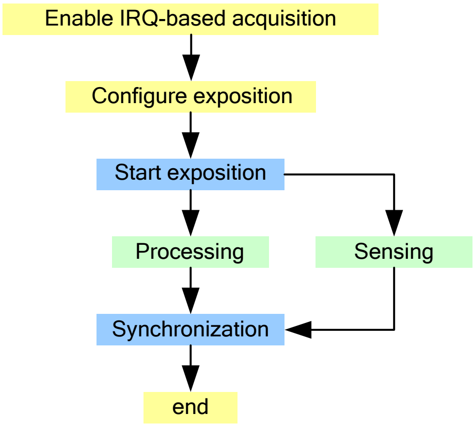


Figure 22. INTERRUPTION-BASED ACQUISITION.

5.4.3. EXTERNAL INTERFACES

Like any embedded vision system, communication interfaces are important system characteristics that application developers must consider.The Eye-RIS_VSoC includes the following Input/Output Interfaces:

- A JTAG Debug Module to load and execute programs on NIOS-II microprocessor, pause them, analyze trace data (internal registers and memories), etc.
- An USB 2.0 port mainly used for fast exchange of images between the Eye-RIS vision system and PC.
- A 32MB SDRAM interface to expand the memory map.
- An external Serial Flash interface.
- A set of I/O peripherals:
    - 1 UART
    - 1 SPI (Serial Port Interface) Master
    - 1 SPI slave
    - 2 PWMs
    - 1 Rising Edge Interrupt input pin
    - 1 Falling Edge Interrupt input pin
    - GPIO[15:0], 16 software-programmable general purpose digital I/O lines.

Due to the limited number of pins available on the package in order to reduce cost, the UART, SPI, PWM and interrupt ports are multiplexed with the GPIO.

Those interfaces that have not been presented before will be described in following subsections.

5.4.3.1. USB 2.0

A USB 2.0 Interface is connected to the PC through a QuickUSB module [USBa-c]. QuickUSB is based on Cypress EZ-USB FX2LP microcontroller [Anaf2v1b]. The interface implemented in the Eye-RIS_VSoC in order to communicate with Cypress EZ-USB FX2LP microcontroller is described in Figure 23. An additional function of USB interface is to feed the system when connected to a host that provides power (for example a PC).

5.4.3.2. EXTERNAL SDRAM INTERFACE

The Eye-RIS_VSoC has a dedicated interface to connect a 32Mbytes external SDRAM. This memory is dedicated to store the application code running on NIOS-II and images. The lower 16Mbytes are reserved to store the NIOS-II code whereas the upper 16Mbytes are available to store up to 440 grey images and 512 binary images, see Figure 24.

Due to the limited amount of memory included in the Eye-RIS_VSoC, an external SDRAM interface has been implemented in order to provide the possibility to increase the amount of available memory for the system that includes the VSoC. Table.9 shows the main characteristics of this interface.

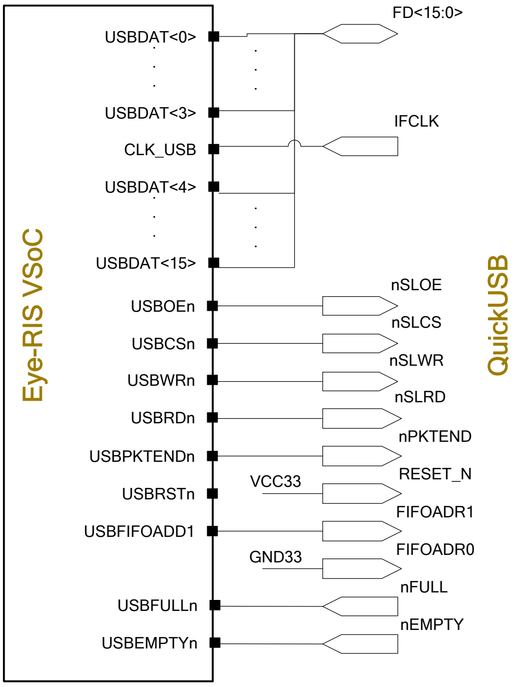


Figure 23. INTERFACE BETWEEN EYE-RIS VSOC AND QUICK USB MODULE.

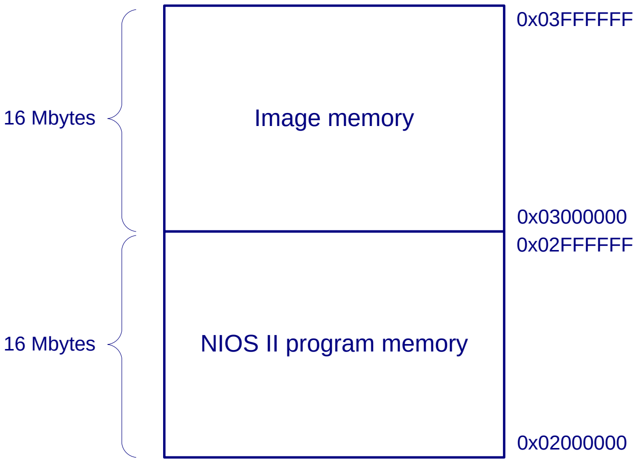


Figure 24. SDRAM MEMORY MAP.

NAME	VALUE
Data bus width	32 bits
Chip Selects	1
Banks	4
Row Address Width	12
Column Address Width	9
CAS Latency Cycles	3
Initialization refresh cycles	2
Issue one refresh command every	15.625 us
Delay after power up, before initialization	100 us

Duration of refresh command (t_rfc)	70 ns
Duration of precharge command (t_rp)	20 ns
Access Time (t_ac)	5.5 ns
Write recovery time (t_wr, No auto precharge)	14 ns

Table.9. SDRAM Interface parameters.

5.4.3.3. External Serial Flash Interface (EPCS I/O)

Additionally, the Eye-RIS_VSoC contains an interface to a non-volatile flash memory, which consists of a serial flash memory with a standard SPI interface and has a size of 8 Mbytes. This memory block is employed when the Eye-RIS system works in stand-alone mode. Once an application is ready, it can be stored on the flash memory and the system programmed to start running such code immediately after power-up.

5.4.3.4. Master & Slave SPI Ports

The Eye-RIS_VSoC includes a master SPI port and a slave SPI port. The master port is intended for accessing serial flash memory, or other peripheral devices. The slave port is intended as an alternative interface to a host micro-controller. Table.10 and Table.11 show the characteristics of SPI ports.

SPI Master Features	
Clk	17Mhz
Clock Polarity	0
Clock Phase	0
Data Bits	8 (MSB first)
Chipselect signals	1 (SSn)

Table.10. SPI Master characteristics.

SPI Slave Features	
Clock Polarity	0
Clock Phase	0
Data Bits	8 (MSB first)

Table.11. SPI Slave characteristics.

5.4.3.5. UART

The embedded universal asynchronous receiver/transmitter peripheral implements a method to communicate serial character streams between the Eye-RIS_VSoC and an external device.

The implemented UART provides adjustable baud and RTS/CTS flow control signals. The baud rate is derived from the system clock using a software-controlled 16-bit divisor. With a 100MHz system clock, baud rates from 100MHz to 1.5KHz may be programmed. Baud Rate is configurable, but the rest of the features parity, data length and stop bits, are fixed.

Table.12 summarizes the features of the UART implemented within the Eye-RIS_VSoC. The UART core uses a logic 0 for mark, and a logic 1 for space. It provides an active-high interrupt request (IRQ) output that can request an interrupt when new data has been received, or when the core is ready to transmit another character.

UART voltage levels are LVTTL compatible (0-3.3V) so it must be noticed that when connecting the UART to a standard RS-232 port, an external transceiver (i.e. the MAX3232), is needed in order to comply with the voltage levels defined in the standard.

UART FEATURES	
Name	UART1
Default Baud Rate	115200 (can be modified by software)
Parity	None
Data Bits	8
Stop Bits	1
Flow Control	CTS/RTS
Output voltage	3.3 LVCMOS

Table.12. UART characteristics.

5.4.3.6. GPIO

The General Purpose Inputs/Outputs interface implemented in Eye-RIS_VSoC has similar characteristics to the GPIO implemented in Eye-RIS_v1. In the case of Eye-RIS_VSoC due to the limited number of package pins, these I/O lines do not have dedicated pins. Instead, they are multiplexed with other signal pins, under software control. Each GPIO line can be configured as GPIO or other peripheral, individually. When acts as a peripheral, the written value to GPIO data register is ‘0’.

There are 3 registers associated to GPIO:

- **PIO_DATA_GPIO<15:0>:** Register used to send or receive data between Nios processor and the pins.
- **PIO_ DIR_GPIO<15:0>**: Controls the direction of each GPIO while they are in GPIO mode:
  - When set to ‘0’, pins are treated as outputs, and content of PIO_DATA_GPIO is sent to the pins.
  - When set to ‘1’, pins are treated as inputs and pins value is copied to PIO_DATA_GPIO.
- **PIO_MUX_GPIO<7:0>**: Selects the Alternate Functionality. Each bit controls an interface:
  - Bit 0: Uses GPIO<3:0> ports as UART interface.
  - Bit 1: Uses GPIO<7:4> ports as Master SPI interface.
  - Bit 2: Uses GPIO<11:8> ports as Slave SPI interface.
  - Bit 3: Uses GPIO<12> as PWM1 interface.
  - Bit 4: Uses GPIO<13> as PWM2 interface.
  - Bit 5: Uses GPIO<14> as Low Active Interrupt Input.
  - Bit 6: Uses GPIO<15> as High Active Interrupt Input.

5.4.3.7. INTERRUPT INPUT PINS

GPIO[14] and GPIO[15] can be configured as Interrupt lines, with a dedicated IRQ line each:

- GPIO[14] performs an interrupt when there is a high-to-low transition.
- GPIO[15] performs an interrupt when there is a low-to-high transition.

If these signals are configured as Interrupt lines, the data written in bit 14 or 15 of PIO_DATA_GPIO register is set to ‘0’. Therefore, the transition will activate only the dedicated IRQ line, not the global GPIO IRQ line.

5.5. THE Q-EYE AND THE EYE-RIS IN OPERATION

This section demonstrates the Q-Eye CVIS and the vision systems Eye-RIS in operation. Examples are included to illustrate the usage of embedded processing for image acquisition enhancement and for vision tasks. As already explained in Chapter 3, optical acquisition parameters can be controlled in different ways to enhance quality of acquired images. Particularly, acquisition parameters can be made to depend on the input signal thus implementing adaptive acquisition processes. For instance, the average intensity level of a frame can be calculated by the Q-Eye in different ways (either using the diffusive grid or connecting all local data nodes to a global node), and then employed to reprogram the exposure time of next exposition.

5.5.1. HDR SIGNAL ACQUISITION

HDR ACQUISITION USING WELL ADJUSTMENT. Besides linear integration, image acquisition according a nonlinear law can be implemented by changing the pixel reset values during exposure. This is the so-called stepped reset technique, where integration process of the brighter photodiodes is temporarily stopped at intermediate reset values thus implementing a nonlinear compression law [Gama02] – see Figure 25.

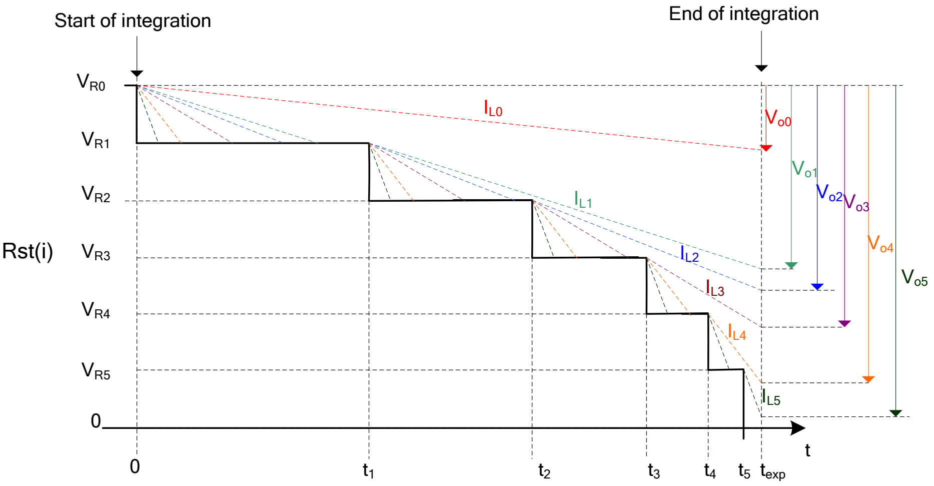
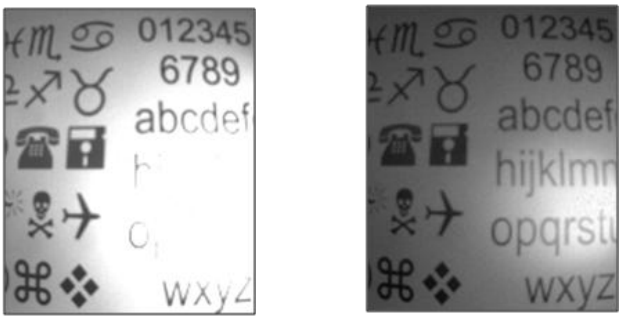


Figure 25. RESET CONTROL SIGNAL EVOLUTION.

The compression law is defined by the levels of the reset control signal of photo-sensor and the integration time allowed for each level. Figure 26 shows images captured by the Q-Eye by using a compression law with a 20dB increment in the intra-frame Dynamic Range respect to the linear integration.



Linear acquisition HDR acquisition

Figure 26. HDR IMAGE ACQUISITION PROGRAMMED IN THE Q-EYE.

The extension of the integration intervals in Figure 25 can be programmed by different ways. For instance, by counting the number of pixels ($Nsat_i$) that are saturated by the end of each subinterval and making,

$$\frac{Nsat_i}{N_T}=\frac{V_{Ri}}{V_{R0}} \qquad \textbf{Eq. 370}$$

where N_T is the total pixel number and V_{Ri} is the reset level for each sub-interval. This enables obtaining HDR images with equalized histograms. For example, if the reset level are equally spaced, then in each integration interval the relation of saturated pixel must be [5/6, 4/6, 3/6, 2/6, 1/6], where the last fraction corresponds to the last integration interval.

The integration time control is carried out by the processing circuitry included in the Q-Eye pixel. For each reset level V_{Ri}, the integration node of photodiode (Floating Diffusion node) is read by a non-destructive mechanism at equally spaced instants of time. A thresholding operation is applied to these intermediate images. The threshold level depends on which reset voltage is being applied to the gate of reset transistor. The global mean of binary image resulting from the thresholding operation is calculated. This mean value is a measurement of the number of saturated pixels in the image. When this mean value crosses a certain threshold, the next reset level is applied. Figure 27 shows a diagram describing this adaptive algorithm. In the case of Eye-RIS system, the NIOS microprocessor reads the global mean value converted by the Miscellaneous ADC and performs the last comparison to take the decision if the reset level have to be changed or not. In case that the reset level must be changed the NIOS microprocessor will send the corresponding order to the Q-Eye Controller.

The adaptation to the illuminance conditions in the scene can generate flickering among acquired images due to sudden changes in lighting conditions. Therefore, an inter-frame control algorithm can be implemented in the NIOS microprocessor in order to avoid this flickering.

HDR IMAGING USING MULTIPLE ACQUISITION. Another common technique to acquire HDR images which has been implemented in the Q-Eye pixel is based on multiple acquisitions increasing exposure time according to a specific law. During the sensing phase the integration node of photodiode will be read several times following a non-destructive procedure. These captures constitutes the multiple acquisitions which will be independently processed first, in order to remove the spatial noise associated to the pixel output buffer behavior, and then all results will be combined to obtain a HDR compression. Figure 28 shows an example of this multiple acquisition technique implemented in the Q-Eye.

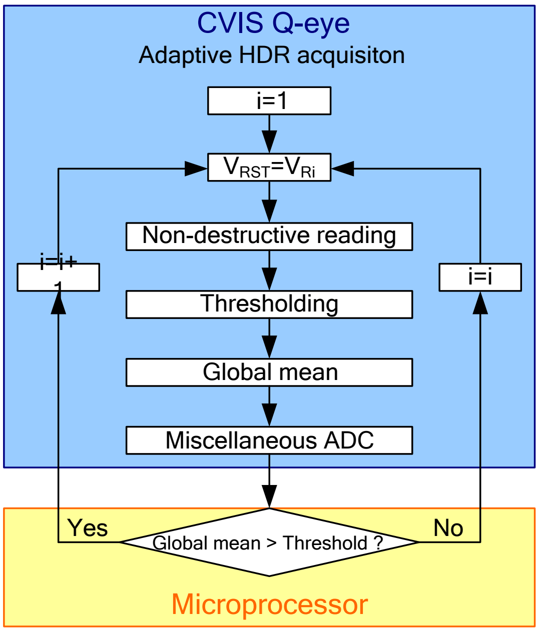


Figure 27. ADAPTIVE **HDR** ALGORITHM.

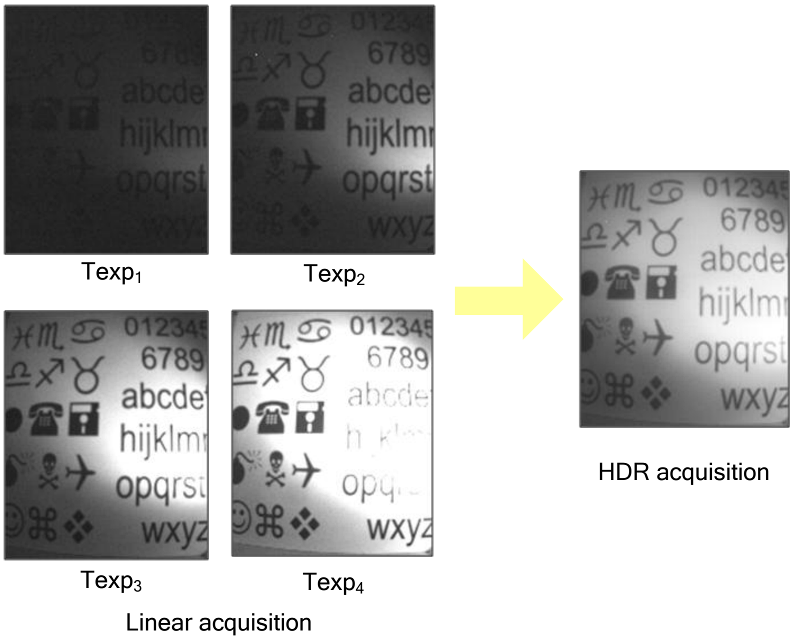


Figure 28. MULTIPLE ACQUISITION TECHNIQUE IMPLEMENTED BY Q-EYE.

5.5.2. SPATIAL FILTERING

As outlined in previous Chapters, *spatial filtering* is usually employed at the early stages of the vision processing chain – see Figure 7 and Figure 8 in Chapter 3. Typical spatial filtering operations are aimed to attenuate noise and spurious, edge sharpening and image feature enhancement, etc. [Gonz92] [Gonz15] [Toma06]. Following enhancement phase, *features* of interest can be detected to identify objects in the scene for their further characterization. For instance, borders, corners, blobs, etc.

Spatial filtering can be implemented in the Q-Eye and Eye-RIS systems by applying suitable spatial *convolution masks* [Anaf2v1c]. Image features can be also detected by applying spatial convolution masks, as described in Chapter 3. As an alternative, spatial *low-pass* filters can be implemented through *diffusive* processes emulating Gaussian filtering. In this case, *high-pass* filters can be implemented by using additional arithmetic operation on top of that [Fern12].

Regarding convolutions, they are implemented by the Q-Eye though a sequence of FPP instructions; decision-making is not needed for these operations and hence the intervention of the NIOS microprocessor is not needed either. This is key for high operation speed at system level. Also, in order to reduce silicon area and power consumption of the FPP circuitry, the MAC which develops the arithmetic operations employs only three multiplicative factors, namely: 2, 1, or ½. Thus, the accumulation of neighbor contributions is carried out serially accessing each neighbor through the *image shifter block*. Readers are referred to Section 3.2 in Chapter 3 for details about these blocks. This reduced number of MAC block multiplicative factors and the limitation in the number of iterations result in restrictions on the templates coefficients which can be implemented by the Q-Eye [12]. A general definition for the template elements is:

$$\left\{ \frac{p}{q} \middle| \, p \in \mathbb{Z}[-8,8], q \in \mathbb{N}\{1,2,4,8\} \right\} \qquad \textbf{Eq. 371}$$

[12] Due to the area restriction in the pixel, the accuracy of analogue operations is limited. Convolutions are hence low precision operations. The quality of the operation depends on the number of non-zero elements of the template and the dynamic range of the filter, defined by the difference between the minimum and the maximum absolute weight values of the template. The higher dynamic range is, the lower the precision of the result, because the number of iterations in the Multiplier-Accumulator Circuit (MAC) is higher.

In order to guarantee minimum accuracy level, the elements of template must belong to the interval $\mathbb{Z}[-8,8]$. Thus, a global pre-scale is applied to the input image to preclude saturations during the operation development. In this case the allowed coefficients are given by:

$$\left\{ \frac{p}{q} \middle| p \in \mathbb{Z}[-1,1], q \in \mathbb{N}\{1,2,4,8\} \right\} \quad \textbf{Eq. 372}$$

However, this does not limit the variety of filters that can be implemented in the Q-Eye. Same examples are listed in the next paragraphs.

SHARPEN FILTER. This filter is commonly used for image enhancement and produces a result equivalent to that obtained by applying a Laplacian filter to the input image and adding the latter to the result. Usual convolution masks which for this filter are:

$$T_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad T_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad \textbf{Eq. 373}$$

where the template T_2 can be expressed as a combination of two templates:

$$T_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \textbf{Eq. 374}$$

Figure 29 shows the results obtain by the Q-Eye implementing the convolution mask T_2.

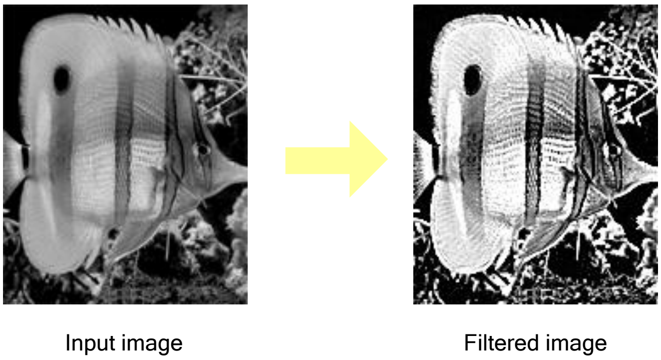


Figure 29. SHARPEN FILTER DEVELOPED BY THE Q-EYE.

LAPLACE FILTER. This function performs a high-pass filter over the input image. It can be implemented by the convolution masks:

$$T_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad T_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad \textbf{Eq. 375}$$

This filter is applied for detecting edges. It is a discrete formulation of the second-derivative, isotropic Laplacian operator [Gonz15]. Figure 30 represents the results obtained by the convolution mask T_2 implemented by the Q-Eye.

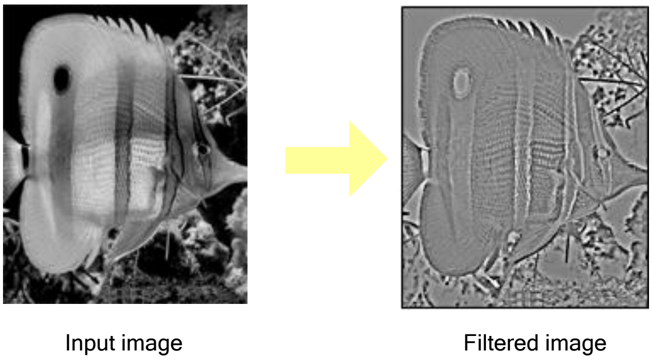


Figure 30. LAPLACIAN FILTERED IMPLEMENTED IN THE Q-EYE SYSTEM.

SOBEL FILTER. This other high-pass filter is applied to detect directional edges. It is a discrete formulation of the first-derivative operator, gradient [Gonz15]. Convolution masks to detect edges in different directions are defined by the following templates:

$$T_1 = \begin{pmatrix} -1 & 0 & -1 \\ -2 & 0 & -2 \\ -1 & 0 & -1 \end{pmatrix} \quad T_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \qquad \textbf{Eq. 376}$$

$$T_3 = \begin{pmatrix} 0 & -1 & -2 \\ -1 & 0 & -1 \\ -2 & -1 & 0 \end{pmatrix} \quad T_4 = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & -2 \end{pmatrix} \qquad \textbf{Eq. 377}$$

Templates T_1 and T_2 detect edges in the horizontal and vertical directions, while templates T_3 and T_4 find edges in diagonal directions. Figure 31 shows the results obtained by the Q-Eye.

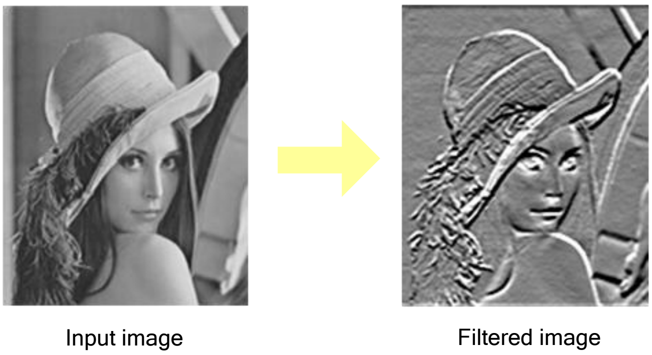


Figure 31. SOBEL FILTER IMPLEMENTED IN THE Q-EYE SYSTEM.

PREWITT FILTER. It is another approximation to the first-derivative, applied to find edges. Convolutions masks that define this filter are represented by T_1 and T_2.

$$T_1 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad T_2 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \qquad \textbf{Eq. 378}$$

Figure 32 presents the results obtained by the implementation of a vertical Prewitt filter in the Q-Eye processor.

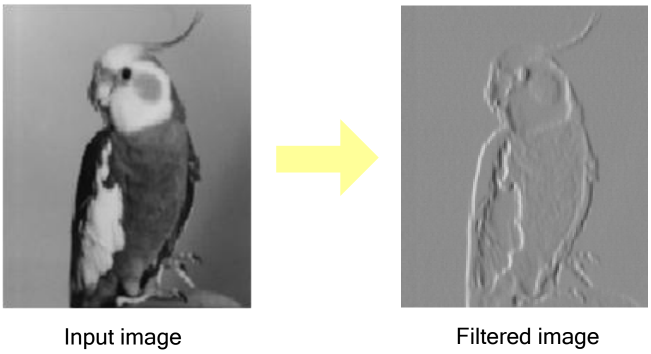


Figure 32. PREWITT FILTER CARRIED OUT BY THE Q-EYE.

LOW-PASS FILTERING THROUGH DIFFUSIONS. The processing array of Q-Eye includes a diffusion grid which allows implementing Gaussian filters with the sigma parameter defined by the diffusion time. Figure 33 shows the result obtained by the Q-Eye system when a diffusion process with $\sigma = 0.4$ is carried out.

A remarkable characteristic of this diffusion operation is that can be masked, only pixels point out by a binary image defined as a mask are updated. Figure 34 presents an example carried out by the Q-Eye where a diffusion mask is defined.

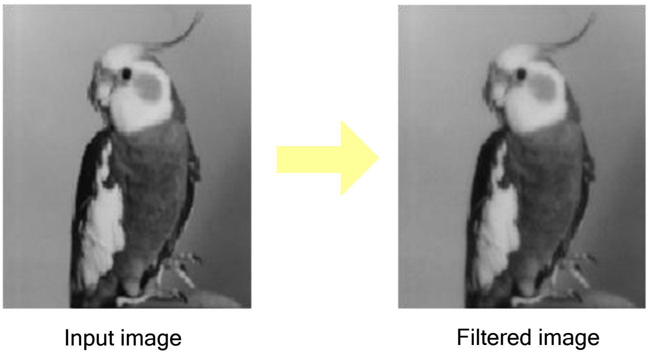


Figure 33. GAUSSIAN FILTER IMPLEMENTED BY Q-EYE.

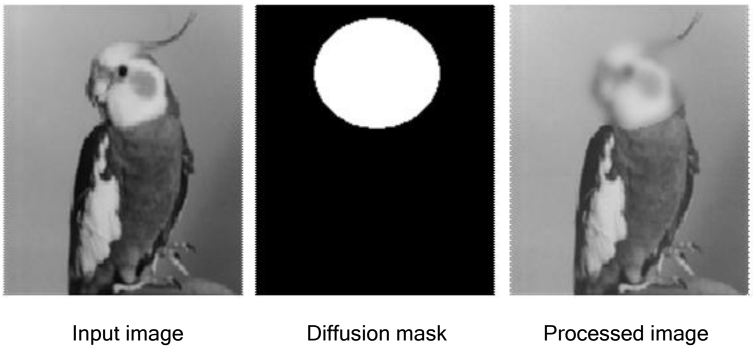


Figure 34. MASKED DIFFUSION PROCESS.

HIGH-PASS FILTERING THROUGH DIFFUSIONS. Diffusive processes implement low-pass filtering functions. High-pass and band-pass filters can be implemented by combining low-pass filtering and arithmetic operations. When the accuracy of output image is not guaranteed due to this image results from the application of convolution mask with a high number of non-zero coefficients (which provokes an accumulation of errors associated to the sequence of analog operations), the use of Gaussian filters in combination with arithmetic operations lead to very efficient filter implementation in terms of speed, accuracy and power consumption.

Figure 35 shows the combined use of arithmetic operations and Gaussian filters to implement a high-pass filter in the Q-Eye.

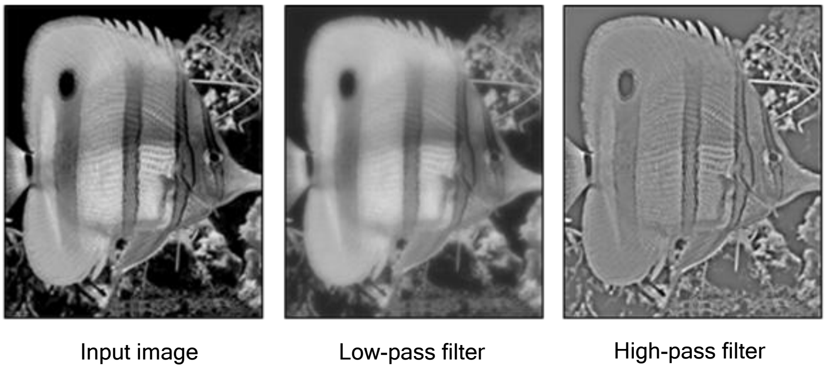


Figure 35. HIGH-PASS FILTER BASE ON LOW-PASS FILTER AND SUBSTRACTION.

The high-pass filtered image is the result of subtraction of the input image and low-pass filtered version.

5.5.3. EXTRACTION OF IMAGES FEATURES AND CALCULATION OF INTERMEDIATE IMAGE REPRESENTATIONS

In the vision processing chain, feature detection is typically followed by *segmentation*. An important task for segmentation is *thresholding*. The outcome of thresholding operation is a binary image from which the useful information can be extracted by applying *combinational* and *morphological* functions. There are several thresholding operations available in the Q-Eye – described in subsequent paragraphs.

5.5.3.1. THRESHOLDING

GLOBAL THRESHOLD: This function is mathematically described by:

$$im_{out}(i,j)=\begin{cases}1 & im_{in}(i,j)\geq th\\ 0 & im_{in}(i,j)<th\end{cases} \qquad \textbf{Eq. 379}$$

This threshold operation is *global*, all pixels are compared with the same threshold value (th). Figure 36 presents a global thresholding developed by the Q-Eye.

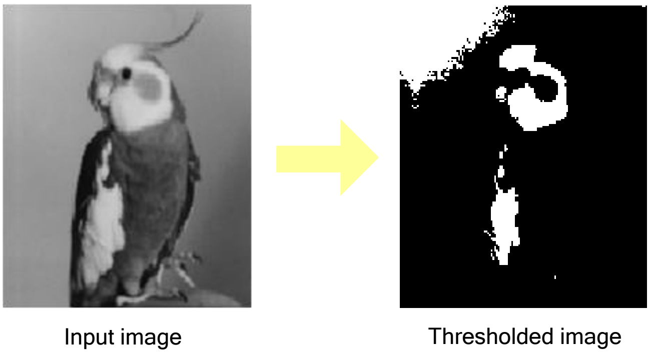


Figure 36. GLOBAL THRESHOLD IMPLEMENTED BY THE Q-EYE.

ADAPTIVE GLOBAL THRESHOLD: In this case, the global operation employs a threshold value calculated at the outcome of an adaptive algorithm with the following steps:

- An initial estimate for threshold reference (th) is made.

- The image is segmented using *th*. Two sets of pixels are created as a result, S_1 representing the pixels whose grey level is above th and S_2 representing the pixels whose grey level is below th.
- The mean value of the two sets of pixels, M_1 for S_1 and M_2 for S_2, are calculated.
- A new threshold is calculated:

$$th = \frac{1}{2} \cdot \left(M_1 + M_2 \right) \qquad \textbf{Eq. 380}$$

- Successive iterations are made until the value of th is smaller than a specific tolerance.

Figure 37 shows the results of applying this adaptive threshold operation.

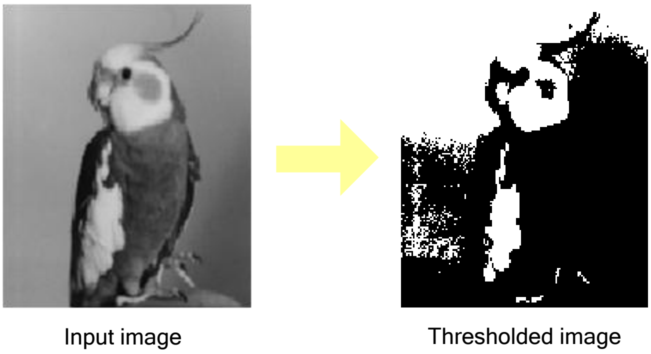


Figure 37. ADAPTIVE GLOBAL THRESHOLD IN THE Q-EYE.

LOCAL THRESHOLD: This threshold operation is represented by:

$$im_{out}(i,j) = \begin{cases} 1 & im_{in}(i,j) \geq th(i,j) \\ 0 & im_{in}(i,j) < th(i,j) \end{cases} \qquad \textbf{Eq. 381}$$

In this case, the threshold value is not a global constant for all pixels. Instead, each pixel is compared to a different value. This operation is an extremely efficient operation as it is performed simultaneously in all pixels.

Figure 38 presents the binary image obtained in the Q-Eye system when the threshold image is obtained by applying a Gaussian filter to the input image. This Gaussian filter has to be associated a high σ parameter in order to obtain the information contained in the low frequency domain to calculate the local threshold.

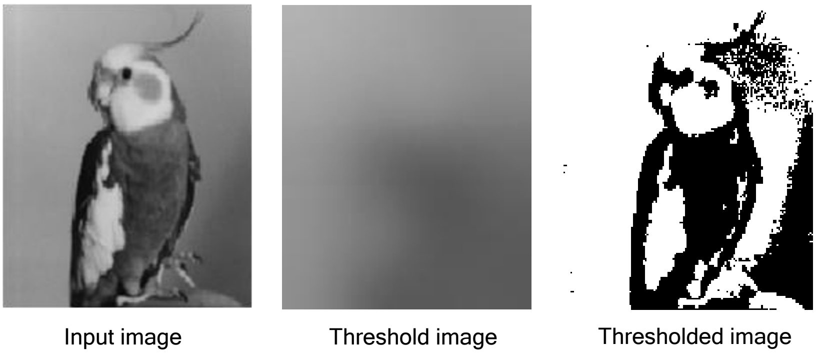


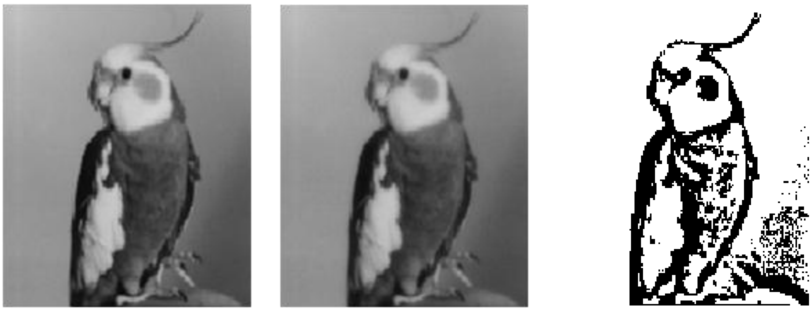
Figure 38. LOCAL THRESHOLD IMPLEMENTED BY Q-EYE.

ADAPTIVE LOCAL THRESHOLD: The threshold image considered to carry out the local threshold is calculated by statistically examining the intensity values of a local neighborhood around each pixel. In this case, a spatial low-pass filter of the local intensity distribution is performed over the input image. The size of the neighborhood (or, in other words, the bandwidth of the low-pass filter) has to be large enough to cover sufficient foreground and background pixels; otherwise a poor threshold will be obtained.

Figure 39 shows an example of adaptive local threshold performed by Q-Eye.

5.5.3.2. BINARY OPERATIONS

The binary images obtained after the thresholding operation are processed by binary functions with the main objective of selecting the objects of interest in a scene depending on their topological features. Binary operations are more robust than MAC and other analogue operations, because the result is always regenerated. Thus, a large number of operations can be concatenated without compromising accuracy.



Input image Low-pass filtered image Thresholded image

Figure 39. ADAPTIVE LOCAL THRESHOLD.

The Q-Eye pixel includes circuitry for parallel implementation of the following binary functions:

- *Logic operations*: not, and, or, xor, nand, nor, nxor, etc.
- Generic *hit and miss* operation
- Set of binary *morphological* operations
- Set of *flooding* functions allowing a basic management of blobs and holes.

HIT-AND-MISS OPERATIONS. They look for a given pattern in a binary image. The pattern is specified by a 3 x 3 matrix where each element can be 1, or equivalently 'W' (white), DNC (Do Not Care) or 0, 'B' (black). For illustration purposes, Figure 40 describes the patterns required to look for bottom-right corners in an image.

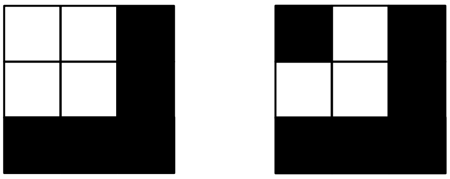


Figure 40. BOTTOM-RIGHT CORNER PATTERNS.

These patterns are represented by the matrix:

$$\begin{pmatrix} DNC & W & B \\ W & W & B \\ B & B & B \end{pmatrix} \qquad \textbf{Eq. 382}$$

Figure 41 shows the outcome of applying the hit-and-miss operation described by this pattern to the image shown at the left. Note that white pixels in the processed image determine the points where a corner meeting the specified pattern has been found.

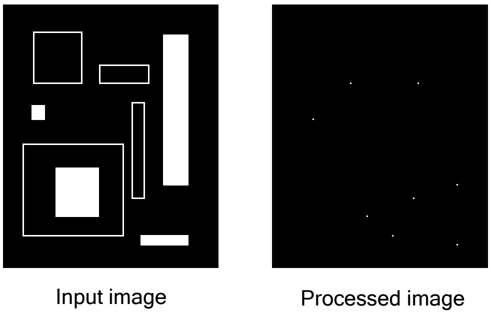


Figure 41. HIT-AND-MISS OPERATION IMPLEMENTED BY Q-EYE.

MORPHOLOGICAL FUNCTIONS. For implementation purposes, the Q-Eye considers that objects are white and the background is black. The binary morphological operations are usually based on some pattern also called ***STRUCTURING ELEMENT***. This structuring element indicates the connectivity associated to the operation. Common structuring elements used in morphological operations are:

- 4-neighbor connectivity: $\begin{pmatrix} DNC & 1 & DNC \\ 1 & 1 & 1 \\ DNC & 1 & DNC \end{pmatrix}$
- 8-neighbor connectivity: $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

ERODE: This function erodes a binary image following a specified pattern, called structuring element. Figure 42 shows a example of applying this operation.

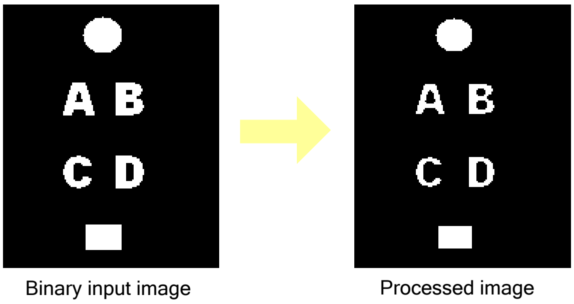


Figure 42. ERODE OPERATION PERFORMED BY Q-EYE SYSTEM.

DILATE: This function dilates a binary image in base to a particular structuring element, see Figure 43.

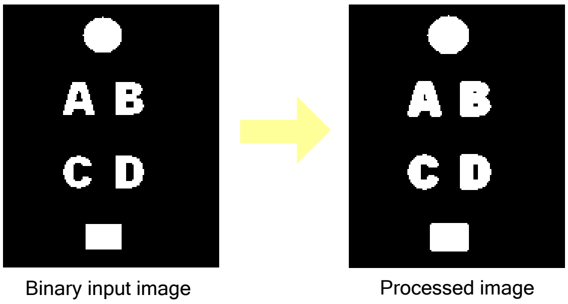


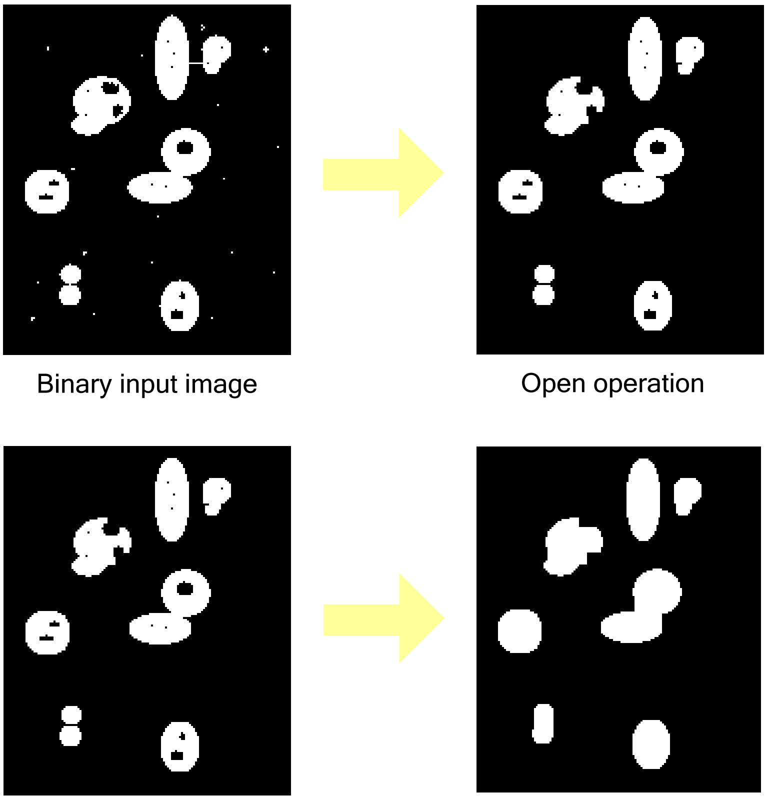
Figure 43. DILATE OPERATION.

OPEN AND CLOSE: The **OPEN** function performs an erosion followed by a dilation of a binary image. It is typically used to either removing noise or breaking isthmuses between touching objects. Regarding the **CLOSE** function, it realizes a dilation followed by an erosion. It is often used to fill holes and connect blobs.

Figure 44 represents the results obtained by open and close operation. The binary input image is a very noise image, there are several noise spots in the background and object of interest have holes. The combined of open and close operations is used to solve these problems.

THIN: This function realizes a directional erosion of a binary image according to the input structuring element which will define the direction of the thinning. Figure 45 presents the results applying the operation in vertical and horizontal directions.

THICKEN: This function realizes a directional dilate of a binary image according to the input structuring element which will define the direction of the thinning. Figure 46 presents the results of applying the operation in vertical and horizontal directions.



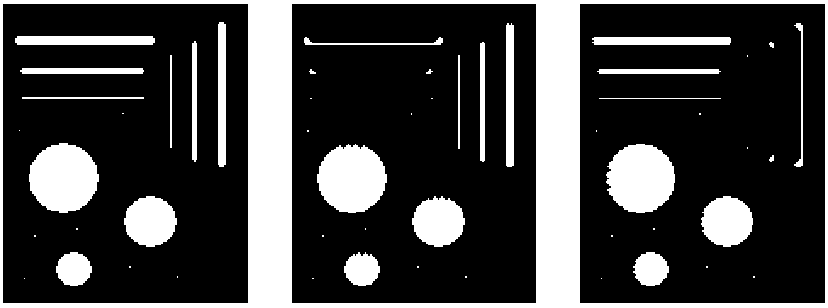
Binary input image

Open operation

Open operation

Several iterations of close operation

Figure 44. OPEN AND CLOSE MORPHOLOGICAL OPERATIONS IMPLEMENTED IN THE Q-EYE.



Input image

Vertical thin operation

Horinzontal thin operation

Figure 45. THIN MORPHOLOGICAL OPERATION IN THE Q-EYE.

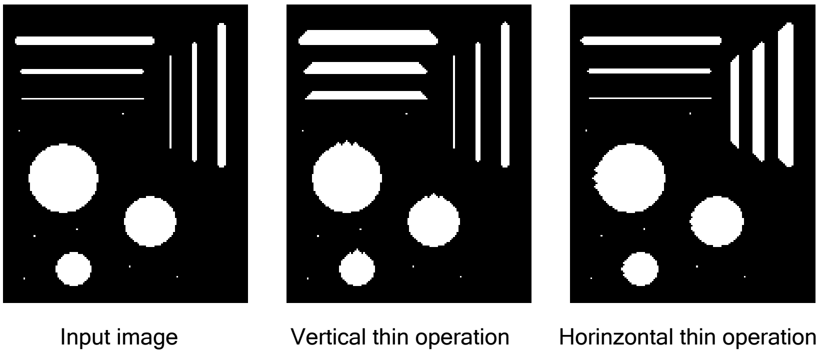


Figure 46. THICKEN OPERATION IN THE Q-EYE.

CENTROID: This function gets the position of the *centroids* of *blobs* (connected components) present in a binary image. This operation peels the image one pixel off as many times as indicated by the user or until no change occurs between iterations, see Figure 47.

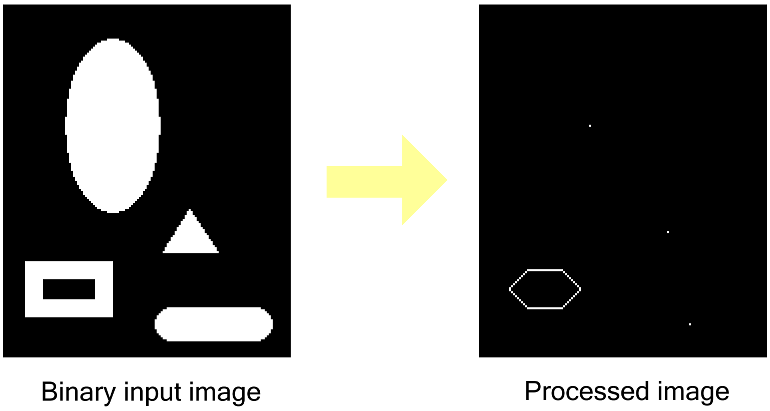


Figure 47. CENTROID OPERATION IN THE Q-EYE.

SKELETON: This function calculates the *skeleton* of a binary image. The skeleton of a binary connected component or blob is often defined as a one-pixel thick line that is in the middle of the object and represents the shape or topology of blob. This operation peels the image one pixel off in the direction of skeleton as many times as indicated by the user or until no change occurs between iterations, see Figure 48.

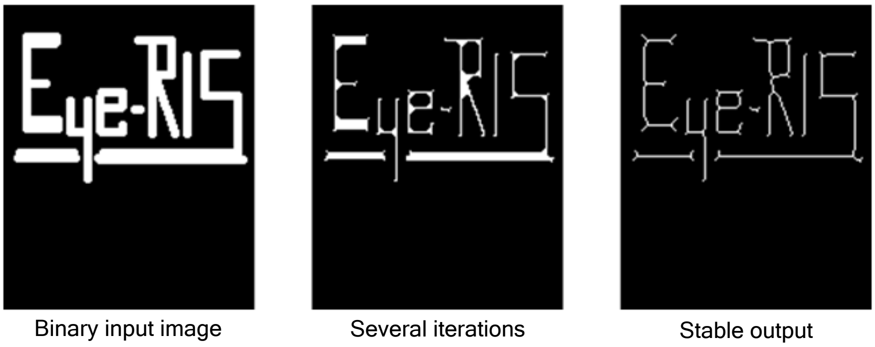


Figure 48. SKELETON OPERATION IN THE Q-EYE.

REMOVE SINGLE POINTS: This function deletes from a binary image the isolated white pixels. This function is used to remove noise in binary images, see Figure 49.

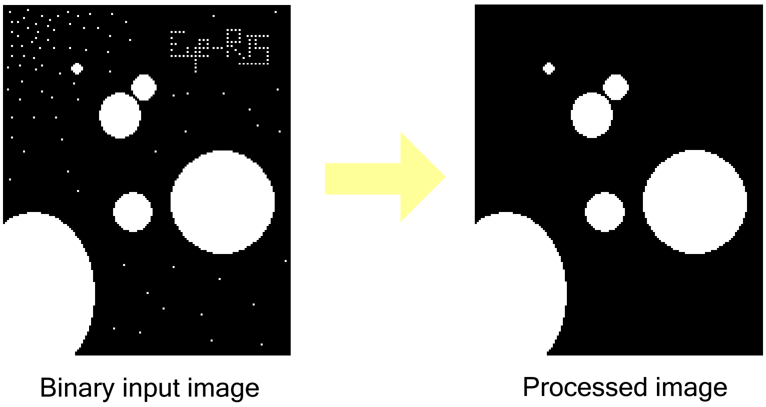


Figure 49. **REMOVE SINGLE POINTS IN THE Q-EYE.**

END POINTS: This function calculates the end points of a binary image, defined as those active pixels that have one and only one active neighbor, see Figure 50.

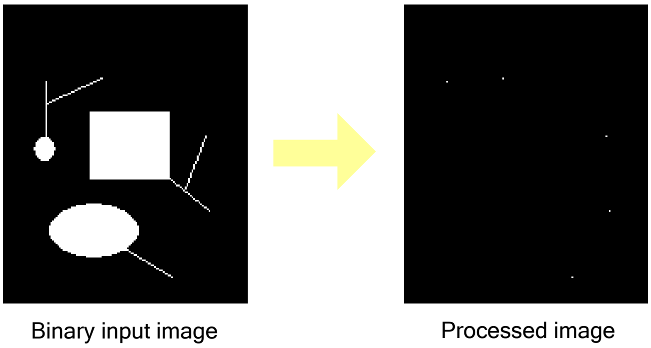


Figure 50. **END POINTS OPERATION IN THE Q-EYE.**

PRUNE: This function iteratively detects and deletes the end points of a binary image, which will typically be the result of a skeletonization. This is done as many times as indicated by the user or until no change occurs between iterations, see Figure 51**.**

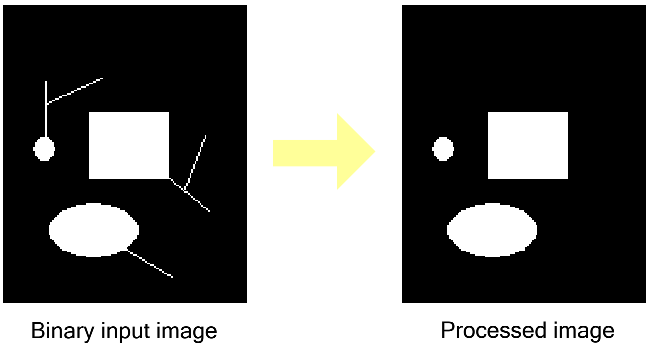
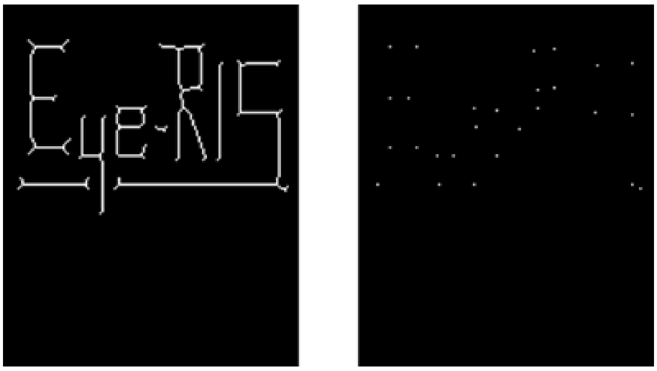


Figure 51. **PRUNE MORPHOLOGICAL OPERATION IN THE Q-EYE.**

SKELETON JOINTS: This function detects the join points of a binary image, defined as those points where three or more branches of a skeleton meet, see Figure 52.

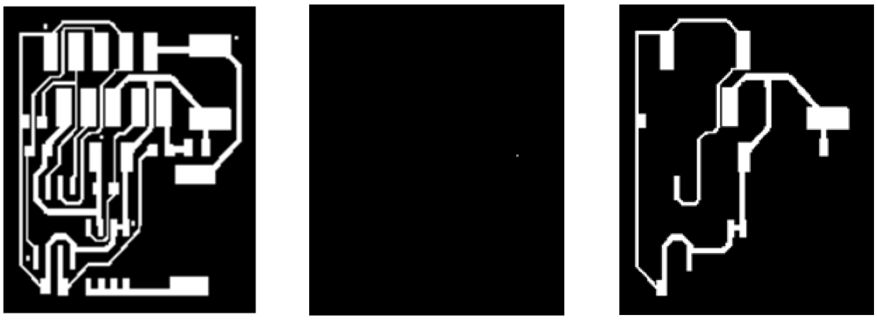
FLOOD FILL: This function reconstructs objects present in a binary image. The seed image is the starting point for the reconstruction, and the blobs image constrains the growth of the region. This operation uses the Resistive Grid module. A passive, continuous time propagation wave fills the blobs, see Figure 53.



Binary input image

Processed image

Figure 52. **SKELETON JOINTS IN THE Q-EYE.**



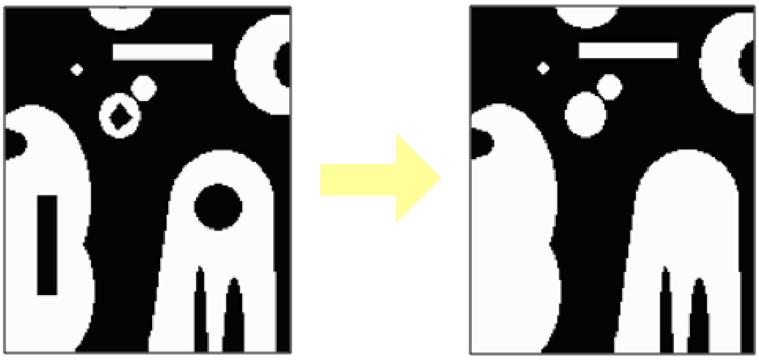
Binary input image

Seed point

Processed image

Figure 53. **FLOOD FILL OPERATION IN THE Q-EYE.**

FILL HOLES: This function fills the holes present in the blobs of binary image. The holes must not touch the border of the image in order to be filled. If they do, they remain unchanged, see Figure 54



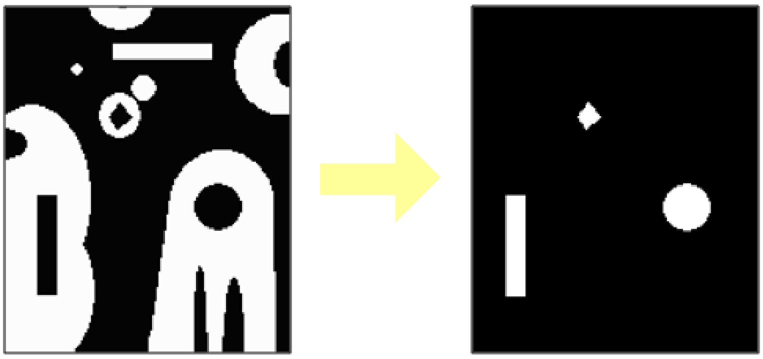
Binary input image

Processed image

Figure 54. **FILL HOLES OPERATION IN THE Q-EYE.**

EXTRACT HOLES: This function extracts the holes present in the blobs of binary images. The holes must not touch the border of the image in order to be extracted. If they do, they remain unchanged, see Figure 55.

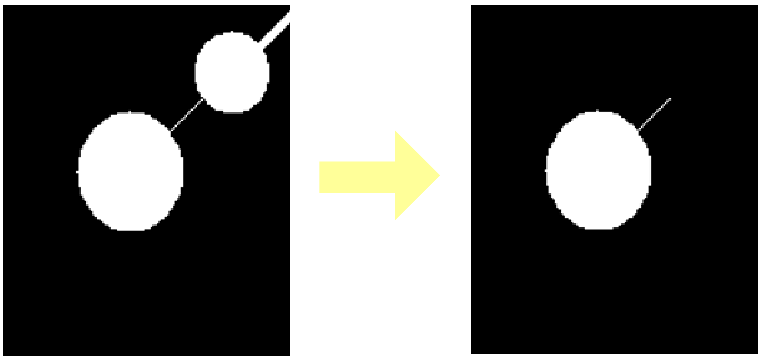
ERASE BORDER BLOBS: This function erases the blobs which are touching the borders of an image. The rest of blobs remain unchanged. This operation uses the Resistive Grid module. A passive, continuous time propagation wave fills the blobs, see Figure 56.



Binary input image

Processed image

Figure 55. EXTRACT HOLES IN THE Q-EYE.



Binary input image

Result with 4-connectivity

Figure 56. ERASE BORDER BLOBS IN THE Q-EYE.

5.5.4. DECISION MAKING

Figure 57 below is a copy of Figure 8 from Chapter 3. It illustrates the progressive reduction of data throughout the vision processing chain. As illustrated in the figure, the number of data is progressively reduced from the initial image to the final decision. The binary images resulting after segmentation phase, where thresholding and morphological operation play an important role, contain the information of interest. This information consists of a much smaller amount of data contained in the acquired images. Therefore, these binary images containing the information of interest (blobs, contours, corners, centroids, etc.) can be downloaded through the binary image interface or using the Address Event interface at a high frame rate, with the objective that the NIOS microprocessor can process these images in order to describe and classify the objects present in the scene for a subsequent decision-making.

The Q-Eye has by itself no capacity for decision-making. It is a pure co-*processor* which only carries out operations. Therefore, the Q-Eye system requires a controller to sequence all microinstructions needed to develop the operations involved in the early-processing. For example, in those cases where an adaptive algorithm is required in order to implement a particular early-processing function, the controller or microprocessor must take decisions based on data provided by the Q-Eye and change some parameters.

In the Eye-RIS systems, the NIOS-II microprocessor is responsible for *high-level* processing, whose inputs are the images resulting from the segmentation phase. The NIOS microprocessor carries out the description and classification of objects of interests in the images obtained by the early-processing. In function of classification results, the NIOS microprocessor will take a specific decision.

As an example in the decision-making by the vision system Eye-RIS, an application developed by Anafocus Company is presented. In this case, the Eye-RIS system is used to add interactive vision to unmanned systems. In this application, a robot car (see Figure 58) tracks the road-lines while actively responding to signal indications and obstacles.

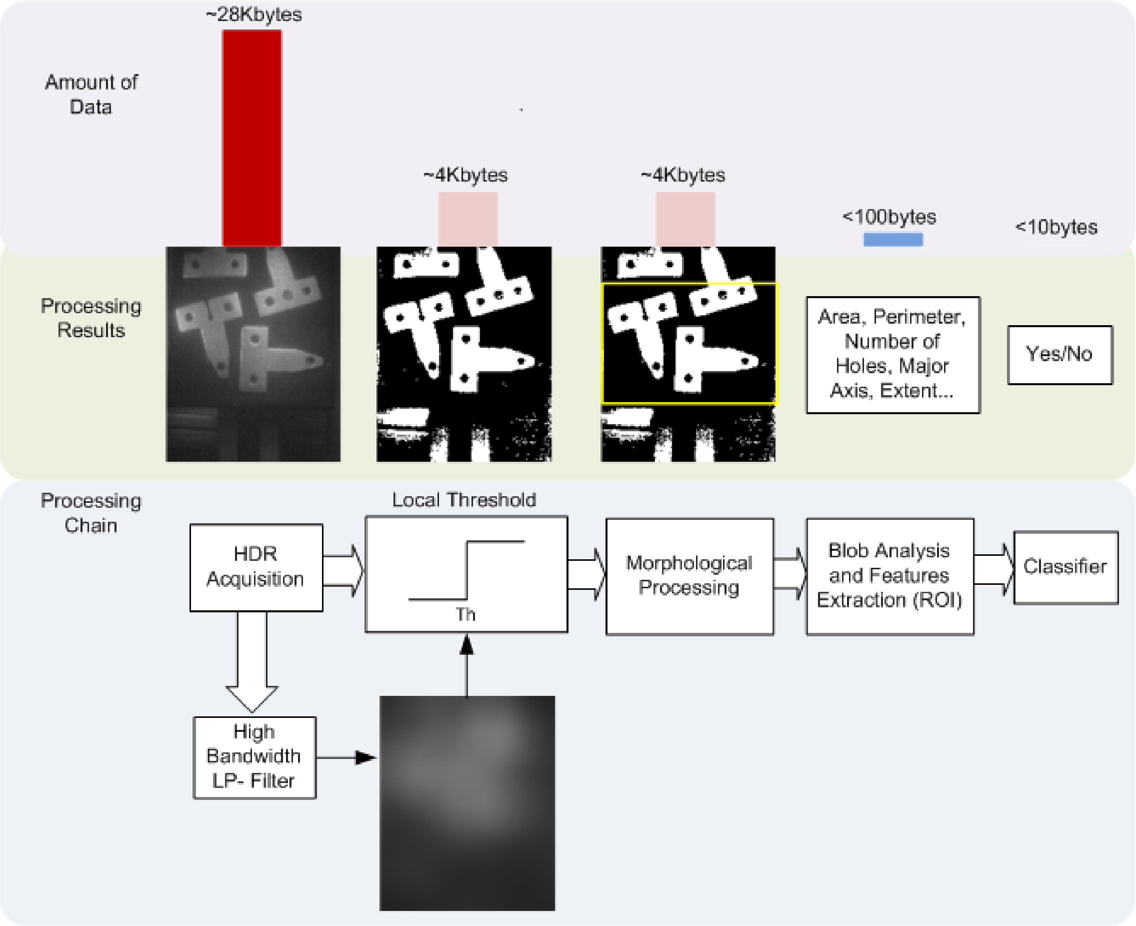


Figure 57. ILLUSTRATION OF THE PROGRESSIVE DATA REDUCTION OF DATA ALONG THE VISION PROCESSING CHAIN AS IT ACTUALLY HAPPENS IN THE EYE-RIS. ALL STEPS OF THE PROCESSING CHAIN ABOVE BUT THE LAST ONE ARE COMPLETED IN THE Q-EYE SENSORY-PROCESSING FRONT-END. THUS, THE DATA SET DELIVERED FOR PROCESSING BY THE HOST DIGITAL PROCESSOR IS QUITE SMALL.

Figure 59 depicts a flow diagram which describes the main tasks involve in this application. The acquisition and early-processing tasks are developed by the smart sensor Q-Eye, while the analysis, classification and decision-making are carried out by the NIOS microprocessor.

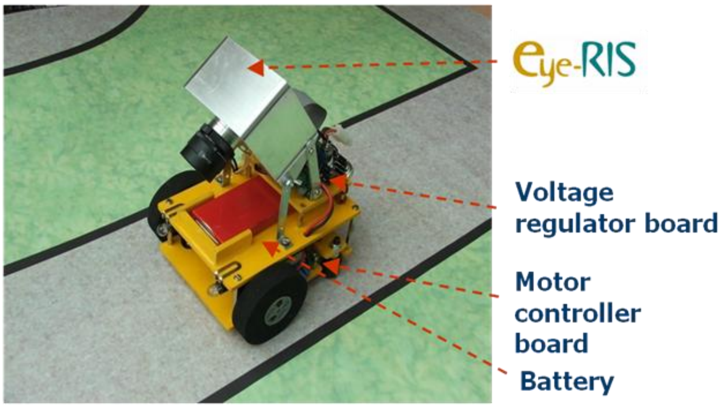


Figure 58. UNMANAGED GROUND VEHICLE.

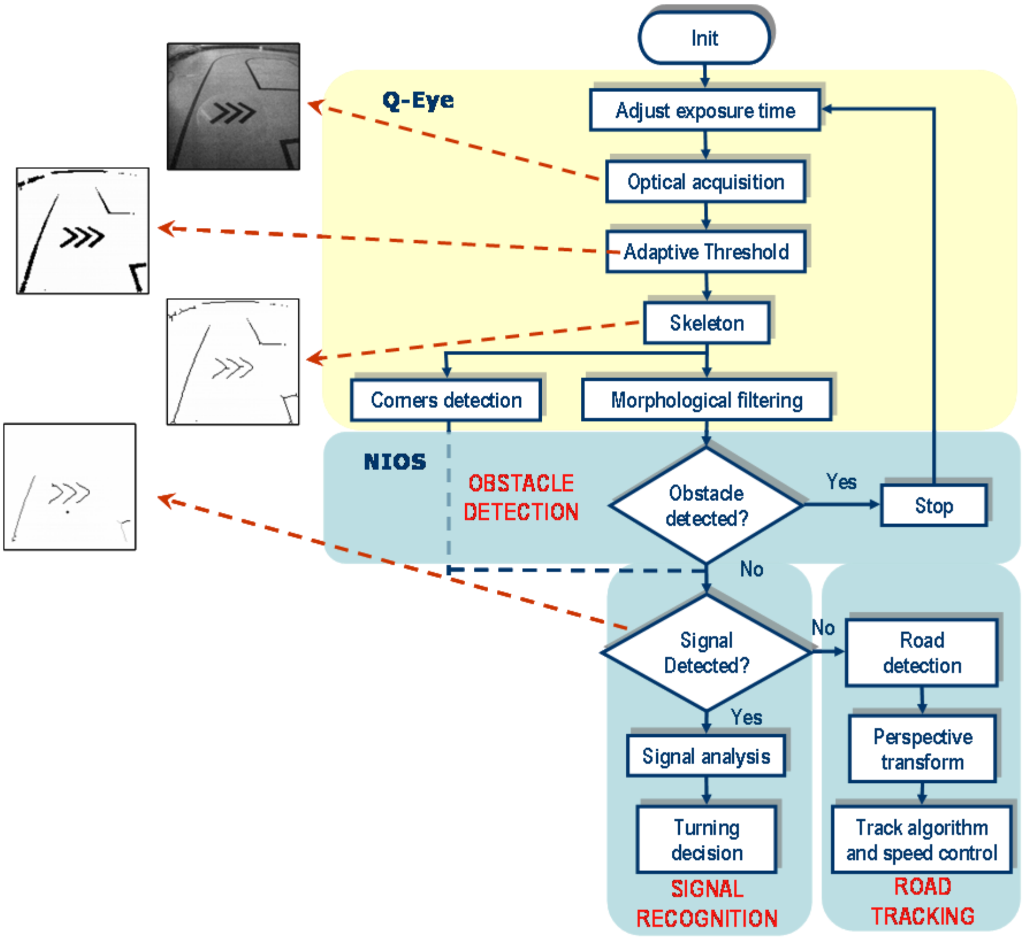


Figure 59. APPLICATION FLOW DIAGRAM.

APPENDICES

APPENDIX A – SETTLING ERRORS IN GREGORIAN’S S&H

The SC circuits implemented in the processing of Q-eye are based mainly in a topology similar to the Gregorian’s S&H circuit, which is described in Figure 1. Therefore, in this appendix, the settling errors associated to the S&H operation are obtained because these results are applicable to the analysis of settling errors in the circuits of processing cell.

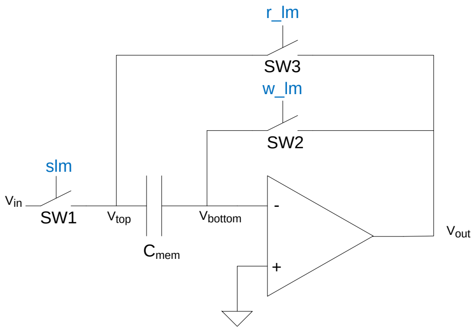


Figure 1.- GREGORIAN’S S&H

The timing diagram associated to the control phases is represented in Figure 2.

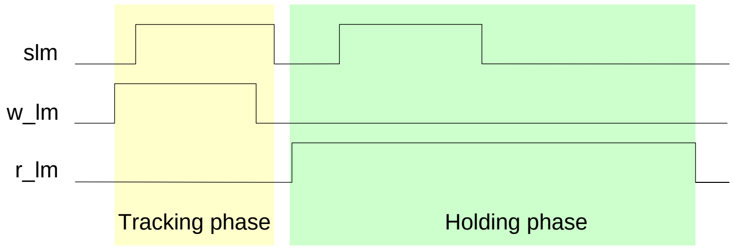


Figure 2.- CONTROL PHASES

1. SETTLING ERRORS IN TRACKING PHASE (WRITING PHASE)

During the tracking phase, the dynamic behavior of circuit is described by the electrical model represented in Figure 3.

If SW2 switch is turned off first at the end of tracking phase, the charge injection and feedthrough introduced by SW1 switch does not affect to the data stored in the S&H. Therefore, this switch can be designed in order to make negligible the effect of its resistance in the dynamic behavior of circuit.

Under this assumption, the transfer function expressed in the Laplace domain which represents the dynamic behavior of S&H circuit during the tracking phase is given by the equation:

$$\frac{v_{bottom}(s)}{v_{ref}(s)} = H_{writing}(s) = \frac{g_m}{\frac{C_o\left(C_{mem}+C_{in}\right)}{g_{sw2}}s^2 + \left(C_o+C_{mem}+C_{in}\right)s + g_m} \qquad \textbf{Eq. 383}$$

The transfer function of circuit has been calculated considering the input through the reference voltage. Furthermore, g_{sw2} is the conductance associated to switch SW2, and additional approximation has been considered to obtain this transfer function: $g_o << g_m, g_{sw2}$

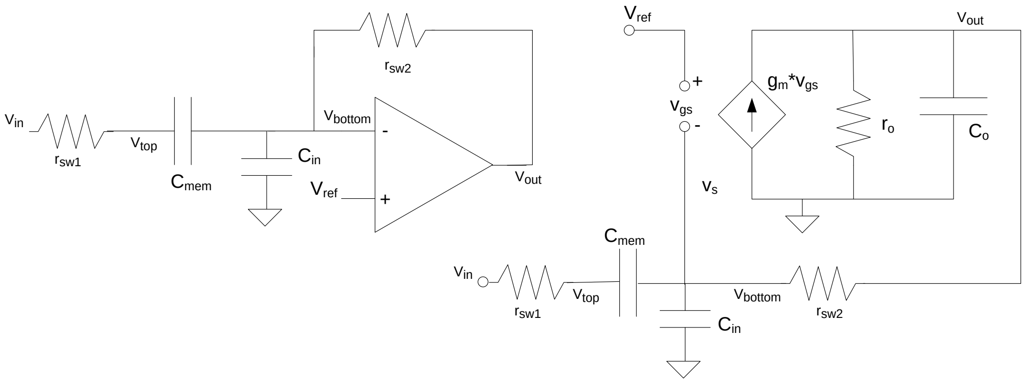


Figure 3.- CIRCUITAL MODEL FOR WRITING PHASE.

The poles of system are given by the characteristic equation:

$$\frac{C_o\left(C_{mem}+C_{in}\right)}{g_{sw2}}s^2+\left(C_o+C_{mem}+C_{in}\right)s+g_m=0 \qquad \textbf{Eq. 384}$$

This second order circuit is going to behave like a first order system when $g_{sw2} >> g_m$, resulting in the following transfer function:

$$H_{writing}(s)=\frac{1}{1+\dfrac{C_o+C_{mem}+C_{in}}{g_m}s}=\frac{1}{1+\tau_{wr}\cdot s} \qquad \textbf{Eq. 385}$$

The time response to a step input of size E can be derived using this transfer function in the Laplace domain. The transfer function corresponding to a step function with magnitude E is $\frac{E}{s}$.

In the Laplace domain, the transform of the output is the product of the circuit transfer function and input’s transform:

$$v_{bottom}=\frac{E}{s}\cdot\frac{1}{1+\tau_{wr}\cdot s} \qquad \textbf{Eq. 386}$$

Computing the inverse Laplace-transform to obtain the time response of S&H circuit in the tracking phase:

$$v_{bottom}=E\left(1-e^{-\frac{t}{\tau_{wr}}}\right)u_o(t) \qquad \textbf{Eq. 387}$$

Where $u_0(t)$, the unitary step function starting at $t=0$, is indicating that the expression is only valid for positive values of t.

The acquisition error, the difference between the sampled value at a specific point in time and the actual input, can be defined as follows:

$$v_{\varepsilon}(t) = -E\cdot e^{-\frac{t}{\tau_{wr}}} \qquad \text{Eq. 388}$$

2. SETTLING ERRORS IN HOLDING PHASE (READING PHASE)

The dynamic behavior in the holding phase is described by the circuital model depicted in Figure 4.

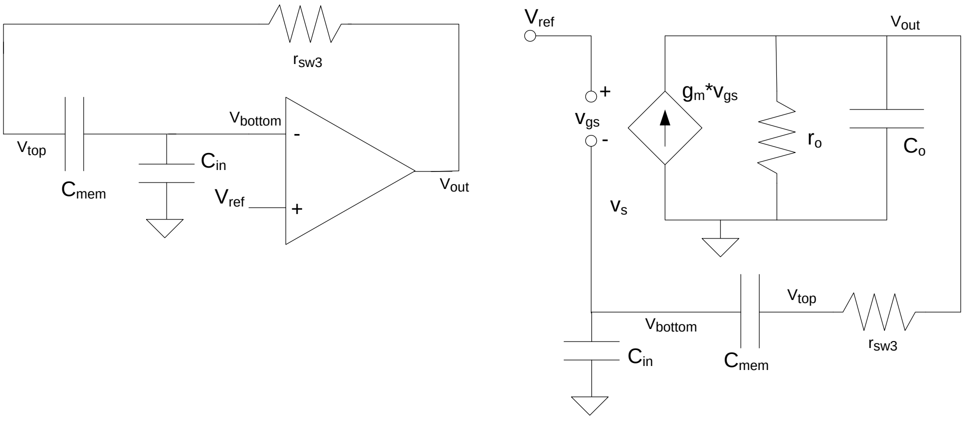


Figure 4.- CIRCUITAL MODEL FOR THE HOLDING PHASE.

In a similar way that tracking phase, to calculate the transfer function associated to the holding operation, the input is applied to the reference voltage v_{ref} and node v_{out} is considered the output.

Designing the impedance of SW3 switch in order to make negligible its effect on the dynamic of circuit, the following transfer function is obtained:

$$\frac{v_{out}(s)}{v_{ref}(s)} = H_{reading}(s) = \frac{\left(1+\frac{C_{in}}{C_{mem}}\right)}{s\cdot\frac{C_{in}+C_o\left(1+\frac{C_{in}}{C_{mem}}\right)}{g_m}+1} \qquad \text{Eq. 389}$$

$$\frac{v_{out}(s)}{v_{ref}(s)} = H_{reading}(s) = \frac{\left(1+\frac{C_{in}}{C_{mem}}\right)}{s\cdot\frac{C_{eq\_rd}}{g_m}+1} \qquad C_{eq\_rd} = C_{in}+C_o\left(1+\frac{C_{in}}{C_{mem}}\right) \qquad \text{Eq. 390}$$

$$H_{reading} = \frac{\left(1+\frac{C_{in}}{C_{mem}}\right)}{s\cdot\frac{C_{eq\_rd}}{g_m}+1} = \frac{\left(1+\frac{C_{in}}{C_{mem}}\right)}{s\cdot\tau_{rd}+1} \qquad \text{Eq. 391}$$

The settling error in response to an input step of size E during the holding phase can be calculated in the same way that acquisition error during tracking phase.

$$v_{bottom} = \frac{E}{s} \cdot \frac{\left(1 + \frac{C_{in}}{C_{mem}}\right)}{1 + \tau_{rd} \cdot s} \qquad \textbf{Eq. 392}$$

Computing the inverse Laplace-transform to obtain the time response of S&H circuit in the holding phase:

$$v_{bottom} = E \cdot \left(1 + \frac{C_{in}}{C_{mem}}\right) \cdot \left(1 - e^{-\frac{t}{\tau_{rd}}}\right) \cdot u_o(t) \qquad \textbf{Eq. 393}$$

The acquisition error is described by the equation:

$$v_{\varepsilon}(t) = -E \cdot e^{-\frac{t}{\tau_{rd}}} \qquad \textbf{Eq. 394}$$

APPENDIX B – DC ERRORS IN MAC OPERATION

1. OFFSET AND FINITE GAIN OF AMPLIFIER, CHARGE INJECTION AND CLOCK FEEDTHROUGH

In this section, the output voltage at the end of MAC operation is determined considering the non-idealities associated to the operational amplifier (offset and finite gain) and charge-injection and clock-feedthrough of MOS switches.

Figure 5 represents the circuital model used to carry out this study and Figure 6 describes the timing diagram of control signals to develop a MAC operation described by the equation:

$$V_{out} = V_{zero} + \frac{C_{in}}{C_{out}}(V_1 - V_2)$$

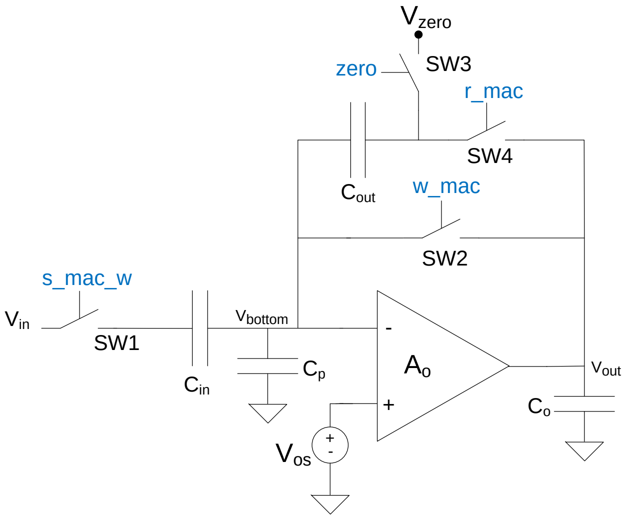


Figure 5.- AMPLIFIER WITH FINITE GAIN AND OFFSET.

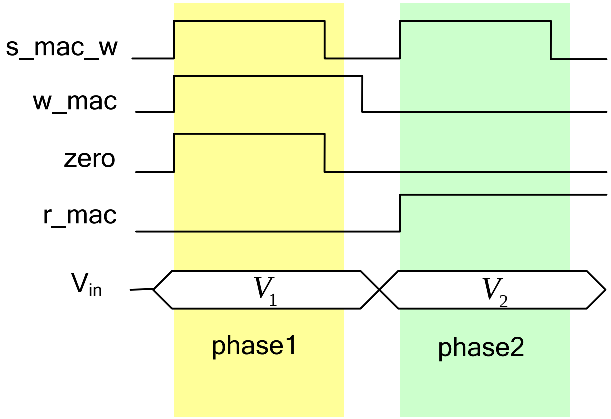


Figure 6.- CONTROL PHASES FOR MAC OPERATION

First, the charge stored in the capacitors of MAC circuit during the two phases of MAC operation will be calculated.

For phase 1 is obtained:

$$Q_{in}\left(phase\_1\right)=C_{in}\left(V_1-\frac{V_{os}}{1+\frac{1}{A_0}}\right)+Q_{SW1\_ph1} \qquad \textbf{Eq. 395}$$

$$Q_{out}\left(phase\_1\right)=C_{out}\left(V_{zero}-\frac{V_{os}}{1+\frac{1}{A_0}}\right)+Q_{SW3} \qquad \textbf{Eq. 396}$$

$$Q_{p}\left(phase\_1\right)=C_{p}\left(-\frac{V_{os}}{1+\frac{1}{A_0}}\right)+Q_{SW2} \qquad \textbf{Eq. 397}$$

Q_{SW1_phi1} is charge introduce by the MOS transistor SW1 because of charge injection and feedthrough effects when it is turned off in phase 1; Q_{SW2} and Q_{SW3} are the charges introduced respectively by MOS transistors SW2 y SW3 at the end of phase 1.

For phase 2 the charge stored in capacitors is given by:

$$Q_{in}\left(phase\_2\right)=C_{in}\left(V_2-V_{os}+\frac{1}{A_o}V_{out}\right)+Q_{SW1\_ph2} \qquad \textbf{Eq. 398}$$

$$Q_{out}\left(phase\_2\right)=C_{out}\left(V_{out}-V_{os}+\frac{1}{A_o}V_{out}\right) \qquad \textbf{Eq. 399}$$

$$Q_{p}\left(phase\_2\right)=C_{p}\left(-V_{os}+\frac{1}{A_o}V_{out}\right) \qquad \textbf{Eq. 400}$$

Q_{SW1_phi2} is the charge injected by switch SW1 at the end of phase 2.

Applying the charge conversion law in the node v_{bottom} :

$$Q_{in}\left(phase\_1\right)+Q_{out}\left(phase\_1\right)+Q_{p}\left(phase\_1\right)=Q_{in}\left(phase\_2\right)+Q_{out}\left(phase\_2\right)+Q_{p}\left(phase\_2\right) \qquad \textbf{Eq. 401}$$

The output voltage at the end of MAC operation considering the non-idealities is determined by the expression:

$$V_{out}=\frac{1}{1+\varepsilon}\cdot\left[V_{zero}+\frac{C_{in}}{C_{out}}\cdot\left(V_1-V_2\right)+\frac{\varepsilon}{1+\frac{1}{A_o}}\cdot V_{os}+\frac{1}{C_{out}}\left(Q_{SW2}+Q_{SW3}+Q_{SW1\_ph1}-Q_{SW1\_ph2}\right)\right] \qquad \textbf{Eq. 402}$$

Being:

$$\varepsilon=\frac{1}{A_o}\left(1+\frac{C_{in}+C_p}{C_{out}}\right) \qquad \textbf{Eq. 403}$$

Previously, the errors in a basic MAC operation have been introduced. This basic operation is used to implement convolutions or spatial filters.

The **MAC** block together with the analogue shifting block permits the implementation of spatial filters defined by 3x3 templates. In this case, each contribution associated to a neighbor requires a multiplication-accumulation operation defined by the term:

$$n\cdot\frac{C_{in}}{C_{out}}\cdot\left(V_i-V_{zero}\right) \qquad \textbf{Eq. 404}$$

The weight or scaled factor is carried out by the programmable factor $\frac{C_{in}}{C_{out}}$ and the accumulation of consecutive multiplication-accumulation process denoted by n.

Due to the circuit topology considered for the implementation of **MAC** block, the scaled factor used during the consecutive multiplication-accumulation process must remain constant. Under this condition, the convolution operation can be implemented as the sum of intermediate values resulting from the multiplication-accumulation process associated to each group of neighbors with the same contribution weight.

Figure 7 shows the sequence of instructions required to perform three consecutive multiplication-accumulation operations in the **MAC** block, described mathematically by:

$$V_{out}=\frac{C_{in}}{C_{out}}\cdot\left[\left(V_1-V_{zero}\right)+\left(V_2-V_{zero}\right)-\left(V_3-V_{zero}\right)\right] \qquad \textbf{Eq. 405}$$

Following, it is calculated the charge stored in the capacitors of MAC circuit during the two phases of MAC operation which corresponds to the second multiplication-accumulation iteration.

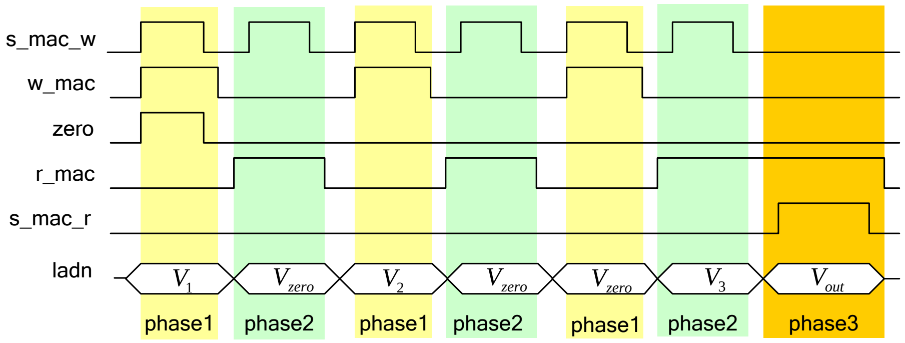


Figure 7.- CONSECUTIVE MULTIPLICATION-ACCUMULATION OPERATIONS.

For phase 1, the charge stored in capacitors of scaled factor and parasitic capacitors is:

$$Q_{in}\left(phase\_1\right)=C_{in}\left(V_1-\frac{V_{os}}{1+\frac{1}{A_0}}\right)+Q_{SW1\_ph1} \qquad \textbf{Eq. 406}$$

$$Q_{out}\left(phase\_1\right)=C_{out}\left(V_{out}\left(1\right)-V_{os}+\frac{1}{A_o}V_{out}\right) \qquad \textbf{Eq. 407}$$

$$Q_p\left(phase\_1\right)=C_p\left(-\frac{V_{os}}{1+\frac{1}{A_0}}\right)+Q_{SW2} \qquad \textbf{Eq. 408}$$

Where $V_{out}(1)$ is the output voltage at the end of phase 2 for first iteration.

Charge stored in phase 2 is given by:

$$Q_{in}\left(phase\_2\right)=C_{in}\left(V_2-V_{os}+\frac{1}{A_o}V_{out}\left(2\right)\right)+Q_{SW1\_ph2} \qquad \textbf{Eq. 409}$$

$$Q_{out}\left(phase\_2\right)=C_{out}\left(V_{out}\left(2\right)-V_{os}+\frac{1}{A_o}V_{out}\left(2\right)\right) \qquad \textbf{Eq. 410}$$

$$Q_{p}\left(phase\_2\right)=C_{p}\left(-V_{os}+\frac{1}{A_o}V_{out}\left(2\right)\right) \qquad \textbf{Eq. 411}$$

Applying the conversion law of charge in the node v_{bottom} :

$$\begin{aligned}V_{out}\left(2\right)=&\frac{1}{1+\varepsilon}\cdot\left[\frac{C_{in}}{C_{out}}(V_1-V_2)+\left(1+\frac{1}{A_o}\right)\cdot V_{out}\left(1\right)\right]+\\&+\frac{1}{1+\varepsilon}\left[\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\frac{\Delta Q_{SW1}}{C_{out}}+\frac{Q_{SW2}}{C_{out}}\right]\end{aligned} \qquad \textbf{Eq. 412}$$

Being:

$$\Delta Q_{SW1}=Q_{SW1\_phi1}-Q_{SW1\_phi2} \qquad \textbf{Eq. 413}$$

And substituting $V_{out}(1)$ it is obtained:

$$\begin{aligned}V_{out}\left(2\right)=&\left(\frac{1}{1+\varepsilon}+\frac{\left(1+\frac{1}{A_0}\right)}{\left(1+\varepsilon\right)^2}\right)\cdot\frac{C_{in}}{C_{out}}\cdot\left(V_1-V_2\right)+\frac{\left(1+\frac{1}{A_o}\right)}{\left(1+\varepsilon\right)^2}\cdot V_{zero}+\frac{\left(1+\frac{1}{A_o}\right)}{\left(1+\varepsilon\right)^2}\cdot\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot V_{os}+\frac{1+\frac{1}{A_o}}{\left(1+\varepsilon\right)^2}\cdot Q_{SW3}+\\&+\left(\frac{1}{1+\varepsilon}+\frac{\left(1+\frac{1}{A_0}\right)}{\left(1+\varepsilon\right)^2}\right)\cdot\frac{\frac{1}{A_o}}{\left(1+\frac{1}{A_o}\right)}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\left(\frac{1}{1+\varepsilon}+\frac{\left(1+\frac{1}{A_0}\right)}{\left(1+\varepsilon\right)^2}\right)\cdot\frac{\Delta Q_{SW1}}{C_{out}}+\left(\frac{1}{1+\varepsilon}+\frac{\left(1+\frac{1}{A_0}\right)}{\left(1+\varepsilon\right)^2}\right)\cdot\frac{Q_{SW2}}{C_{out}}\end{aligned} \qquad \textbf{Eq. 414}$$

Generalizing to n iterations, it is obtained:

$$\begin{aligned}V_{out}\left(n\right)=&\sum_{i=1}^{n}\frac{\left(1+\frac{1}{A_o}\right)^{i-1}}{\left(1+\varepsilon\right)^{i}}\cdot\frac{C_{in}}{C_{out}}\left(V_1-V_2\right)+\frac{\left(1+\frac{1}{A_o}\right)^{n-1}}{\left(1+\varepsilon\right)^{n}}\cdot V_{zero}+\frac{\left(1+\frac{1}{A_o}\right)^{n-1}}{\left(1+\varepsilon\right)^{n}}\cdot\frac{\frac{1}{A_0}}{1+\frac{1}{A_o}}\cdot V_{os}+\frac{\left(1+\frac{1}{A_o}\right)^{n-1}}{\left(1+\varepsilon\right)^{n}}\cdot Q_{SW3}+\\&+\sum_{i=1}^{n}\frac{\left(1+\frac{1}{A_o}\right)^{i-1}}{\left(1+\varepsilon\right)^{i}}\cdot\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\sum_{i=1}^{n}\frac{\left(1+\frac{1}{A_o}\right)^{i-1}}{\left(1+\varepsilon\right)^{i}}\cdot\frac{Q_{SW2}}{C_{out}}+\sum_{i=1}^{n}\frac{\left(1+\frac{1}{A_o}\right)^{i-1}}{\left(1+\varepsilon\right)^{i}}\cdot\frac{\Delta Q_{SW1}}{C_{out}}\end{aligned} \qquad \textbf{Eq. 415}$$

Now, we will rearrange the equation considering the approximation:

$$\frac{1}{1+\varepsilon} \approx 1-\varepsilon \qquad \textbf{Eq. 416}$$

Obtaining:

$$\begin{aligned} V_{out}(n) &= (1-\varepsilon)\cdot\sum_{i=1}^{n}(1-\delta)^{i-1}\cdot\frac{C_{in}}{C_{out}}\cdot(V_1-V_2)+(1-\varepsilon)\cdot(1-\delta)^{n-1}\cdot V_{zero}+(1-\varepsilon)\cdot(1-\delta)^{n-1}\cdot\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot V_{os}+ \\ &+(1-\varepsilon)\sum_{i=1}^{n}(1-\delta)^{i-1}\cdot\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+(1-\varepsilon)\cdot(1-\delta)^{n-1}\cdot Q_{SW3}+ \\ &+(1-\varepsilon)\sum_{i=1}^{n}(1-\delta)^{i-1}\cdot\frac{\Delta Q_{SW1}}{C_{out}}+(1-\varepsilon)\sum_{i=1}^{n}(1-\delta)^{i-1}\cdot\frac{Q_{SW2}}{C_{out}} \end{aligned} \qquad \textbf{Eq. 417}$$

The parameter δ is given by:

$$\delta=\frac{1}{A_o}\cdot\frac{C_{in}+C_p}{C_{out}} \qquad \textbf{Eq. 418}$$

Therefore, the output associated to n consecutive multiplication-accumulation operations of **MAC** block is given by the equation:

$$\begin{aligned} V_{out}(n) &\approx (1-\varepsilon)\cdot\left(\sum_{i=1}^{n}(1-\delta)^{i-1}\right)\cdot\left[\frac{C_{in}}{C_{out}}\cdot(V_1-V_2)+\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot\frac{C_{in}+C_p}{C_{out}}\cdot V_{os}+\frac{\Delta Q_{SW1}}{C_{out}}+\frac{Q_{SW2}}{C_{out}}\right]+ \\ &+(1-\varepsilon)\cdot(1-\delta)^{n-1}\cdot\left(V_{zero}+Q_{SW3}+\frac{\frac{1}{A_o}}{1+\frac{1}{A_o}}\cdot V_{os}\right) \end{aligned} \qquad \textbf{Eq. 419}$$

2. CAPACITANCE MISMATCH

In this section, the effect on the results of MAC operation due to mismatch between capacitors of circuit is studied.

The scale factor in the multiplication-accumulation process is implemented by the ratio between capacitances C_{in}/C_{out}. In the Q-eye system the capacitances C_{in} and C_{out} are programmable:

$$C_{in}=n\cdot C_u \qquad n=1,2 \qquad \textbf{Eq. 420}$$

$$C_{out}=m\cdot C_u \qquad m=1,2 \qquad \textbf{Eq. 421}$$

The MOS capacitors are implemented with unitary capacitors, whose capacitance values change randomly due to mismatch phenomena. The total capacitance is given by the sum of unitary capacitances:

$$C_{in} = \sum_{i=1}^{n} C_{u,i}$$ **Eq. 422**

$$C_{out} = \sum_{j=1}^{m} C_{u,j}$$ **Eq. 423**

The unitary capacitances follow a Gaussian distribution. Therefore, the variances related with the differences of each scale factor capacitance are:

$$\sigma^2\left(\Delta C_{in}\right) = \sum_{i=1}^{n}\left(\frac{\partial C_{in}}{\partial C_{u,i}}\right)^2 \cdot \sigma^2\left(\Delta C_u\right) = \sum_{i=1}^{n}\sigma^2\left(\Delta C_u\right) = n \cdot \sigma^2\left(\Delta C_u\right)$$ **Eq. 424**

$$\sigma^2\left(\Delta C_{out}\right) = \sum_{j=1}^{m}\left(\frac{\partial C_{in}}{\partial C_{u,j}}\right)^2 \cdot \sigma^2\left(\Delta C_u\right) = \sum_{j=1}^{m}\sigma^2\left(\Delta C_u\right) = m \cdot \sigma^2\left(\Delta C_u\right)$$ **Eq. 425**

$$\frac{\sigma^2\left(\Delta C_{in}\right)}{C_{in}^2} = \frac{1}{n} \cdot \frac{\sigma^2\left(\Delta C_u\right)}{C_u^2}$$ **Eq. 426**

$$\frac{\sigma^2\left(\Delta C_{out}\right)}{C_{out}^2} = \frac{1}{m} \cdot \frac{\sigma^2\left(\Delta C_u\right)}{C_u^2}$$ **Eq. 427**

In the previous expressions, we have considered that errors associated to each unitary capacitor are uncorrelated.

Considering the mismatch effect, the variance of scale factor $k = \dfrac{C_{in}}{C_{out}}$ is determined by:

$$\sigma^2\left(\Delta k\right) = \left(\frac{\partial k}{\partial C_{in}}\right)^2 \cdot \sigma^2\left(\Delta C_{in}\right) + \left(\frac{\partial k}{\partial C_{out}}\right)^2 \cdot \sigma^2\left(\Delta C_{out}\right)$$ **Eq. 428**

$$\sigma^2\left(\Delta k\right) = \frac{1}{C_{out}^2} \cdot \sigma^2\left(\Delta C_{in}\right) + \frac{C_{in}^2}{C_{out}^4} \cdot \sigma^2\left(\Delta C_{out}\right)$$ **Eq. 429**

$$\frac{\sigma^2\left(\Delta k\right)}{k^2} = \frac{1}{C_{in}^2} \cdot \sigma^2\left(\Delta C_{in}\right) + \frac{1}{C_{out}^2} \cdot \sigma^2\left(\Delta C_{out}\right)$$ **Eq. 430**

Therefore, the scale factor value is given by:

$$k + \sigma\left(\Delta k\right) \approx \frac{C_{in}}{C_{out}} + \sqrt{\frac{C_{in}^2}{C_{out}^2} \cdot \frac{\sigma^2\left(k\right)}{k^2}}$$ **Eq. 431**

$$k + \sigma\left(\Delta k\right) = \frac{n}{m} \cdot \left(1 + \sqrt{\frac{1}{n} \cdot \frac{\sigma^2\left(\Delta C_u\right)}{C_u^2} + \frac{1}{m} \cdot \frac{\sigma^2\left(\Delta C_u\right)}{C_u^2}}\right)$$ **Eq. 432**

$$k + \sigma\left(\Delta k\right) = \frac{n}{m}\left(1 + \sqrt{\left(\frac{1}{n} + \frac{1}{m}\right) \cdot \frac{\sigma^2\left(\Delta C_u\right)}{C_u^2}}\right)$$ **Eq. 433**

This equation means that the mismatch between the capacitors of MAC circuit generates a gain error respect to the nominal value of scale factor.

The Pelgrom’s model describes the mismatch error associated to a process parameter P between two devices by the equation:

$$\sigma^2(\Delta P) = \frac{A_P^2}{W \cdot L} + S_P^2 \cdot D_x^2 \qquad \textbf{Eq. 434}$$

$\sigma^2(\Delta P)$ is the variance of the difference of parameter P between the elements, A_P and S_P are the area and spacing proportionality technological constants for parameter P, W and L are the dimensions of each element and D_x is the spacing between them. If the elements are laid out in close proximity with a layout style that eliminates most sources of systematic errors (e.g., common centroid), mismatch is mostly dominated by the $A_P^2 / W \cdot L$ term.

In this case, the capacitance associated to a MOS unitary capacitor is:

$$C_u = C_{ox} \cdot W_u \cdot L_u \qquad \textbf{Eq. 435}$$

And the variance of the difference related with the technological parameter C_{ox} is:

$$\sigma^2(\Delta C_{ox}) = \frac{A_{Cox}^2}{W_u \cdot L_u} \qquad \textbf{Eq. 436}$$

Therefore, the variance of the difference of the unitary capacitance is determined by the equation:

$$\frac{\sigma^2(\Delta C_u)}{C_u^2} = \frac{A_{Cox}^2}{C_{ox}^2} \cdot \frac{1}{W_u \cdot L_u} \qquad \textbf{Eq. 437}$$

And the scale factor is given by the equation:

$$k + \sigma(\Delta k) = \frac{n}{m}\left(1 + \sqrt{\left(\frac{1}{n} + \frac{1}{m}\right) \cdot \frac{\sigma^2(\Delta C_u)}{C_u^2}}\right) \qquad \textbf{Eq. 438}$$

The mismatch effect between capacitors decreases when their areas increase. The reduction of gain error generated by the mismatch between capacitors involves an increment in power consumption to maintain the operating speed of MAC circuit, because the polarization current of circuit needs to be increased. Moreover, larger capacitors imply to increase el pixel area reducing the effective spatial resolution. There is a tradeoff between operation accuracy and power consumption; and between accuracy and spatial resolution of smart sensor.

REFERENCES

[Abbo08] A. A. Abbo et al. "Xetal-II: A 107 GOPS, 600 mW Massively Parallel Processor for Video Scene Analysis". *IEEE J. Solid-State Circuits*, Vol. 43, No. 1, pp. 192-201, Jan. 2008.

[Adle97] V. Adler and E. G. Friedman, "Delay and Power Expressions for a CMOS Inverter Driving a Resistive-Capacitive Load". *Analog Integrated Circuits and Signal Processing*, Vol. 14, no. ½, pp. 29-39, September 1997.

[Agra95] S. Agrawal and R. Gupta, *System Partitioning Using Global Data-Flow*. Univ. Illinois Memo, UIUC DCS, 1995.

[Aho88] A. Aho et al., *Compilers: Principles, Techniques and Tools*. MA: Addison-Wesley, 1988.

[Akah06] N. Akahane et al., "A Sensivity and Linearity Improvement of a 100-dB Dynamic Range CMOS Image Sensor Using a Lateral Overflow Integration Capacitor". *IEEE Journal of Solid-State Circuits*, Vol. 41, April 2006.

[Anaf1v3] Anafocus (Innovaciones Microelectronicas S.L.), "Eye-RIS v1.3 Hardware Description". *Eye-RIS v1.3 Use's Manual*.

[Anaf2v1a] Anafocus (Innovaciones Microelectronicas S.L.), "Eye-RIS v2.1 CFPP Code User Manual". *Eye-RIS v2.1 User's Manual*.

[Anaf2v1b] Anafocus (Innovaciones Microelectronicas S.L.), "Eye-RIS v2.1 Hardware Description". *Eye-RIS v2.1 User's Manual*.

[Anaf2v1c] Anafocus (Innovaciones Microelectronicas S.L.), "Eye-RIS v2.1 EBL Reference". *Eye-RIS v2.1 User's Manual*.

[Anaf2v1d] Anafocus (Innovaciones Microelectronicas S.L.), "Eye-RIS v2.1 EIPL Reference". *Eye-RIS v2.1 User's Manual*.

[Anaf2v1e] Anafocus (Innovaciones Microelectronicas S.L.), "Eye-RIS v2.1 ADK Description". *Eye-RIS v2.1 User's Manual*.

[ANAFOCUS] www.anafocus.com.

[Ando02] K.J. Ando, "Overview of Astronomy Arrays at Raytheon Infrared Operations". *SPIE Conference on Detectors in Astronomy*, Hawai, June 2002.

[Ando97] F. Andoh et al., " A 250,000 Pixel Image Sensor with FET Amplification at Each Pixel for High Speed Televison Cameras". *IEEE J. Solid-State Circuits*, vol. SC-32, pp. 187-198, 1997.

[Bai00] Y. Bai and J. Montroy, "Development Of Hybrid CMOS Visible Focal Plane Arrays At Rockwell". *Proceedings of the SPIE*, Vol. 4028, pp. 174, Orlando, April 2000.

[Ball81] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes". *Pattern Recognition*, Vol. 13, pp. 111-122, 1981.

[Banb99] H. Banba et al., "A CMOS Bandgap Reference Circuit with Sub-1-V Operation", *IEEE Journal of Solid-State Circuits*, Vol. 34, No.5, May 1999.

[Barr94] E. Barros et al., "A Method For Partitioning UNITY Language In Hardware And Software". *Proc. EURODAC*, pp. 220-225, 1994.

[Belb09] A.N. Belbachir, *Smart Cameras*. Springer, ISBN:978-1-4419-4419-0952-7, 2009.

[Bend96] A. Bender, "Design of an optimal loosely coupled heterogeneous multiprocessor system". *Proc. EDTC*, pp. 275-281, 1996.

[Berm04] A. Bermak et al., "A Low Power Current-Mode Pixel with On-Chip FPN Cancellation and Digital Shutter". *Proceedings of IEEE ISCAS'04*, Vol.2, pp.345-348, May 2004.

[Bore98] G D. Boremann, "Basic Electro-Optics For Electrical Engineers". *Tutorial Tests in Optical Engineering, SPIE Optical Engineering Press*, ISBN:0-8194-2806-X, 1998.

[Cann86] J. Cannis, "A Computational Approach To Edge Detection". *IEEE Transaction Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 679-698, Nov. 1986.

[Carm01] R. Carmona-Galán, *Analysis and Design Techniques for VLSI Hardware for Real-Time Image Processing*. PhD Dissertation, Univ. Sevilla, 2001.

[Carm03] R. Carmona et al., "A Bio-Inspired 2-Layer Mixed-Signal Mixed-Signal Flexible Programmable Chip for Early Vision". *IEEE Transactions on Neural Networks*, Vol. 14, No.5, pp. 1313-1336, September 2003.

[Carm13] R. Carmona-Galán et al., "A Hierarchical Vision Processing Architecture Oriented to 3D Integration of Smart Camera Chips*". Journal of System Architectures* (special issue on *Smart Camera Architectures*), **2013.** Digital Object Identifier 10.1016/j.sysarc.2013.03.002

[Cauw96] G. Cauwenberghs, "An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory". *IEEE Transactions on Neural Networks*, Vol. 7, No. 2, pp. 346-361, March 1996.

[CCIR90] CCIR, *Rec. 601-2: Encoding Parameters of Digital Television for Studios*. Recommendations of the international Consultative Committee for Radio-Communications (CCIR). Vol. XI, pp. 95-104, Geneva, Switzerland 1990.

[Chen12] Y. Chen et al., "Column-Parallel Digital Correlated Multiple Sampling for Low-Noise CMOS Image Sensors". *IEEE Sensors Journal*, Vol. 12, pp. 793-799, 2012.

[Cho95] D.H. Cho et al., "An Accurate Intrinsic Capacitance Modeling for Deep Submicrometer MOSFET's". *IEEE Transactions on Electron Devices*, Vol. 42, No. 3, March, 1995.

[Chua02] L.O. Chua and T. Roska, *Cellular Neural Networks and Visual Computing*. Cambridge University Press, Cambridge UK, 2002.

[Chua88] L.O. Chua and L. Yang, "Cellular Neural Network: Theory". *IEEE Transactions on Circuits and Systems*, Vol.35, pp. 1257-1272, No. 10, October 1988.

L.O. Chua and L. Yang, "Cellular Neural Networks: Applications". *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 35, No. 10, pp. 1273-1290, October 1988.

[Chua91] L.O. Chua and P. Thiran, "An Analytic Method for Designing Simple Cellular Neural Networks". *IEEE Transactions on Circuits and Systems*, Vol. 38, No. 11, pp. 1332-1341, November 1991.

[Chua93] L. O. Chua and T. Roska, "The CNN Paradigm". *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 40, No.3, pp. 147-156, March 1993.

[Chul10] C.B. Kim et al., "In-Pixel Edge Detection Circuit without Non-uniformity Correction for Infrared Focal Plane Array (IRFPA)". *IEICE Transactions on Electronics*, Vol. E91.C, Issue 2, pp. 235-239, 2010.

[COGNEX] Cognex Ltd, http://www.cognex.com/ProductsServices/InspectionSensors.

[Davi12] E. R. Davies, "Computer and Machine Vision: Theory, Algorithms Practicalities"-, Fourth edition, Ed. Elsevier, 2012.

[Deck98] S. J. Decker et al., "A 256x256 CMOS Imaging Array with Wide Dynamic Range Pixels and Column-parallel Digital Output". *IEEE J. of Solid State Circuits*, Vol. 33, pp. 2081-2091, Dec. 1998.

[Delb06] T. Delbruck and P. Lichsteiner, "Freeing Vision from Frames". *The Neuromorphic Engineer*, Vol. 3, pp. 3-4, May 2006.

[DeMi97] G. De Micheli, *Hardware-Software Codesign*. Special Issue. *Proc. of IEEE*, Vol. 85, No. 3, March 1997.

[Deva08] G. Devaraj et al., "Applying Algorithms". *Vision System Design*, Vol. 13, No. 11, pp. 17-20 and 85-87, Nov. 2008.

[Domi97] R. Domínguez-Castro et al., "A 0.8um CMOS Programmable Mixed-Signal Focal-Plane Array Processor with On-Chip Binary Imaging and Instructions Storage". *IEEE Journal of Solid State Circuits*, Vol. 32, No. 7, pp. 1013-1026, July 1997.

[Domi98] R. Domínguez-Castro et al., "Four-Quadrant One-Transistor Synapse for High Density CNN Implementations". *Proc. of the Fifth IEEE International Workshop on Cellular Neural Networks and their Applications*, pp. 243-248, London, UK, April 1998.

[Duda72] R.O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures". *Comm. ACM*, Vol. 15, pp. 11-15, January, 1972.

[Dunc90] R. Duncan, R. "A Survey of Parallel Computers Architectures". *IEEE Computer*, Vol. 23, Issue 2, pp. 5-16, 1990.

[Eklu96] J.E. Eklund et al., "VLSI Implementation of a Focal Plane Image Processor-A Realization of the Near-Sensor Image Processing Concept". *IEEE Trans. VLSI*, Vol.4, no.3, pp.322-335, Sept. 1996.

[EImg14] *Proceeding of Image Sensors and Imaging Systems 2014*. SPIE Electronic Imaging 2014. San Francisco, February 2014.

[EMVA] "EMVA Standard 1288". *Standard for Characterization and Presentation of Specification Data for Image Sensors and Cameras*, Release A2.01.

[Enz96] C.C. Enz and G.C. Temes, "Circuit Techniques for Reducing the Effects of Op-Amp Imperfections: Autozeroing, Correlated Double Sampling and Chopper Stabilization". *Proc. of the IEEE*, Vol. 84, No. 11, Pg. 1584-1614, Nov. 1996.

[Espe94a] S. Espejo et al., "Smart-Pixel Cellular Neural Networks in Analog Current-Mode CMOS Technology". *IEEE Journal of Solid-State Circuits*, Vol. 29, pp. 895-905, IEEE August 1994.

[Espe94b] S. Espejo, *Cellular Neural Network: VLSI Design and Modeling*. Ph. D. Thesis, Univ. Sevilla, 1994.

[Fern11] J. Fernández-Berni et al., "FLIP-Q: A QCIF Resolution Focal-Plane Array for Low-Power Image Processing". *IEEE J. of Solid-State Circuits*, Vol. 46, No. 3, pp. 669-680, March 2011.

[Fern12] J. Fernández-Berni et al., *Low-Power Smart Imagers for Vision-Enabled Sensor Networks*. ISBN 978-1-4614-2391-1, Springer, May 2012.

[Fiet12] R.D. Fiete. *Modeling the Imaging Chain of Digital Cameras*. SPIE Press 2012.

[Fisc82] J.H. Fischer, "Noise Sources and Calculation Techniques for Switched Capacitor Filters". *IEEE Journal of Solid-State Circuits*, Vol. 17, No. 4, pp. 742-752, August 1982.

[Flyn99] M. J. Flynn, "Very High Speed Computing Systems". *Proceedings of the IEEE*, Vol. 54, pp. 1901-1909, December 1999.

[Font11] R. Fontaine, "A Review of the 1.4μm Pixel Generation". *Int. Image Sensor Workshop*, Hokkaido, Japan, June 2011.

[Foss92] E. R. Fossum, "Active pixel sensors (APS). Are CCDs dinosaurs?", Proc. SPIE, vol. 1990, pp. 2-14, 1992. ([7],Chapter 1)

[Foss97] E. Fossum, "CMOS Image Sensors: Electronic Camera-On-Chip". *IEEE Transactions on Electron Devices*, Vol. 44, Issue 10, pp.1689-1698, October 1997.

[Fowl98] B. Fowler et al., "Techniques for Pixel Level Analog to Digital Conversion", *SPIE Conference on Infrared Readout Electronics IV*. 1998.

[Fox98] E. C. Fox, "A High Speed Linear CCD Sensor With Pinned Photo-Diode Photosite For Low Lag And Low Noise Imaging". *Proc. SPIE*, vol.3301, p.17, 1998.

[Gama02] A. El Gamal, "High Dynamic Range Image Sensors". Tutorial at the *IEEE International Solid-State Circuits Conference*, 2002.

[Gama05] A. El Gamal and H. Eltoukhy, "CMOS Image Sensors". *IEEE Circuits and Devices Magazine*, pp. 6-20, June 2005.

[Gama98] A. El Gamal et al., "Modeling and Estimation of FPN Components in CMOS Image Sensors". *Proc. SPIE Solid State Sensor Array: Development and Applications II*, April 1998.

[Garr08] P. Garrou, C. Bower and P. Ramm, *Handbook of 3D Integration*. Wiley VCH 2008.

[Gend91] Y. Gendai et al., "An 8-b 500-MHz ADC", *Proc. IEEE 1991 Int. Solid-State Circuits Conf*., pp. 172-173, San Francisco, Feb. 1991.

[Gilb68] B. Gilbert, "A Precise Four-Quadrant Multiplier with Subnano-Second Response". *IEEE Journal of Solid-State Circuits*, Vol.3, No.4, pp. 365-373, December 1968.

[Gobe83] C. A. Gobet and A. Knob, "Noise Analysis of Switched Capacitor Networks". *IEEE Transactions on Circuits and Systems*, Vol. CAS-30, No. 1, Pg. 37-43, Jan.1983.

[Gonz92] R.C. González and R.E. Woods, *Digital Image Processing*. Prentice Hall.

[Gonz15] R.C. González et al., *Digital Image Processing Using MATLAB - 2nd Ed*. Gatesmark Publishing, 2015.

[Goos97] G. Goossens et al., "Embedded software in real-time signal processing systems: Design technologies". *Proc. IEEE*, Vol.85, pp. 436-454. April 1997.

[Greg94] R. Gregorian and G.C. Themes, *Analog MOS Integrated Circuits for Signal Processing*. John Wiley & Sons, New York, 1994.

[Guck83] J. Guckenheimer and P Holmes, *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. Springer-Verlag, New York, 1983.

[Gupt95] R. K. Gupta, *Co-Synthesis of Hardware and Software for Digital Embedded Systems*. Boston: Kluwer Academic Publishers, 1995.

[Hast01] A. Hastings, *The Art of Analog Layout*. Prentice Hall, 2001.

[Henn06] J. Hennessy and D. Patterson. *Computer Arquitecture: A Quantitative Approach*. Morgan Kaufmann 2006.

[Hira96] A. Hirata et al., "Estimation of Short-Circuit Power Dissipation for Static CMOS Gates". *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Science*, Vol. E79-A, no. 3, pp. 304-311, March 1996.

[Hornbe] A. Hornberg, *Handbook of Machine Vision*. Wiley-VCH, 2010, ISBN 9783527405848.

[Indi99] G. Indivery, "Neuromorphic Analog VLSI Sensor For Visual Tracking: Circuits And Application Examples". *IEEE Trans. Circuits and Systems II*, vol. 46(11), pp. 1337-1347, November 1999.

[IS14] *Proceeding of Image Sensors 2014*. London, March 2014.

[ISORG] www.isorg.fr

[ISW13] *Proceedings 2013 Image Sensing Workshop*. Salt-Lake City, June 2013.

[ITRS13] *2013 International Technology Roadmap for Semiconductors*. http://www.itrs.net/

[Jane02] J. Janesick, "Scientific CMOS Pixels". *SPIE Conference on Detectors in Astronomy,* Hawai, June 2002.

[Jane03] J. Janesick, "Charge Coupled CMOS And Hybrid Detector Arrays". *SPIE Astronomy Optics Instrum. Conf/Focal Plane Arrays for Space Telescopes* (AM119), San Diego, CA, August 2003.

[Jehy03] J. Rhee and Y. Joo, "Wide Dynamic Range CMOS Image Sensor with Pixel Level ADC". *Electronics Letters,* vol. 39, No.4, pp. 360-361, 20th February, 2003.

[John97] D.A. Johns and K. Martin, *Analog Integrated Circuit Design*. John Wiley & Sons, 1997.

[Kana07] A. Kananen, *A VLSI Array Processor Architecture For Emulating Resistive Network Filtering.* PhD Dissertation, Helsinki University of Technology, 2007.

[Kass99] A. A. Kassim et al., "A comparative study of efficient Generalized Hough transform techniques". *Image and Vision Computing*, Vol. 17, Issue 10, pp. 737-748, August 1999.

[Kawa08] S. Kawahito et al., "A CMOS Image Sensor Integrating Column-Parallel Cyclic ADCs with On-Chip Digital Error Correction Circuits". *IEEE Int. Solid-State Circuits Conf.*, pp 56-595, Feb. 2008.

[Khro13] Khronos group, *Openvx 1.0, Hardware Acceleration API For Computer Vision Applications and Libraries*. http://www.khronos.org/openvx, 2013, [Online; accessed 08-March-2013].

[Klei01] S. Kleinfelder et al., "A 10,000Frames/s 0.18um CMOS Digital Pixel Sensor with Pixel-Level Memory". *2001 International Solid State Circuits Conference*, February 2001.

[Knap96] D. Knapp, *Behavioral Synthesis*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[Koch95] C. Koch and H. Li (Editors), *Vision Chips: Implementing Vision Algorithms with Analog VLSI Circuits*. IEEE Computer Society Press, Los Alamitos, 1995.

[Koze93] T. Kozek et al., "Genetic Algorithm for CNN Template Learning". *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 40, No. 6, pp. 392-402, June1993.

[Ku92] D. Ku and G. De Micheli, "Relative Scheduling Under Timing Constraints: Algorithms For High Level Synthesis Of Digital Circuits". *IEEE Trans. CAD/ICAS*, pp. 696-718, June 1992.

[Kubo82] S. Kuboki, "Nonlinearity Analisys of Resistors String A/D Converters". *IEEE Trans. Circuits Syst.*, vol. CAS-29, pp. 383-389, June 1982.

[Leñe14] J.A. Leñero-Bardallo et al., "Review of ADCs for Imaging". *Image Sensors and Imaging Systems 2014-SPIE Electronic Imaging 2014,* January 2014.

[Liña02a] G . Liñán, *Design of Low-Power, Programmable Mixed-Signal Chips for Real-Time Vision*. PhD Dissertation, Univ. Sevilla 2002.

[Liña02b] G . Liñán et al., "ACE4k: An Analog I/O 64x64 Visual Microprocessor Chip with 7-bit Analog Accuracy". *International Journal of Circuit Theory and Applications*, Vol. 30, pp. 89-116, Wiley March-June 2002.

[Liña04] G. Liñán et al., "A 1000FPS@128x128 Vision Processor with 8-Bit Digitized I/O". *IEEE Journal of Solid-State Circuits*, Vol. 39, No. 7, pp. 1044-1055, July 2004.

[Liña98] G. Liñan et al., "A 64x64 CNN with Analog and Digital I/O". *Proc. of the IEEE International Conference on Electronics, Circuits and Systems*, pp. 203-206, Lisbon, Portugal, September 1998.

[Lopi11] A. Lopich and P. Dudek, "A SIMD Cellular Processor Array Vision Chip with Asynchronous Processing Capabilities". *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 58, No. 10, pp. 2420-2431, Oct. 2011.

[Malo01] F. Maloberti, *Analog Design for CMOS VLSI Systems.* Kluwer Academic Publishers, 2001.

[MATROX] Matrox Imaging, http://www.matrox.com/imaging/en/products/software

[Mats90] T. Matsumoto et al., "CNN Cloning Templete: Connected Component Detector". *IEEE Transaction Circuits and Systems*, Vol. 37, pp. 633-635, May 1990.

[Maxi01] A. Maxim and M. Gheorghe, "A Novel Physical Based Model of Deep-Submicron CMOS Transistors Mismatch for Monte Carlo SPICE Simulation". *Proc. IEEE ISCAS 2001,* Vol. 5, pp. 511-514, May 2001.

[Meij01] G.C.M. Meijer (Editor), *Smart Sensor Systems*. John Wiley & Sons, Chichester 2001.

[Moha94] M. Ismail and T. Fiez. *Analog VLSI: Signal and Information Processing*. McGraw-Hill, New York, 1994.

[Naka06] J. Nakamura (editor), *Image Sensors and Signal Processing for Digital Still Cameras*. Taylor & Francis 2006.

[Nicl05] C. Niclass, A. Rochas, P. Besse and E. Charbon, "Design and characterization of a CMOS 3-D image sensor based on single photon avalanche diodes", IEEE Journal of Solid-State Circuits, Vol. 40, Issue 9, pp. 1847-1854, September 2005.

[Nicl11] C. Niclass et al., "A 100m-Range 10-Frame/S 340×96-Pixel Time-Of-Flight Depth Sensor in 0.18µm CMOS". *Proc. of the European Solid-State Circuits Conference*, pp. 107-110, Sept. 2011.

[NIOSII] *NIOS-II Processor Reference Handbook*. ALTERA website, http://www.altera.com.

[Nobl68] P. Noble, "Self-Scanned Image Detector Arrays". *IEEE Trans. Electron Devices*, vol. ED-15, p.202,1968.

[Noss96] J. A. Nossek, "Design and Learning with Cellular Neural Networks". *International Journal of Circuit Theory and Applications*, Vol. 24, pp. 15-24, 1996.

[Ohta08] J. Ohta, *Smart CMOS Image Sensors And Applications*. CRC Press, 2008.

[Pard96] F. Pardo, J.A. Boluda, J.J. Perez, S. Felici, B. Dierick i and D. Scheffer, "Response properties of a foveated space-variant CMOS image sensor", ISCAS'96, Vol. 1, pp. 373-376, May 1996.

[Paul96] Paul C. Yu and Hae-Seung Lee, " A 2.5V, 12-b, 5MSample/s Pipelined CMOS ADC", IEEE J. Solid-State Circuits, Vol. 31, No.12, December 1996. ([11], Chapter 1)

[Payb14] A. Paybe et al., "A 512×424 CMOS 3D Time-of-Flight Image Sensor with Multi-Frequency Photo-Demodulation up to 130MHz and 2GS/s ADC". *IEEE Int. Solid-State Circuit Conference*, February 2014.

[Pelg13] M. Pelgrom et al., "Nyquist AD Converters, Sensor Interfaces, and Robustness". Chapter 13 - *A Designer's View on Mismatch*, pg. 245-267, Springer, 2013.

[Pelg89] M. Pelgrom et al., "Matching Properties of MOS Transistors". *IEEE Journal of Solid-State Circuits*, vol. 24, No. 5, October 1989.

[Pfei10] U. Pfeiffer and E.Öjefors, "Terahertzh CMOS/BiCMOS Process Technologies". *Proc. of the 36th European Solid-State Circuits Conference* (IEEE ESSCIRC 2010), pp. 52-60, Sept. 2010.

[Pirs98] P. Pirsch and H. J. Stolberg, "VLSI Implementation of Image and Multimedia Processing Systems". *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 8, No. 7, pp. 878-891, November, 1998.

[Plas94] R. Van de Plassche, *Integrated Analog-to-Digital And Digital-to-Analog Converters*. Kluwer Academic Publishers, 1994.

[Prov11] J. Provoost and P. de Moor, "Backside-Illuminated Image Sensors: Optimizing Manufacturing for A Sensitivity Payoff". *Vision Systems Design*, November, 2011.

[Rama94] P. Ramanathan and J. Stankovic, "Scheduling Algorithms And Operating System Support For Real-Time Systems". *Proc. IEEE*, Vol. 82, pp. 6-24, January 1994.

[Raza01] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill International Edition, 2001.

[Raza95] B. Razavi, *Principles of Data Conversion System Design*. IEEE Press, 1995.

[Rodr93] A. Rodríguez-Vázquez et al., "Current Mode Techniques for the Implementation of Continuous and Discrete-Time Cellular Neural Networks". *IEEE Transactions on Circuits and Systems*, Vol. 40, pp. 132-146, IEEE March 1993.

[Rodr02] A. Rodríguez-Vázquez et al., "Trade-Offs in the Design of CMOS Comparators". in *Trade-offs in Analog Circuit Design*. *The Designer's companion*, pp. 407-440, Ed. Kluwer Academic Publishers, 2002.

[Rodr03] A. Rodríguez-Vázquez et al., "Mismatch-Induced Trade-offs and Scalability of Analog Preprocessing Visual Microprocessor". *Analog Integrated Circuits and Signal Processing*, Vol. 37, pp. 73-83, Kluwer Academics 2003.

[Rodr04] A. Rodríguez-Vázquez et al., "ACE16k: The Third Generation of Mixed-Signal SIMD-CNN ACE Chips Toward VSoCs". *IEEE Transactions on Circuits and Systems-I*, Vol. 51, No. 5, pp. 851-863, May 2004.

[Rodr10a] A. Rodríguez-Vázquez et al., "A CMOS Vision System On-Chip with Multicore Sensory Processing Architecture for Image Analysis above 1,000F/s". *SPIE Conference Sensors, Cameras, and Systems for Industrial/Scientific Applications* XI, January 2010.

[Rodr10b] A. Rodríguez-Vázquez et al., "A CMOS Vision System On-Chip with Multi-Core, Cellular Sensory-Processing Front-End". Chapter 6 in *Cellular Nanoscale Sensory Wave Computers* (edited by C. Baatar, W. Porod and T. Roska). Springer 2010.

[Rodr14] A. Rodríguez-Pérez et al, "A 515nw, 0-18 db Programmable Gain Analog-to-Digital Converter for In-Channel Neural Recording Interfaces". *IEEE Trans. on Biom. Circ. and Systems*, vol. 8, no. 3, pp. 358–370, 2014.

[Rodr15] A. Rodríguez-Vázquez, "ADCs for Imaging Applications - Comparative Overview and Some Improved Architectures". Plenary Lecture at *Image Sensors 2015,* London 2015.

[Rosk01] T. Roska and A. Rodríguez-Vázquez, *Towards the Visual Microprocessor*. John Wiley and Sons, Chichecter UK, 2001.

[Rosk90] T. Roska and L. O. Chua, "Cellular Neural Networks with Nonlinear and Delay-Type Template Elements". *Proceedings of the First IEEE International Workshop on Cellular Neural Networks and Their Applications*, pp. 12-25, Budapest 1990.

[Rosk93] T. Roska and L.O. Chua, "The CNN Universal Machine: An Analogic Array Computer". *IEEE Transactions on circuits and Systems-II: Analog and Digital Signal Processing*, Vol.40, No.3, pp. 163-173, March 1993.

[Rosk95] T. Roska and L.Kek, "Analogic CNN Program Library (Version 6.1)". *Report DNS-6-1995, Analogical and Neural Computing Laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences*, Budapest, 1995.

[Russ92] J.C. Russ, *The Image Processing Handbook*. CRC Press 1992.

[Saku90] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model And its Applications to CMOS Inverter Delay and Other Formulas". *IEEE Journal of Solid-State Circuits*, Vol. 25, no. 2, pp. 584-594, April 1990.

[Sedr04] A.S. Sedra and K.C. Smith, *Microelectronic Circuits*. Oxford University Press, 2004.

[Seit00] P. Seitz, "Solid-State Image Sensing". *Handbook of Computer Vision and Applications*, vol. 1, pp. 165-222, Academic Press, 2000.

[Seit11] P. Seitz and A.J.P. Theuwisen (Ed.), *Single Photon Imaging*. Springer 2011.

[Serr95] T. Serrano and B. Linares-Barranco, "A Modular Current-Mode High-Precision Winner-Take-All Circuit". *IEEE Trans. Circuits and Systems II*, vol. 42(2), pp. 132-134, February 1995.

[Shap01] L. Shapiro G. and Stockman, *Computer Vision*. Prentice-Hall, Inc. 2001.

[Shi05] Y. Shi and S. Lichman, "Smart Cameras: A Review". *Proc. of Asia-Pacific Workshop on Visual Information Processing,* pp. 95-100, December 2005.

[Shin94] K. Shin and P. Ramanathan, "Real-time computing: A new discipline of computer science and engineering". *Proc. IEEE*, Vol. 82, pp. 6-24, January 1994.

[Shyu82] J. B. Shyu et al., "Random Errors in MOS Capacitors". *IEEE J. Solid-State Circuits*, Vol.17 Pg. 1070-1076, 1982.

[Sira91] E. Siragusa and I. Galton, "A Digitally Enhanced 1.8V 15-bit 40-MSamples/s CMOS Pipelined ADC". *IEEE J. Solid-State Circuits*, Vol. 39, No.12, December 2004.

[Snoe06] M.F. Snoeij et al., "A CMOS Imager With Column-Level ADC Using Dynamic Column Fixed-Pattern Noise Reduction". *IEEE Journal of Solid-State Circuits*, Vol. 41, Pg. 3007-3015, Dec. 2006.

[Socl85] S. Soclof, *Analog Integrated Circuits*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[Stey93] M. Steyaert et al., "A 100-MHz 8-bit CMOS Interpolating A/D Converter". *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 28.1.1 – 28.1.4, San Diego, May 1993.

[Szel11] Richard Szeliski, "Computer Vision: Algorithms and Applications", Ed. Springer, 2011.

[Taka13] I. Takayanagi and J. Nakamura, "High-Resolution CMOS Video Image Sensors". *Proceedings of the IEEE*, Vol. 101, No. 1, pp. 61-73, Jan. 2013.

[Theu95] A.J.P- Theuwisen, *Solid-State Imaging with Charge-Coupled Devices*. Kluwer Academic Publishers 1995.

[Tian01] H. Tian et al.l, "Analisys of Temporal Noise in CMOS Photodiode Active Pixel Sensor". *IEEE Journal of Solid-State Circuits*, Vol. 36, No. 1, January 2001.

[Toma06] C. Tomasi, "Early Vision", *Encyclopedia Of Cognitive Science* (edited by bL. Nadel). John Wiley & Son, 2006.

[TOSHIBA] http://www.toshiba-teli.co.jp/en/products/industrial/sps/sps.htm

[Toum93] C. Toumazou, J. B. Hughes and N. C. Battersby (Eds.), "Switched-Currents: An Analogue Technique for Digital Technology", Peter Peregrinus, London, England, 1993. ([49],Chapter 1)

[Tu98] N. Tu et al., "CMOS Active Pixel Image Sensor with Combined Linear and Logarithmic Mode Operation". *IEEE Canadian Conf. on Electrical and Computer Engineering,* Vol. 2, May 1998.

[USBa] http://www.quickusb.com/media/QuickUSB_Datasheet_Rev_B.pdf.

[USBb] http://www.quickusb.com/media/QuickUSB_User_Guide_v2.14.3.pdf.

[USBc] CYPRESS perform, "EZ-USB FX2LPTM USB Microcontroller", CY7C68013A/14A/15A/16A.

[Vahi94] F. Vahid, J. Gong and D. Gajski, "A binary-constraint search algorithm for minimazing hardware during hardware/software partitioning", Proc. EURODAC, pp. 214-219, 1994. ([16],Chapter 4)

[Wan11] G. Wan et al., "A Dual In-Pixel Memory CMOS Image Sensor for Computation Photography". *2011 IEEE VLSI Circuits Symposium* (, pp. 94-95.

[Weck67] G. Wecklers, "Operation of p-n Junction Photodetectors In A Photon Flux Integrating Mode". *IEEE J. Solid-State Circuits*, Vol. SC-2, p.65, 1967.

[Wegm87] G. Wegmann et al., "Charge Injection In Analog MOS Switches". *IEEE Journal of Solid-State Circuits*, Vol. 22, Pg. 1091-1097, Dec. 1987.

[Weiz02] X. Weize and E.G. Friedman,"Clock feedthrough in CMOS analog transmission gate switches". *Proc. IEEE International ASIC/SOC Conference*, Pg. 181-185, Sept. 2002.

[Wieg89] R. J. Wiegerink et al., “Offset cancelling circuit”. *IEEE J. Solid-State Circuit*, Vol. 24, Pg. 651-658, June 1989.

[Wolf92] W. Wolf, and E. Frey, “Tutorial on embedded system design”, Proc. ICCD, pp. 18-21, October 1992. ([20],Chapter 4)

[Wolf94] W. Wolf, “Hardware-Software Co-Design Of Embedded Systems”. *Proc. IEEE*, Vol. 82, pp. 967-989, July 1994.

[Yadi91] O. Yadid-Pecht et al., “A Random Access Photodiode Array For Intelligent Image Capture”. *IEEE J. Solid-State Circuits*, vol. SC-26, pp. 1116-1122, 1991.

[Yadi93] O. Yadid-Pecht, “The Automatic Wide Dynamic Range Sensor”. *Proc. 1993 SID Int. Symp.*, Seattle, Washington, pp. 495-498, May 18-20, 1993.

[Yang94] W. Yang, “A Wide-Dynamic-Range Low Power Photosensor Array”. *Proc. IEEE 1994 ISSCC*, vol. 37, pp. 230-231, 1994.

[Yang11] Yang Ni, YiMing Zhu and Bogdan Arion “ A 768x576 Logarithmic Image Sensor with Photodiode in Solar Cell mode” 2011 INTERNATIONAL IMAGE SENSOR WORKSHOP, Hokkaido (Japan), June 2011.

[Yole14a] “Yole Sees $13BN CMOS Image Sensor Market by 2018”. *Optics.Org.,* Jan 20, 2014. http://optics.org/news/5/1/23.

[Yole14b] 2014 Yole Development – Market and Technology Reports. *CMOS Image Sensors*

[Yuzo05] Y. Furukawa et al., “Analog Integrated Circuit for Edge Detection with Massively Parallel Processing Based on Vertebrate Outer Retina”. *Sensors and Materials*, vol.17, no.5, pp. 299-306, January 2005.

[Zara11] Á. Zarándy, *Focal-Plane Sensor-Processor Chips*. Springer. ISBN 978-1-4419-6474-8, 2011.

ABBREVIATIONS AND CONSTANTS

ABC	Analog to Binary Conversion
ACE	Analogic Cellular Engine
ADC	Analog-to-Digital Converter/Conversion
ADE	Aplication Development Environment
APAP	Analog Parallel Array Processors
APS	Active Pixel Sensor
ASIP	Application Specific Information Processing
BAC	Binary to Analog Conversion
BSI	Back Side Illumination
CCD	Charge Coupled Device
CDS	Correlated Double Sampling
CFPP	Code for Focal Plane Processor
CIA	Charge Integration Amplifier
CIS	CMOS Image Sensor
CIF	Common Intermediate Format
Ck_Control	System control clock
CLK	External Master Clock
CMOS	Complementary Metal-Oxide Semiconductor
CNN	Celullar Neural Network
CNNUM	Cellular Neural Network Universal Machine
CG	Conversion Gain
CGU	Clock Generation Unit
CTRL_CLK	Control Clock Generation
CVIS	CMOS Vision Sensors
DAC	Digital to Analog Converter
DDR	Double Data Rate
DIP	Digital Image Processor
DN	Digital Number
DNC	Do Not Care
DNL	Differential Non Linerity
DR	Dynamic Range
DSNU	Dark Signal Non Uniformity
EEPROM	Electrically Eresable Programmable Read Only Memory
EMR	ElectroMagnetic Radiation
Eye-RIS	Eye-RIS from AnaFocus
Eye-RIS_VSoC	Eye-RIS Vision System on Chip
FD	Floating Diffusion
FSR	Full Signal Range
ENG	Global Enable
EN_ROPU	Enable of Reset on Power Up block
FPGA	Field Programmable Gate Array

FPN	Fixed Pattern Noise
FPP	Focal Plane Processor
FpS	Frame Per Second
FSI	Front Side Illumination
GAPU	Global Analogic Program Unit
GigE	Gigabit Ethernet
GBTN	Gloal Binary Test Node
GOpS	Giga Operations per Second
GPIO	General Purpose Inputs/Outputs
HDR	High Dynamic Range
INL	Integral Non Linearity
IRQ	Interruption ReQuest
JTAG	Joint Test Action Group (Standard IEEE 1149.1)
ladn	Local analog data node
lddn	Local digital data node
LAM	Local Analog Memory
LDM	Local Digital Memory
LLU	Local Logic Unit
LVCMOS	Low Voltage CMOS
LVDS	Low-Voltage Differential Signaling
MAC	Multipler-Accumulator Circuit
MLP	Multi-Layer Perceptron
NN	Neural Network
PBCTRL	Programming Block Control
PBMASK	Programming Block Mask
PBMEM	Programming Block Memory
PCU	Programmable Control Unit
PDF	Probability Density Function
PIO	Parallel Input/Output
PoR	Power on Reset
PPS	Passive Pixel Sensor
PSD	Power Spectral Density
PTAT	Proportional To Absolute Temperature
PWM	Pulse-Width Modulation
QC	Q-eye Controller
QCIF	Quarter CIF
Q-Eye	Q-Eye CVIS form AnaFocus
QSXGA	Quad Super Extended Graphics Array
RAM	Random Access Memory
RESET	System reset signal
RISC	Reduced Instruction Set Computer
RNN	Recurrent Neural Network
SC	Switched Capacitor
SCIS	Smart CMOS Image Sensors

S&H	Sample and Hold
SF	Source Follower
SIMD	Singel Instruction Multiple Data
SNR	Signal Noise Ratio
SPAD	Single Photon avalanche Diodes
SPI	Serial Port Interface
SWaP	Size, Weight and Power
ToF	Time of Flight
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VGA	Video Graphics Array
WTA	Winner-Take-All
k	Boltzmann´s constant
T	Absolute temperature
q	electron charge