ESCUELA POLITÉCNICA SUPERIOR

UNIVERSIDAD DE SEVILLA

TRABAJO DE FIN DE GRADO

INGENIERÍA ELECTRÓNICA INDUSTRIAL

CONSTRUCCIÓN DE UN PULSÓMETRO USANDO FPGAS

Autor: Daniel Muela Galán

Tutores: Carlos Jesús Jiménez Fernández

Carmen Baena Oliva

Trabajo de Fin de Grado

Ingeniería Electrónica Industrial

CONSTRUCCIÓN DE UN PULSÓMETRO USANDO FPGAS

Autor:

Daniel Muela Galán

Tutores:

Carlos Jesús Jiménez Fernández

Carmen Baena Oliva

Departamento de Tecnología Electrónica

Escuela Politécnica Superior

Universidad de Sevilla

Sevilla, 2017

Solo una cosa hace que un sueño sea imposible: el miedo a fracasar.

Paulo Coelho, El Alquimista

AGRADECIMIENTOS

Toda la afición e interés por la electrónica empezó de pequeño. La curiosidad de cómo funcionaban los circuitos de los juguetes y aparatos electrónicos que desmontaba han sido los que me han llevado a conocer más sobre esta ciencia.

Después de años de esfuerzos y perseverancia en estudios previos y en el grado de Ingeniería Electrónica Industrial. Quiero agradecer a mi familia que ha sido mi referencia de esfuerzo y sacrificio, compañeros de clase que hoy en día son grandes amigos, profesores que han puesto todas sus ganas y cariño en formarme. Por supuesto a los tutores del trabajo que sin ellos hubiese sido muy complicado abordar el proyecto. Y especialmente quiero agradecer a mi pareja y a su familia por sacarme la mejor sonrisa.

RESUMEN

El presente proyecto tiene como objetivo principal la construcción de un pulsómetro o púlsimetro utilizando el sensor MAX30102 (Maxim Integrated) e implementando el procesado en una FPGA, concretamente en una placa Nexyx4 DDR (Digilent).

Este trabajo comprende desde el estudio de las técnicas que existen para medir el pulso cardíaco y los sensores disponibles en el mercado, hasta el diseño e implementación del sistema completo en una FPGA. El sistema implementado incluye el diseño de la interfaz I2C para la comunicación con el sensor, el diseño del circuito encargado de la configuración y la transmisión de datos con el sensor, el procesado de los datos recibido y mostrar la información del número de pulsaciones por minuto en visualizadores 7-segmentos.

La realización de este trabajo ha supuesto el aprendizaje del manejo del lenguaje VHDL para la descripción y verificación de los diseños, la utilización de la herramienta Xilinx ISE como entorno de desarrollo y la placa Nexys4 DDR con una FPGA Artix 7 como plataforma de implementación.

También se ha seguido una metodología de diseño con verificaciones a distintos niveles para detectar los posibles fallos y garantizar el buen funcionamiento del sistema antes de su programación.

Palabras Clave: Pulsómetro, Diseño Digital, FPGAs, Sensor de pulso cardíaco.

ABSTRACT

The current project has as main goal the construction of a heart rate monitor with the use of the MAX30102 (Maxim Integrated) sensor and implementing the processing in an FPGA, specifically on a Nexyx4 DDR (Digilent).

This work includes from the study of the techniques that exist to measure the heart rate and the sensors available in the market, to the design and implementation of the complete system in an FPGA. The implemented system includes the design of the I2C interface for communication with the sensor, the design of the circuit for the configuration and the reception of data from the sensor, the processing of the received data and the display of the information of the number of beats per minute in 7-segment display.

The realization of this work has involved learning the VHDL language for the description and verification of the designs, the use of the Xilinx ISE environment as the CAD tool and the Nexys4 DDR board with an Artix 7 FPGA as an implementation platform.

It has also been followed a design methodology with verifications at different levels to detect possible failures and ensure the smooth operation of the system before programming.

Keywords: Digital Design, FPGAs, Heart Rate Sensor.

INDICE DE LA MEMORIA

INDICE DE ILUSTRACIONES

1 INTRODUCIÓN

Un proyecto completo que comprendiera la mayoría de los conceptos de la Electrónica Digital y parte de la Analógica puede ser por ejemplo un sistema de medida.

Entre la gran variedad de aplicaciones electrónicas, entre los últimos años han tenido un gran desarrollo los dispositivos personales relacionados con la salud. La ingeniería electrónica puede ayudar a salvar vidas de personas o por otro lado puede infórmale de los parámetros más importantes del cuerpo humano.

En la actualidad, no solo en hospitales sino en gimnasios o incluso en la calle, existen multitudes de dispositivos (Wearables) que están interconectados de manera cableada o inalámbrica con otros dispositivos de uso habitual como es un Smartphone y a través de él, a servidores como son las bases de datos que registra el historial de un paciente o su actividad física.

Uno de los dispositivos más extendidos son los medidores de pulso cardíaco. Tradicionalmente utilizados en hospitales, pero hoy en día su uso se ha expandido en otros muchos ámbitos.

En el ámbito clínico, la medida del pulso cardíaco es muy importante y no admite errores. Por otro lado, y acercándose al deporte, tanto deportistas de alto nivel como aficionados, utilizan con mayor frecuencia el uso de tecnologías de monitorización de parámetros corporales para mejorar objetivos deportivos y/o personales. En estos casos las exigencias de precisión no son tan elevadas.

Con todas estas premisas, el objetivo de este proyecto es construir un prototipo de un pulsómetro o pulsímetro utilizando un sistema digital basado en una FPGA. El sistema debe interaccionar con un sensor de pulso cardíaco, adquirir los datos, procesarlos y mostrar en un visualizador los valores de la frecuencia cardíaca.

Los aspectos en los que se ha trabajado en este proyecto son diferentes, amplios y relacionados entre sí para conseguir el objetivo propuesto.

Ha requerido de un estudio preliminar de las soluciones que hay hoy en día para medir el ritmo cardíaco.

Se ha tenido que estudiar el mercado de sensores de ritmo cardíaco, los tipos que hay, sus características técnicas, el coste, incertidumbre de medida, etc. y seleccionar

el más adecuado de cara al proyecto. También ha requerido el estudio de las características, interfaz y tipo de datos que proporciona el sensor seleccionado.

Desde el punto de vista de diseño, este proyecto requiere el diseño de una interfaz de comunicación con el sensor, un circuito de procesado de la información recibida y un circuito de visualización de los resultados. Todos estos circuitos tienen que ser verificados en las diferentes etapas de su diseño para comprobar su correcto funcionamiento.

Desde el punto de vista del aprendizaje, este proyecto requiere conocimientos de aspectos muy variados. Requiere el estudio del funcionamiento de los sensores de pulso cardíaco, sus interfaces y la forma en la captan la información. Requiere también el estudio de los dispositivos FPGA, que son los que se van a utilizar para la implementación de los diseños. Además, requiere el aprendizaje y manejo del lenguaje de descripción de hardware VHDL, que es el que se va a utilizar para la descripción de los diseños.

Finalmente, también hay que tener en cuenta los aspectos relacionados con la metodología de diseño y las herramientas CAD necesarias para desarrollarlo. En un diseño de mediana complejidad como el que se aborda, el seguimiento de una metodología de diseño adecuada, con verificaciones a distintos niveles y en todos los diseños que se van realizando es fundamental para no acumular errores y conseguir un buen diseño. En este proyecto se ha utilizado la herramienta Xilinx ISE para la elaboración, la verificación e implementación de los diseños. Se han utilizado diferentes mecanismos de verificación (no únicamente simulaciones) para la comprobación de su correcto funcionamiento.

.

La estructura de la memoria es la siguiente:

En el capítulo 2 se presentan las técnicas actuales más utilizas para la medida del pulso cardíaco. También se comentan los sensores más comercializados y la selección del más adecuado, además de las herramientas y sistemas necesarios que se ha utilizado a lo largo del trabajo.

En el capítulo 3 se describe el protocolo de comunicación que usa el sensor, la elaboración del circuito encargado de controlar la comunicación con el sensor y las verificaciones que se han realizado.

En el capítulo 4 se exponen los subsistemas necesarios, el diseño que se ha desarrollado y las verificaciones oportunas para la adquisición de datos. Así como el análisis de los resultados obtenidos.

En el capítulo 5 se explica el diseño realizado para procesar los datos y presentarlos en el visualizador. También se presentan las comprobaciones y resultados conseguidos.

Finalmente, en el capítulo 6 condensan las conclusiones más importantes del proyecto.

2 PULSO CARDÍACO Y PROCESADO DE INFORMACIÓN

La medida del ritmo cardíaco no solo tiene como objeto evaluar la eficiencia física sino también ayuda a encontrar posibles problemas en la salud. A medida que se envejece pueden existir cambios en la frecuencia cardíaca que indiquen una afección y requiera atención médica.

Las dos técnicas no invasivas más utilizadas hoy en día para determinar el ritmo cardíaco son la electrocardiografía y la pulsioximetría. Ambas proporcionan un valor en la unidad de medida de BPM, es decir, batidos o latidos por minutos.

La electrocardiografía es un método que capta y amplía el potencial eléctrico del corazón a través de electrodos situados en el tórax y extremidades. Los datos eléctricos medidos son representados en una gráfica llamada electrocardiograma. El aparato electrónico basado en la electrocardiografía es el electrocardiógrafo. Suelen trabajar en un rango de 15 a 300 BPM con una precisión de ± 2 BPM o ± 1%, en el peor de los casos. Los electrocardiógrafos suelen ser caros y en la mayoría de los dispositivos es posible visualizar el electrocardiograma en tiempo real en una pantalla LCD (ver figura 1).

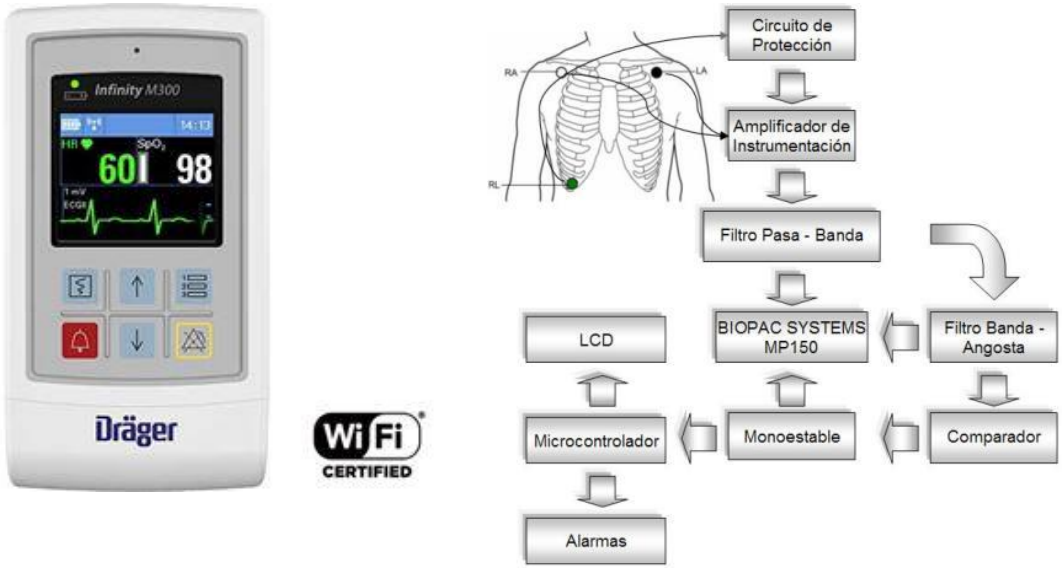


Ilustración 1 De izquierda a derecha, electrocardiógrafo y esquema de subsistemas de un electrocardiógrafo

Por otro lado, la pulsioximetría es una técnica de medición no invasiva del oxígeno transportado por la hemoglobina en el interior de los vasos sanguíneos y del ritmo cardíaco. Para comprender el funcionamiento, es necesario tener conocimiento acerca del modelo físico de dualidad onda-partícula y de la ley de Beer-Lambert que relaciona

la absorción de luz (luz roja y/o infrarroja) con las propiedades del material atravesado (la piel). Entonces, basándose en el comportamiento frente a las diferentes longitudes de onda, se puede detectar el pulso cardíaco y también la presencia o ausencia de oxígeno.

La medición se apoya en que el flujo de sangre arterial es pulsátil y el resto de los fluidos y tejidos no. La pulsación de sangre arterial modula la luz que lo atraviesa, mientras que los otros fluidos y tejidos mantienen una absorción constante. Se asume que la componente arterial pulsátil (CA) hace variar la luz absorbida por el aumento de longitud del camino óptico en el lugar de medición. La suma de la componente variante y la constante representa una gráfica que se denomina fotoplestimografía (ver figura 2).

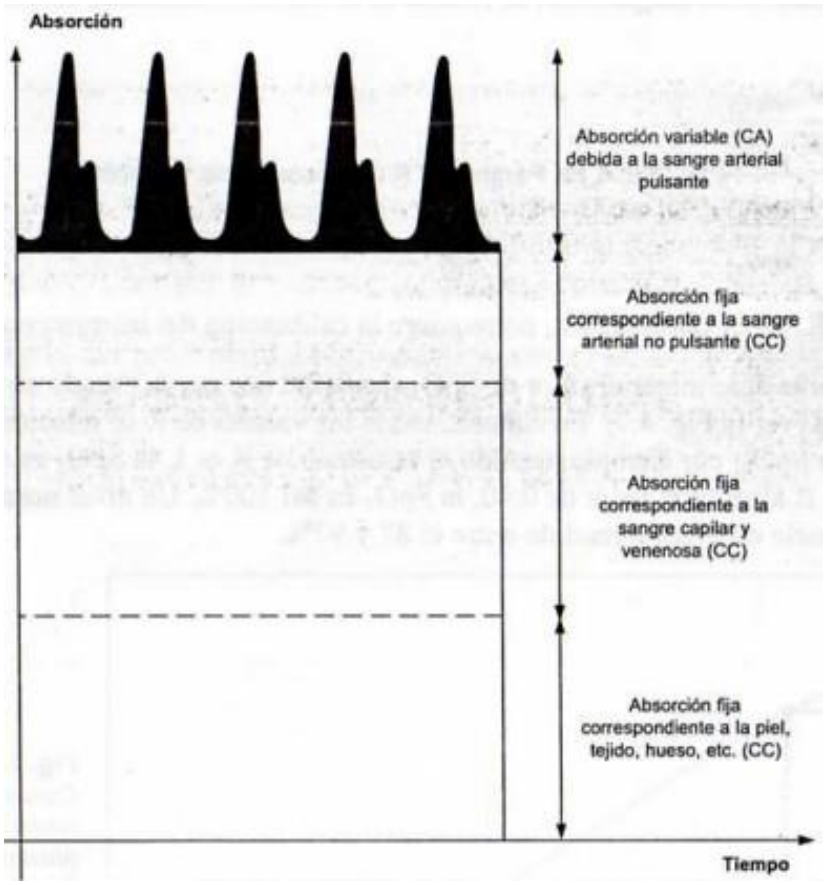


Ilustración 2 Absorción de la epidermis (piel) frente a la luz que lo atraviesa.

La medida del ritmo cardíaco utilizando la pulsioximetría tiene asociado una serie de errores que dificultan la medida según las condiciones en las que se esté mensurando. Las siguientes condiciones pueden dar un resultado erróneo de la frecuencia cardíaca:

- Baja perfusión. Bajo gasto cardíaco o sitio de medición con vasoconstricción (frío).

- Sitio elegido para la colocación de la sonda. Es recomendable los lóbulos de la oreja y los dedos índices de las manos. Siendo posible la palma de la mano en niños pequeños.
- Interferencia EMI.
- Movimiento de la sangre venosa.
- Ruido por la luz que emiten los LED (diodo emisor de luz) y llega al detector sin pasar por el lugar de medición.
- Ruido debido al movimiento del cable.
- Fuentes de luces de alta intensidad, como lámpara cialítica.
- Existencia de hemoglobinas disfuncionales.
- Desconexión del sensor.

Hay dos tipos de pulsioximetria, por transmisión o reflexión de la luz (ver figura 3).

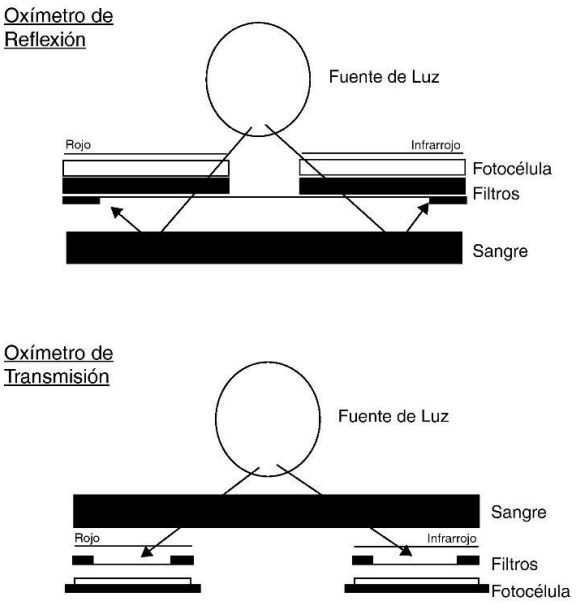


Ilustración 3 Pulsioximetría de reflexión y transmisión respectivamente

El dispositivo que se encarga de medir la saturación de oxígeno en sangre y el ritmo cardíaco basado en la pulxiosimetría es el pulsímetro o pulsómetro. Cada vez más, es posible encontrar pulsómetros en dispositivos deportivos como las Smartband o Smartwatch y por supuesto en centros médicos. La mayoría de pulsímetros poseen una pantalla donde poder visualizar la frecuencia cardíaca. En general, el rango de medida abarca desde los 30 BPM a 250 BPM con precisiones de +/- 2 BPM y además tienen un consumo eléctrico y coste bajo (ver figura 4).

Ilustración 4 De izquierda a derecha, oxímetro de pulso y smartband.

Incluso sin disponer de estos aparatos, es posible determinar aproximadamente el ritmo cardíaco si dispone de un Smartphone con Android, gracias a la aplicación Icare Monitor de iCare Health Studio.

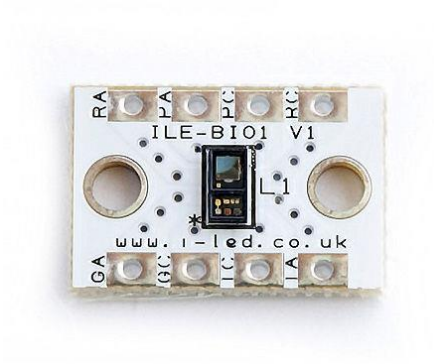
Por lo tanto, se ha seleccionado la pulxiosimetría y no la electrocardiografía como la técnica a seguir debido a que supone bajo coste, precisiones aceptables y amplios rangos.

2.1 SENSORES DE PULSO CARDÍACO

Adentrándose en la tecnología utilizada por los pulsímetros, aparecen una variedad de sensores donde es difícil una elección dado que sus características técnicas son bastante similares. A continuación, se muestran los modelos de sensores ópticos para la medida de frecuencia cardíaca más comercializados.

Sensor Biofy Eco1 SFH 7050 de ILS

Sensor que utiliza led rojo, led verde y led Infrarrojo para la emisión de luz y un fotodiodo para recibir la reflexión de luz de la piel. Este sensor no contiene subsistemas de comunicación para la adquisición de datos como son los ADCs, filtros, registros de datos, etc. Por ello no se ha seleccionado para este proyecto (ver figura 5).



Pin configuration

Pin	Name	Function
1	GC	Green LED Cathode
2	GA	Green LED Anode
3	RA	Red LED Anode
4	PA	Photodiode Anode
5	PC	Photodiode Cathode
6	RC	Red LED Cathode
7	IA	Infrared LED Anode
8	IC	Infrared LED Cathode

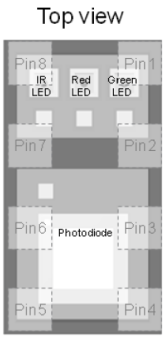


Ilustración 5 Sensor Biofy Eco1 SFH 7050

Sensor SFH 7060

Este sensor incluye más leds que el anterior pero tampoco posee ADC, ni registros, ni interfaz de comunicación (ver figura 6).

Pin configuration

Pin	Name	Function
1	RA	Red LED Anode
2	RC	Red LED Cathode
3	G1C	Green LED 1 Cathode
4	G1A	Green LED 1 Anode
5	G2A	Green LED 2 Anode
6	PA	Photodiode Anode
7	PC	Photodiode Cathode
8	G2C	Green LED 2 Cathode
9	G3A	Green LED 3 Anode
10	G3C	Green LED 3 Cathode
11	IC	Infrared LED Cathode
12	IA	Infrared LED Anode

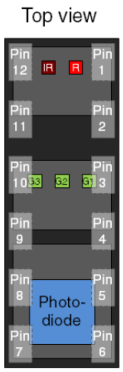


Ilustración 6 Esquema de pines del sensor SFH 7060.

Sensor NJL5501R y Módulo AFE4403

Este sensor presenta características análogas a los anteriores y tampoco tiene interfaz de comunicación. Para mejorar este inconveniente, se puede hacer uso del módulo AFE4403 que da soporte al muestreo y captura de datos e interfaz de comunicación SPI del sensor con una plataforma como puede ser un microcontrolador o una FPGA. Sin embargo, es preferible unificar todas las tensiones de los subsistemas y el sensor en tan solo una tensión de entrada ya que los leds, el fotodiodo y el módulo AFE4403 trabajan a diferentes tensiones y por ello no se ha elegido (ver figura 7).

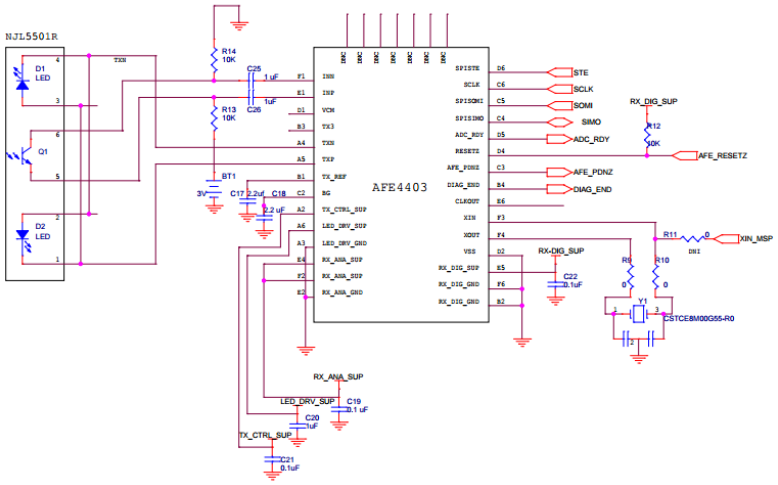


Ilustración 7 Esquema del sensor NJL5501R y módulo de adquisición AFE4403

Sensor Si114X

Los sensores Si1141/42/43 de Silicon Laboratories son tres modelos cuyas características son idénticas. Solo contiene leds infrarrojos para la emisión de luz, aunque el receptor también puede recibir espectro visible. Utiliza la técnica de reflexión y tiene comunicación serie I2C. A pesar de que sus aplicaciones finales son muy variadas, queda incompleto si se quisiera hacer medidas emitiendo luz visible (ver figura 8).

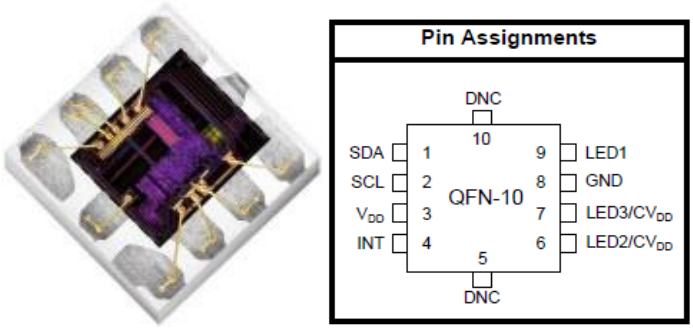


Ilustración 8 Sensor y esquemas de pines del Si1141X

Sensor MAX30102

El sensor MAX30102 (ver figura 9) del fabricante Maxim integrated se centra principalmente en la medida del ritmo cardíaco y la saturación del oxígeno en sangre utilizando la técnica de reflexión de luz. Este contiene dos leds (rojo e infrarrojo), un fotodiodo de amplio rango de longitud de onda para la recepción de la luz visible e infrarroja, ADC de 18 bits de resolución para la conversión de datos analógicos a digital, filtro digital, registros de datos e interfaz I2C de comunicación además de un bajo consumo que hacen de este sensor el más sofisticado y preciso del mercado. Para completarlo y dar solucionar a las diferentes tensiones con las que trabajan los leds y los subsistemas descritos existen dos encapsulados, el convertidor de nivel lógico MAX14595 y el convertidor DC-DC MAX1921 integrados ambos en la misma PCB del sensor llamada MAXREFDES117 (ver apartado 2.3).

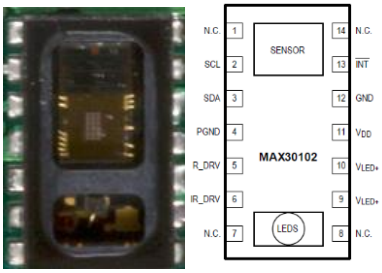


Ilustración 9 Sensor y esquema de pines MAX30102

Por consiguiente, si se observa con profundidad los datasheets de cada sensor para medir el ritmo cardíaco, en general son relativamente muy similares. Con el objetivo de reducir la elección de los sensores, se ha considerado el sensor que posea todos los subsistemas necesarios para la comunicación y la adquisición de las señales provenientes del fotodetector, así como la gestión de las señales de control para los leds, ADC, registros, etc. De esta forma se descartan los sensores SFH y NJL5501R. Para reducir aún más dicha selección, se ha prescindido del sensor Si1141/42/43 por no disponer de diferentes leds y por su baja resolución del ADC en comparación con el MAX30102. Por lo tanto, el sensor MAX30102 se elige como sensor que más se adapta a este proyecto.

2.2 SENSOR MAX30102

El sensor MAX30102 posee una interfaz de comunicación serie I2C. Para ello, dispone de tres señales INT, SDA y SCL que será el camino para acceder internamente a sus subsistemas (ver figura 10).

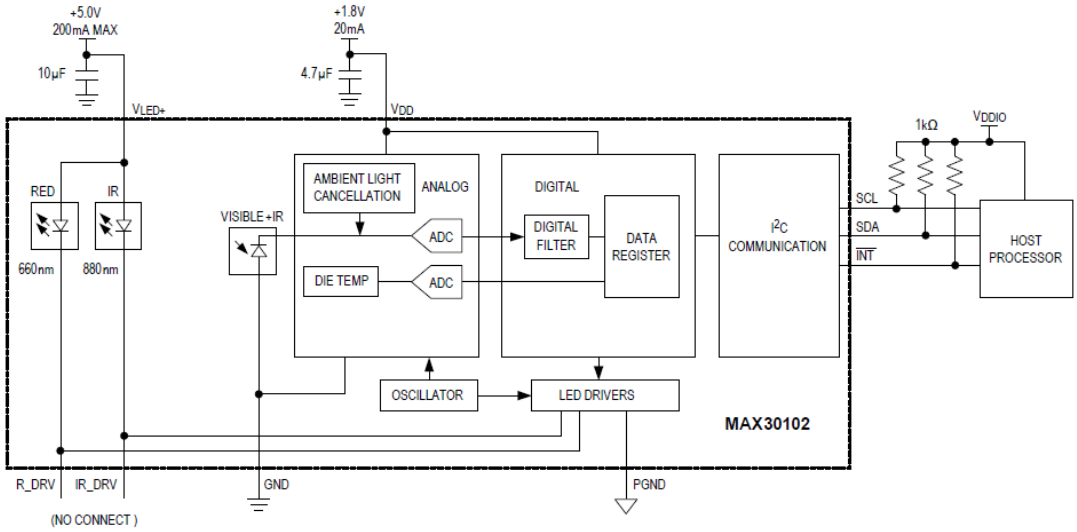


Ilustración 10 Esquema de subsistemas del sensor MAX30102

Un aspecto interesante de este sensor es que contiene un mapa de 20 registros los cuales se pueden configurar a través de la comunicación serie I2C dando flexibilidad al uso de este. De esta manera se accede al ajuste de interrupciones, registro de datos de la FIFO, modos de funcionamientos, corriente y ancho de pulso de los leds, frecuencia de muestreo, resolución, etc. (ver figura 11).

Register Maps and Descriptions

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
STATUS											
Interrupt Status 1	A_FULL	PPG_ RDY	ALC_ OVF	PROX_ INT				PWR_ RDY	0x00	0X00	R
Interrupt Status 2							DIE_TEMP _RDY		0x01	0x00	R
Interrupt Enable 1	A_FULL_ EN	PPG_ RDY_EN	ALC_ OVF_EN	PROX_ INT_EN					0x02	0X00	R/W
Interrupt Enable 2							DIE_TEMP _RDY_EN		0x03	0x00	R/W
FIFO											
FIFO Write Pointer				FIFO_WR_PTR[4:0]					0x04	0x00	R/W
Overflow Counter				OVF_COUNTER[4:0]					0x05	0x00	R/W
FIFO Read Pointer				FIFO_RD_PTR[4:0]					0x06	0x00	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x07	0x00	R/W
CONFIGURATION											
FIFO Configuration	SMP_AVE[2:0]			FIFO_ ROLL OVER_EN	FIFO_A_FULL[3:0]				0x08	0x00	R/W
Mode Configuration	SHDN	RESET				MODE[2:0]			0x09	0x00	R/W
SpO_2 Configuration	0 (Reserved)	SPO2_ADC_RGE [1:0]		SPO2_SR[2:0]			LED_PW[1:0]		0x0A	0x00	R/W
RESERVED									0x0B	0x00	R/W
LED Pulse Amplitude	LED1_PA[7:0]								0x0C	0x00	R/W
	LED2_PA[7:0]								0x0D	0x00	R/W
RESERVED									0x0E	0x00	R/W
RESERVED									0x0F	0x00	R/W
Proximity Mode LED Pulse Amplitude	PILOT_PA[7:0]								0x10	0x00	R/W
Multi-LED Mode Control Registers		SLOT2[2:0]				SLOT1[2:0]			0x11	0x00	R/W
		SLOT4[2:0]				SLOT3[2:0]			0x12	0x00	R/W

Register Maps and Descriptions (continued)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
RESERVED									0x13–0x17	0xFF	R/W
RESERVED									0x18-0x1E	0x00	R
DIE TEMPERATURE											
Die Temp Integer	TINT[7:0]								0x1F	0x00	R
Die Temp Fraction					TFRAC[3:0]				0x20	0x00	R
Die Temperature Config								TEMP _EN	0x21	0x00	R
RESERVED									0x22–0x2F	0x00	R/W
PROXIMITY FUNCTION											
Proximity Interrupt Threshold	PROX_INT_THRESH[7:0]								0x30	0x00	R/W
PART ID											
Revision ID	REV_ID[7:0]								0xFE	0xXX*	R
Part ID	PART_ID[7]								0xFF	0x15	R

**XX denotes a 2-digit hexadecimal number (00 to FF) for part revision identification. Contact Maxim Integrated for the revision ID number assigned for your product.*

Ilustración 11 Banco de registros del sensor MAX30102

2.3 PLACA MAXREFDES117

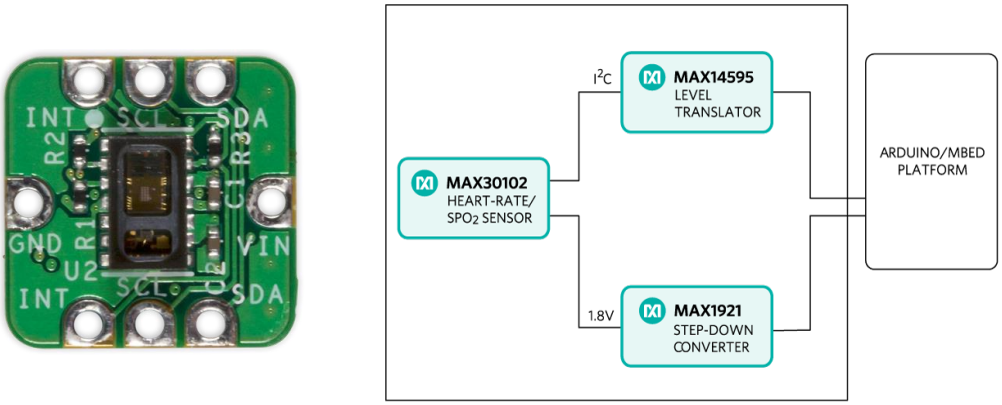


Ilustración 12 Placa MAXREFDES117

La placa MAXREFDES117 está compuesta por el sensor MAX30102 y los módulos MAX14595 y MAX1921 con sus respectivos componentes en SMD según Maxim integrated (ver figura 12).

El módulo MAX1921 es un convertidor de potencia DC-DC que se encarga de reducir la tensión de entrada Vin de la FPGA de 3,3 V a una tensión de alimentación del sensor de 1,8 V.

El módulo MAX14595 es un convertidor de nivel lógico para adaptar las señales del bus I2C del sensor a los pines de la FPGA para que pueda interpretar correctamente el 1 y 0 lógico.

También, el fabricante del sensor proporciona recursos como son los códigos de programación, lista de componentes, esquemáticos, etc. De esta manera, se puede comprobar el funcionamiento o tomar como referencia respecto al protocolo de comunicación, adquisición y procesado. Las dos plataformas que ofrece son Arduino y mbed.

Por otra parte, el código I2C master en VHDL de la referencia [4] da soporte para el diseño del controlador de la interfaz de comunicación I2C.

Por lo tanto, debido a que las prestaciones que ofrece MAXREFDES117 son excelentes se eligió para el desarrollo del proyecto y además la plataforma Arduino sirvió como guía en aspectos de configuración y procesado.

2.4 FPGA

FPGAs (Field Programmable Gate Arrays) son dispositivos programables de silicio basando en bloques lógicos configurables (memorias y puertas lógicas programables) interconectado con los bloques de entrada y salida (IOBs) capaces de realizar las funciones deseadas por el usuario. A diferencia de un microprocesador que tiene una estructura fija y ejecuta distintas aplicaciones, con FPGAs se configura el hardware interno, en definitiva el circuito, dando como resultado distintas funciones. Por lo cual esto supone: tiempos más rápidos de respuesta de E/S y funcionalidad especializada, supera la potencia de cómputo de procesadores de señales digitales, rápida generación de prototipos y verificación sin el proceso de fabricación del diseño personalizado de ASIC, implementa funcionalidad personalizada con la fiabilidad de hardware determinístico dedicado.

Observando la figura 13, se puede ver internamente el esquema de una FPGA.

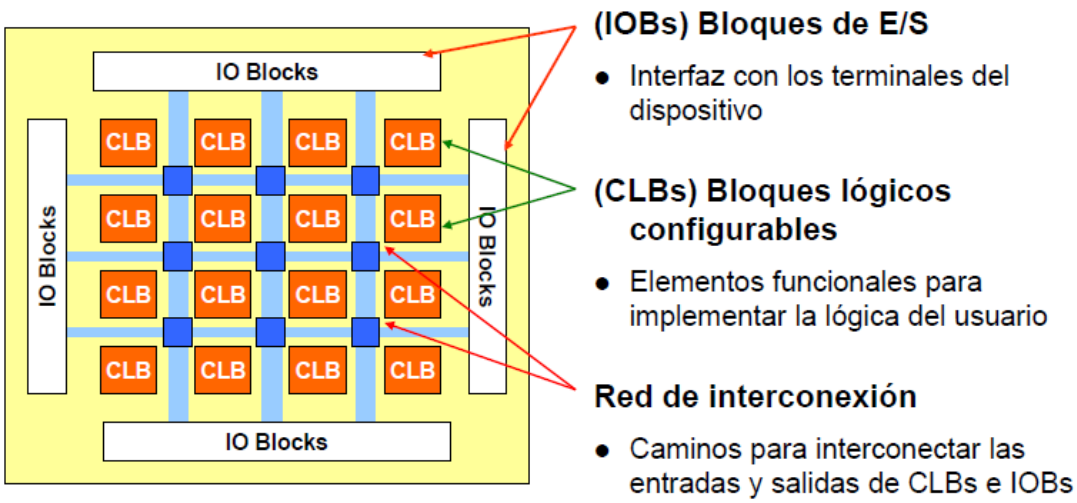


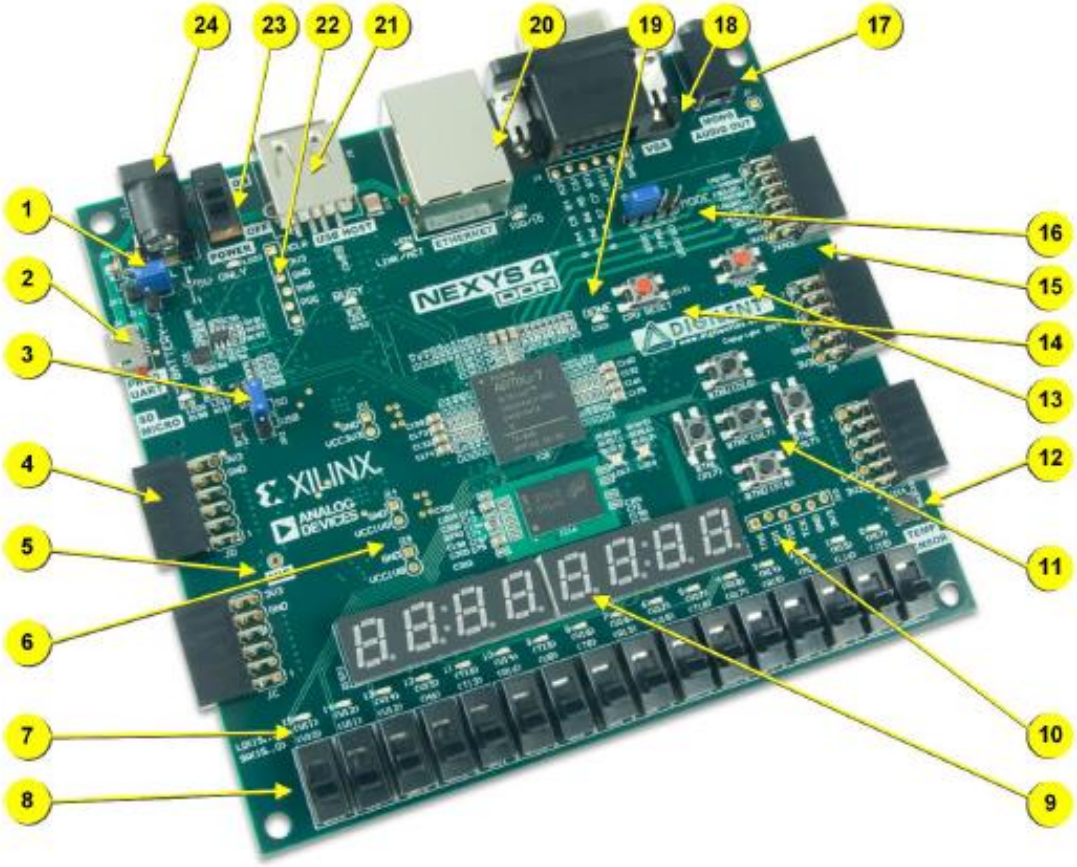
Ilustración 13 Esquema interno de una FPGA

2.4.1 NEXYS 4 DDR

La placa Nexys 4 DDR (ver figura 14) es una plataforma preparada para desarrollar circuitos digitales basada en la última FPGA Artix-7™. Esta placa ofrece las siguientes características internas: 15,850 biestables (logic slices), 4,860 Kbits de bloques de RAM rápida, 6 control de relojes con cada uno con una (PLL), 240 biestables de DSP, reloj interno de 450 MHz y convertidor analógico digital integrado (XADC).

Como puertos y periféricos: 16 switches, 16 leds, dos bloques de 4 dígitos de 7 segmentos, puentes de USB-UART, 2 leds tricolor, conector Micro SD card, salida de 12-bit VGA, salida de audio PWM, micrófono PDM, acelerómetro de 3 ejes, sensor de temperatura chip Ethernet PHY 10/100, 128MiB DDR2, serial Flash, 4 puertos Pmod, Pmod para señales XADC, puerto de programación y comunicación USB-JTAG Digilent, USB HID Host para ratón, teclado y memory sticks.

Es por ello por lo que se ha utilizado este dispositivo dado que es necesario hacer muchas pruebas de funcionamiento y contiene los elementos necesarios. De los más utilizados han sido los puertos de entrada y salida, la comunicación USB-UART, los leds, los switches, los 7 segmentos, el sensor de temperatura y por supuesto, el puerto de programación USB-JTAG.



Callout	Component Description	Callout	Component Description
1	Power select jumper and battery header	13	FPGA configuration reset button
2	Shared UART/ JTAG USB port	14	CPU reset button (for soft cores)
3	External configuration jumper (SD / USB)	15	Analog signal Pmod port (XADC)
4	Pmod port(s)	16	Programming mode jumper
5	Microphone	17	Audio connector
6	Power supply test point(s)	18	VGA connector
7	LEDs (16)	19	FPGA programming done LED
8	Slide switches	20	Ethernet connector
9	Eight digit 7-seg display	21	USB host connector
10	JTAG port for (optional) external cable	22	PIC24 programming port (factory use)
11	Five pushbuttons	23	Power switch
12	Temperature sensor	24	Power jack

Ilustración 14 Elementos de la placa Nexys 4 DDR

2.4.2 ISE DESIGN SUITE 14.7 DE XILINX

Para elaborar un diseño digital basado en FPGAs es necesario un software el cual se apoya en los lenguajes VHDL o Verilog de alto nivel con el propósito de realizar las siguientes tareas: descripción del diseño, verificación de este con las entidades de simulación TestBench (ver apartado 3.3), optimización de los recursos a utilizar con el circuito sintetizado y place & route que supone definir las regiones dónde colocar los componentes, interconectar los bloques lógicos configurables y por último, generar el fichero bitstream para enviar a la FPGA y programarse.

En este proyecto se ha elegido el software ISE Design Suite versión 14.7.

De este software hay que destacar que tiene un analizador lógico interno llamado “Chip Scope” que permite visualizar el comportamiento de señales internas

Por otro lado, para volcar el diseño sobre la FPGA es necesario un software llamado Adept (Digilent), un cable USB y el puerto USB-JTAG de la FPGA.

2.4.3 LENGUAJE DE DESCRIPCIÓN DE HARDWARE HDL

Debido a la complejidad de los circuitos electrónicos digitales es necesario descripciones de alto nivel de la lógica digital que no estuviesen atadas a una determinada tecnología electrónica, tales como la CMOS o la BJT. Así, los HDL fueron creados para hacer posible el diseño de circuitos con un alto nivel de abstracción, y con la posibilidad de incluir en los modelos características propias de los circuitos electrónicos, tales como los flujos de datos y su variación en el tiempo.

El lenguaje VHDL (Very high speed integrated circuit HDL) es un estándar definido por el IEEE desde 1987 hasta su modificación extensa en 2008. Los lenguajes de descripción de hardware describen un sistema desde varios puntos de vista. Según el comportamiento (qué hace), la estructura (de qué está compuesto), las propiedades funcionales (cómo se conecta a otros módulos), las propiedades físicas (retrasos, velocidad de operación, etc). Las descripciones pueden ser a diferentes niveles: transistor, nivel lógico, transferencia de registros (RT) y a nivel de arquitectura e instrucciones. La metodología a seguir es, especificar el diseño para que realice las funciones previstas y simularlo para verificar el correcto funcionamiento con y sin retraso de señales.

Aspectos fundamentales para el diseño de un circuito digital:

Señales y puertos

Las señales transportan información fuera de la entidad a través de los puertos (in, out, in-out, buffer, etc).

Entidad-arquitectura

Las entidades describen las entradas y salidas de los circuitos y la arquitectura su comportamiento. La arquitectura comprende una parte declarativa de señales, constantes, componentes, etc y una parte de ejecución que puede ser concurrente o secuencial (procesos, funciones, circuitos combinacionales, etc).

Paquetes y librería

Los paquetes contienen declaraciones de tipos de señales y/o componentes, funciones, etc. A su vez, constituyen una librería siendo la más usada la Librería IEEE la cual abarca los paquetes y componentes necesarios para operaciones aritméticas, tipos números, funciones, etc. Los componentes suelen ser entidades las cuales son declaradas y usadas desde otra entidad.

3 COMUNICACIÓN CON EL SENSOR MAX30102

En este capítulo se describirá el diseño que se ha elaborado para controlar la comunicación I2C del sensor MAX30102 con la FPGA.

3.1 INTERFAZ I2C

El bus I2C o I^2C (Inter-Integrated Circuit) es un protocolo de comunicación estándar entre circuitos integrados e implementado en 1000 ICs diferentes y fabricado por más de 50 empresas. Debido a su versatilidad, es utilizado en arquitecturas como System Management Bus (SMbus), Power Management Bus (PMbus), Intelligent Platform Management Interface (IPMI), Display Data Channel (DDC) y Advanced Telecom Computing Architecture (ATCA). Este bus fue desarrollado por Philips Semiconductors (hoy NXP Semiconductors parte de Qualcomm) al inicio de la década de 1980. Actualmente los principales fabricantes de dispositivos semiconductores ofrecen circuitos que implementan un bus I2C para su control. Por otro lado, el Comité I2C aparte de otras cuestiones, le asigna las direcciones I2C a cada tipo de circuito (ver apartado 3.1.1). De esta manera, cada función implementada, independientemente del fabricante, posee la misma dirección, es decir, circuitos que realicen funciones equivalentes deberán poseer la misma dirección oficial I2C independientemente del fabricante.

3.1.1 LAS CARACTERÍSTICAS PRINCIPALES

El bus I2C utiliza únicamente dos líneas series que son la línea serie de datos (SDA) y la línea serie de reloj (SCL). Cada dispositivo tiene una dirección específica de manera que se conecta al bus siendo maestro o esclavo. El maestro puede funcionar como maestro transmisor o maestro receptor. Se permite la existencia de multi-maestro con detención de colisiones y arbitración para mejorar la transmisión. En este proyecto, solamente se tendrá un maestro por lo que no será necesaria la arbitración en el bus.

Respecto a las características técnicas del bus es de tipo serie síncrono de 8 bits, SDA es bidireccional y la velocidad de transmisión puede tomar valores de 100 Kbit/s, 400Kbit/s, 1Mbit/s pudiendo llegar hasta 5Mbit/s. Siendo de 400Kbit/s la máxima a considerar en este proyecto. Normalmente, el bus contiene filtros internos (condensadores) para su mejor funcionamiento. Por último, el número de circuitos integrados (ICs) a conectar en el bus está limitado a 128 dispositivos de los cuales 16 direcciones están reservadas, es decir, 112 dispositivos y puede ampliarse.

3.1.2 FUNCIONAMIENTO

El maestro genera una señal de reloj SCL e inicia la transferencia de datos en el bus a través de la señal SDA. Cada secuencia de transmisión es iniciada por una condición de START (S) o REPEATED START (Sr) y finalizada con una condición de STOP (P). La información es trasmitida bit a bit del esclavo (sensor) al maestro (FPGA) o viceversa con un tamaño de 8 bits por transmisión y seguida por un reconocimiento (ACK). Todo esto, en sincronía con los pulsos SCL generados por dicho maestro.

COMANDOS DE CONTROL DEL BUS

Los comandos de control serán el método de gobernar el bus por parte del maestro y se realizará a partir de dos señales SDA y SCL.

Estas dos señales se mantienen en un valor de alta impedancia (Z) cuando el bus no está siendo usado.

El maestro inicia la comunicación a través de la condición START (una condición START ocurre cuando SDA sufre un flanco de bajada estando SCL en nivel alto).

El maestro termina la comunicación liberando al bus a través de la condición STOP (una condición STOP ocurre cuando SDA sufre un flanco subida estando SCL en nivel alto).

El bus permanece ocupado si se produce una condición REPEATED START en vez de una condición STOP. No reconoce una condición de STOP si se realiza en el mismo pulso de SCL que la condición START (ver figura 15).

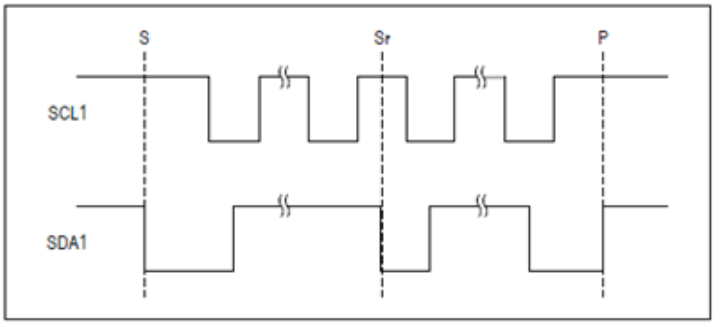


Ilustración 15 Flujo de señales de datos y reloj del bus I2C

El maestro siempre inicia una comunicación con un dispositivo esclavo a través de una condición START seguida del ID del esclavo, es decir, 7 bits de dirección del esclavo y el bit indicador de escritura en el esclavo o de lectura en el esclavo (R/W = 0 escritura, R/W = 1 lectura). El controlador de la interfaz compara cada ID del esclavo bit a bit permitiendo desconectarlo del bus si fuera erróneo. Después de reconocer la condición START y el correcto ID, el esclavo envía o recibe datos.

Al iniciarse una comunicación, únicamente puede ser enviado o recibido un bit a través de SDA cuando la señal SCL están en nivel bajo o dicho de otro modo, puede cambiar SDA de valor cuando SCL están en nivel bajo. Hay una excepción a esto y es cuando se trata de un comando de control del bus donde SDA cambia de valor en un nivel alto de SCL.

RECONOCIMIENTO (Acknowledge)

Un reconocimiento (ACK) se produce cuando el esclavo o el maestro reciben una palabra (8 bits) sin errores de comunicación. Es decir, cuando una transmisión de datos ocurre se invierte 1 ciclo de reloj por bit trasmitido. Al llegar al noveno pulso, el receptor (esclavo o maestro) fuerza a nivel bajo la señal SDA si ha recibido correctamente los datos (ver figura 16).

En el caso opuesto donde no se reciben correctamente los datos, se produciría un error de comunicación y por tanto un no reconocimiento (NACK). Es decir, llegado al último bit de la palabra a transmitir, el receptor fuerza la señal SDA a nivel alto. Los errores de comunicación pueden deberse a la velocidad del esclavo, ruido en el bus, desconexión, etc. Si los problemas persisten, el maestro puede retirar la comunicación al esclavo.

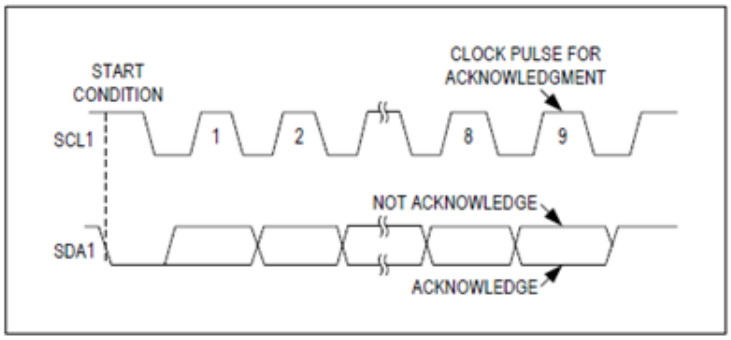


Ilustración 16 Reconocimiento de tramas

Las comunicaciones donde se esté leyendo del esclavo tiene como excepción el NACK en el último bit leído del último dato antes de la condición de STOP.

Los circuitos que estén previstos de interrupciones además de las dos señales descritas, incorporan una señal de interrupción (INT) que permite la gestión de esta por parte del maestro.

CONFIGURACIÓN DEL SENSOR

Llegado a este punto, se va exponer los aspectos fundamentales para comunicarse con el sensor siendo relevantes una configuración previa. Para la configuración se tendrá que escribir y leer en uno o varios registros.

Para escribir en uno o varios registros, es imprescindible realizar los siguientes pasos:

El maestro envía la condición de comienzo (START) acto seguido, envía la dirección del esclavo con el que quiere comunicarse y el modo, que en este caso es escritura (7 bits de dirección y R/W = 0). Tras esto, el receptor envía el ACK (reconocido). A continuación, el maestro envía la dirección del registro de direcciones del esclavo al que quiere escribir (8 bits). Entonces, el receptor vuelve a enviar el ACK. A partir de este momento, el maestro envía un dato (8 bits) o varios seguidos de su correspondiente reconocimiento. Para terminar la escritura el maestro envía la condición de parada (STOP).

Los registros más importantes a configurar y por tanto hay que escribir en ellos son: Interrupt enable 1 y 2, todos los registros de la FIFO, FIFO configuration, mode configuration, SpO Configuration, LED pulse amplitude 1 y 2. El registro de direcciones de registro se incrementa automáticamente y ordenadamente después de cada byte escrito de tal forma que se puede escribir completamente el banco de registros modificables de una vez.

Leer de uno o varios registros implica dos operaciones, una primera de escritura y posteriormente de lectura. La parte de escritura es idéntica a lo ya descrito anteriormente. A partir de que la dirección del registro de direcciones es enviada y recibido el ACK, el maestro envía la condición de REPEATED START (cambiará el sentido de la transacción) y seguidamente vuelve a enviar la dirección del esclavo, pero con el modo de lectura (7 bits de dirección + R/W = 1). Entonces, en el noveno pulso, el receptor lo reconoce y llegado a este momento el esclavo envía un dato o varios al maestro (ACK del maestro al esclavo por dato). Finalizando con un NACK del maestro al esclavo y condición de parada (STOP).

Respecto al puntero de lectura, este se incrementa automáticamente de modo que el sensor envía al maestro datos de los registros en orden secuencial hasta la condición de STOP excepto con el registro FIFO_DATA, donde el registro de direcciones no incrementa automáticamente cuando se leen más de un byte, aunque si lo hace, el registro de datos de la FIFO (FIFO read register) de tal manera que se pueda leer la FIFO completa. Es por ello, que si se quisiera leer el registro de banco completo es necesario, llegado a la dirección FIFO data register, modificar la localización del registro de direcciones con una operación de escritura.

Los registros a leer relevantes para la comunicación son Interrupt Status 1 y 2 y todos los registros de la FIFO. La interrupción se limpia cuando son leídos los registros de estado 1 y 2 o el registro de datos de la FIFO.

La FIFO tiene un tamaño máximo de 192 bytes en la cual se irán depositando cada muestra alternativamente si se trabaja con ambos canales (ver figura 17). Cada muestra tomada es de 3 bytes por canal y hay dos canales que corresponde a los datos de la luz reflejada roja e infrarroja. Se almacena en el orden del byte más significativo al menos significativo (ver figura 18). Para acceder a esta arquitectura de memoria se hace mediante lecturas del citado registro de dato de la FIFO.

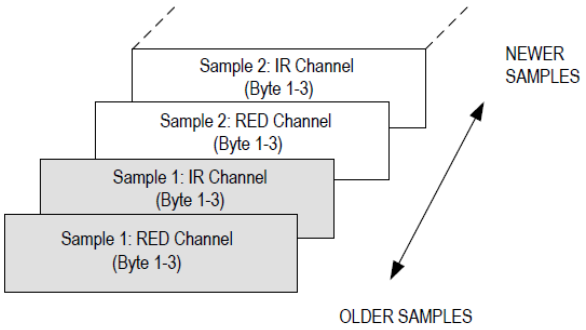


Ilustración 17 Orden de datos en la FIFO

BYTE 1							FIFO_ DATA[17]	FIFO_ DATA[16]
BYTE 2	FIFO_ DATA[15]	FIFO_ DATA[14]	FIFO_ DATA[13]	FIFO_ DATA[12]	FIFO_ DATA[11]	FIFO_ DATA[10]	FIFO_ DATA[9]	FIFO_ DATA[8]
BYTE 3	FIFO_ DATA[7]	FIFO_ DATA[6]	FIFO_ DATA[5]	FIFO_ DATA[4]	FIFO_ DATA[3]	FIFO_ DATA[2]	FIFO_ DATA[1]	FIFO_ DATA[0]

Ilustración 18 Estructura de datos de un canal

3.2 ENTIDAD DE COMUNICACIÓN I2C

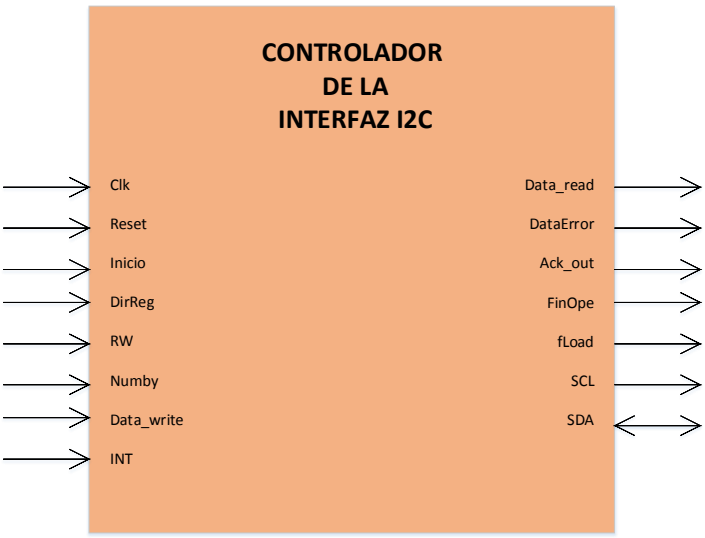


Ilustración 19 Bloque controlador de la interfaz I2C

Esta entidad llamada I2C Master se ha diseñado para el control del citado bus I2C y está basada en máquinas de estados finitos. Para poder emprender este diseño me he referenciado en el código ewiki. El código ewiki se encarga de controlar las transmisiones de un bus genérico I2C diseñado por Scott Larson.

ENTRADAS	**NOMBRE DE SEÑAL**	**TIPO**	**DESCRIPCIÓN**
	Clk	Std_logic	Reloj de 50 MHz encargada de gestionar la máquina de estado
	Reset	Std_logic	Reset asíncrono para establecer unas condiciones iniciales conocidas. Activo en nivel bajo.
	Inicio	Std_logic	Da comienzo a la comunicación. Activo en nivel alto.
	DirgReg	Std_logic_vector (8bits)	Contiene la dirección del registro de direcciones del sensor.
	RW	Std_logic	Bit de escritura o lectura: RW = 0 escritura. RW = 1 lectura.
	Numby	Std_logic_vector (9bits)	Número de veces a escribir o leer de los registros interno del sensor (máximo: 288 bytes).
	Data_write	Std_logic_vector (8bits)	Contiene el byte de dato a escribir
	INT	Std_logic	Señal de interrupción del sensor. Activa en baja.

SALIDAS	SCL	Std_logic	Reloj de 400 KHz del bus I2C.
	Data_read	Std_logic_vector (8bits)	Contiene el valor de una lectura de un registro interno del sensor.
	DataError	Std_logic	Señal de error. Activo en alta
	Ack_out	Std_logic	Señal de reconocimiento del esclavo o maestro. Activo en alto.
	FinOpe	Std_logic	Señal de fin de comunicación. Activo en alto.
	fLoad	Std_logic	Señal de finalización de datos cargados para iniciar la comunicación. Activo en alto.
ENTRA/SALIDA	SDA	Std_logic	Señal de dato del bus I2C (bidireccional).

DATOS DE LA COMUNICACIÓN

- 400 KHz frecuencia máxima del bus.
- Dirección del sensor: 1010111.
- Entradas/salidas en open-drain.
- Tramas de 8 bits.

LA ESTRUCTURA DE LA ENTIDAD I2C MASTER

La entidad I2C Master se divide en 2 partes. La primera parte corresponde a la declaración de señales intermedias y las constantes. Entre ellas están las señales de estados, la dirección del esclavo, contadores y otras señales necesarias.

La segunda parte se trata de la arquitectura donde se derivan a su vez dos procesos. El primero es “clk_process” y genera dos señales de reloj (señales intermedias “scl_clk” y “data_clk”) desfasada un cuarto de ciclo con el fin de que sirva de apoyo a la máquina de estado (process_FsmI2C). El segundo proceso cambia a razón del flanco de subida del reloj de entrada de la entidad (50 Mhz) y a su vez de las señales “data_clk” y “data_clk_prev”. La señal “data_clk_prev” almacena el último valor de “data_clk” para garantizar que se produce un flanco de bajada o subida. Entonces, el flanco de subida de “data_clk” coincide con un flanco de bajada de la señal SCL (línea amarilla de figura 20), momento donde es posible introducir un dato en el bus sin ser un comando. Y en el caso de que sea un flanco de bajada (línea roja de figura 20) concurre con nivel alto de la señal SCL que da lugar a la posibilidad que se verifique el dato introducido o bien que se ejecute un comando.

Ilustración 20 "Data_clk" y "scl_clk" desfasados 1/4 de ciclo

Por otro lado, la señal intermedia scl_clk actualiza la señal de salida SCL en el momento de iniciar una comunicación ya que SCL estará en alta impedancia antes de comenzar las transiciones.

Profundizando en la máquina de estados, se pueden conocer los siguientes estados.

s_INCIO

En este estado se preparan los datos para empezar la comunicación y se pregunta por la señal “Inicio”. Se considera como estado partida, donde las señales SDA y SCL se encuentran en alta impedancia (no activas).

s_START

Se inicia la comunicación enviando al bus el primer bit de la primera trama que corresponde con la dirección del esclavo junto con el bit de escritura o lectura. Se ha recurrido a una señal intermedia “TX” que contiene los 8 bits a enviar y la señal “i” como contador descendente de bits enviados. Debe enviarse siempre el más significativo primero.

s_ESCRIBIR

Estado en el cual se envía los bits de datos. Además, se utiliza una señal intermedia “n_aux” como contador descendente para conocer el número de bytes a escribir.

s_ACKe

Se comprueba que el esclavo ha forzado la señal SDA del bus a nivel bajo como acción de reconocimiento. Si no es así, se activa la señal de error “DataError” pero en el caso de reconocerlo, se activa la señal intermedia de reconocimiento “ACK” y se cuestiona si se trata de una lectura o escritura.

Si es una lectura, a continuación se preguntará por la señal de control “ctrl_wr” la cual permite regresar al estado de s_INICIO o de s_LEER. Como una lectura consta de escritura y lectura, se ha de volver al estado de inicio (s_INICIO) para escribir de

nuevo la dirección del esclavo, aunque esta vez con la señal “RW” a nivel alto (modo lectura).

En el caso que sea escritura se preguntará por la señal de control “ctrl_wr” con la cual se regresa al estado de escritura (s_ESCRITURA) tantas veces como datos tenga que emitir, o parada (s_STOP) si se va a finalizar la transmisión. Según el valor de “ctrl_wr” se escribirá la dirección de registro interno o el dato a escribir en el registro interno del sensor.

s_LEER

En este estado, se recibe cada bit enviado por el sensor al I2C_master en cada flanco de bajada de la señal “Data_clk” y se almacena en la señal de 8 bits “RX”. Igualmente, como se ha hecho en escritura, se ha tomado una señal “i” para contar el número de bits a recibir y “n_aux” para el número de bytes a leer.

s_ACKm

En este estado, la entidad I2C Master manda el reconocimiento al esclavo forzando SDA a nivel bajo menos en la última trama que responde forzando SDA a nivel alto como se ha explicado en el apartado 3.1.2. Para este propósito, se utiliza una señal como bandera “flag_ackm” que se activa una vez que “n_aux” llegue a cero.

s_STOP

Estado de fin de comunicación donde se envía al sensor la condición de parada.

Desde otra perspectiva se puede ver el comportamiento de los estados en la siguiente carta ASM donde se ha eliminado las cajas de salidas condicionales para simplificar la carta (figura 21).

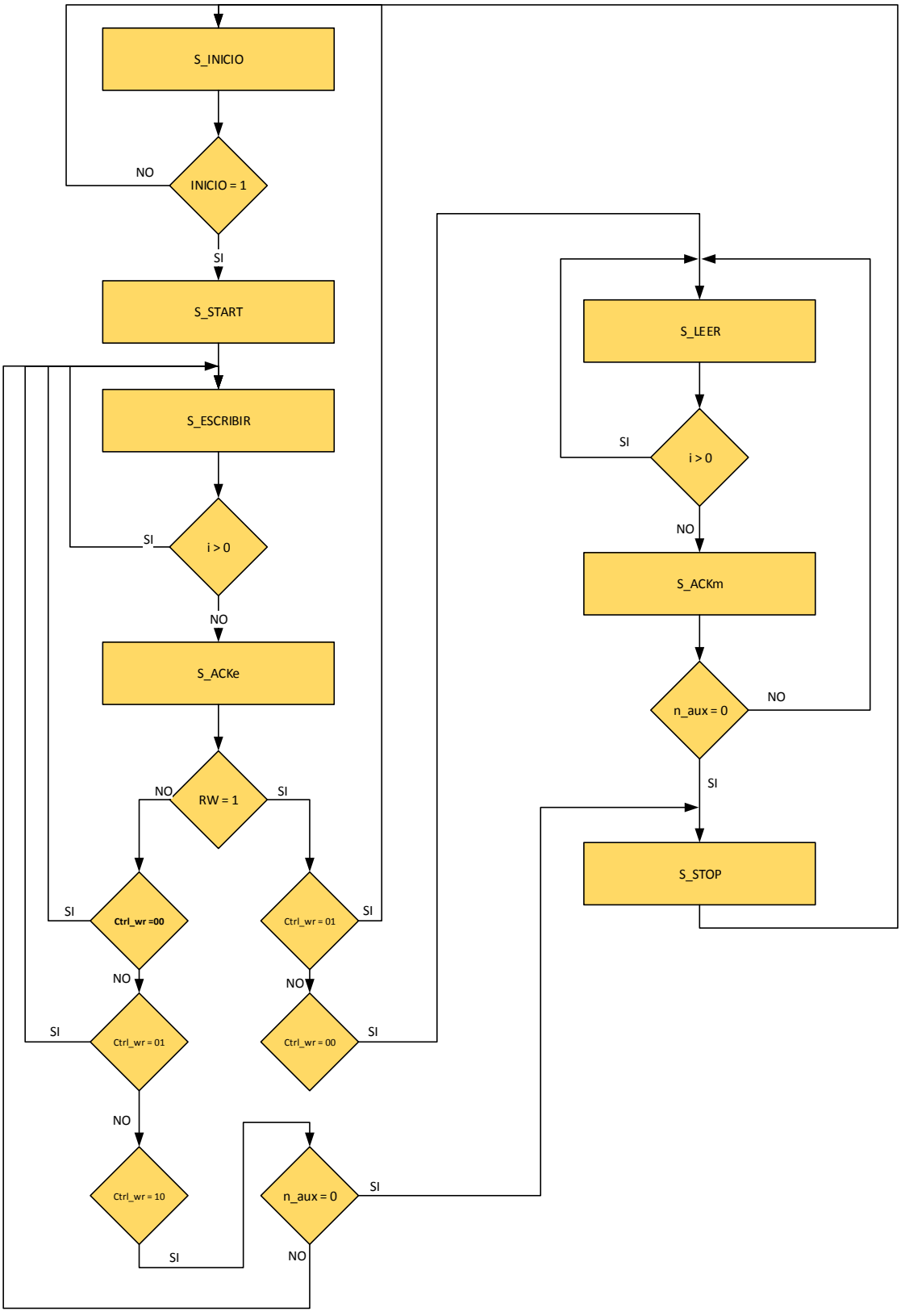


Ilustración 21 Carta ASM de la entidad "I2C_master"

3.3 VERIFICACIÓN FUNCIONAL

Como todo diseño digital es necesario una simulación no solo para comprobar el funcionamiento del circuito, sino para que en una posible ocasión de fallo, poder analizar qué está ocurriendo y qué señales no muestran un comportamiento correcto.

Se entiende por simulación de un circuito al comportamiento que puede llegar a tener dicho circuito de manera determinista. Se comprende que un cambio en una señal digital de salida tiene asociado un retraso de tiempo respecto al cambio de entrada que lo provoca, es por ello por lo que existen simulaciones con retraso y sin retraso. Para conocer el retraso es necesario tener información de las respuestas (sensor) externas esperadas, al no disponer lógicamente de dicha respuesta se han realizado solo simulaciones funcionales. Todo esto es posible para lenguaje VHDL gracias a los TestBench.

Un TestBench es una entidad de simulación del software Xilinx ISE que consiste en introducir valores lógicos en las entradas del circuito en un determinado tiempo y observando las salidas poder corroborar que funciona según lo esperado.

Por lo tanto, se han diseñado una serie de simulaciones que verifican el comportamiento del bus I2C en todas las situaciones posibles (condiciones de inicio, parada, escrituras, lecturas, etc).

En primer lugar, se probó la escritura de un solo dato en el bus. Con la ayuda de la señal “data_error” se pudo localizar donde se producían los errores. Era de esperar que los primeros errores fuesen por los no reconocimientos de datos en el bus de manera que la señal “ACK” (Acknowledge) no se activaba. El problema era que el esclavo forzaba la señal SDA a nivel bajo después del noveno pulso y daba lugar a la activación de data_error. Entonces, en el testbench se ajustó el tiempo de control del esclavo en el bus para el reconocimiento.

Una vez que funcionaba correctamente, se pasó a probar una lectura de dato. En esta prueba también surgió un error inesperado, no generaba el repeated start para cambiar el sentido de la transmisión debido a estar incompleto el circuito combinacional de la habilitación de “SDA” (“sda_ena_n”).

Las siguientes pruebas de escrituras y lecturas de datos se finalizaron correctamente.

Las siguientes imágenes representan parte del comportamiento del control de la interfaz con el testbench:

En la figura 22 se puede observar cómo la señal “inicio” se activa y da comienzo a la comunicación ya que “SDA” presenta un flanco de bajada estando “SCL” en nivel alto.

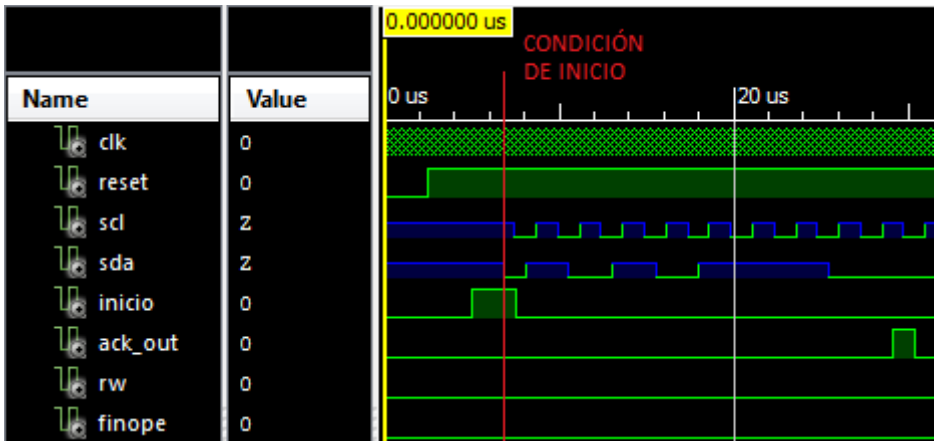


Ilustración 22 Condición de inicio.

La línea roja señala el flanco de subida de “SDA” en un nivel alto de “SCL” y posteriormente el fin de la comunicación (ver figura 23).

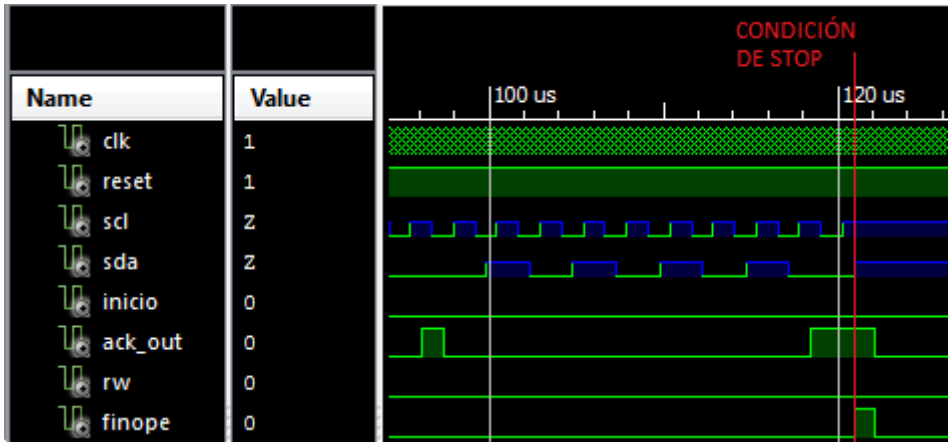


Ilustración 23 Condición de parada.

En la figura 24 hay que destacar que la condición de repeated start es importante que cambie en un ciclo de “SCL” como se muestra.

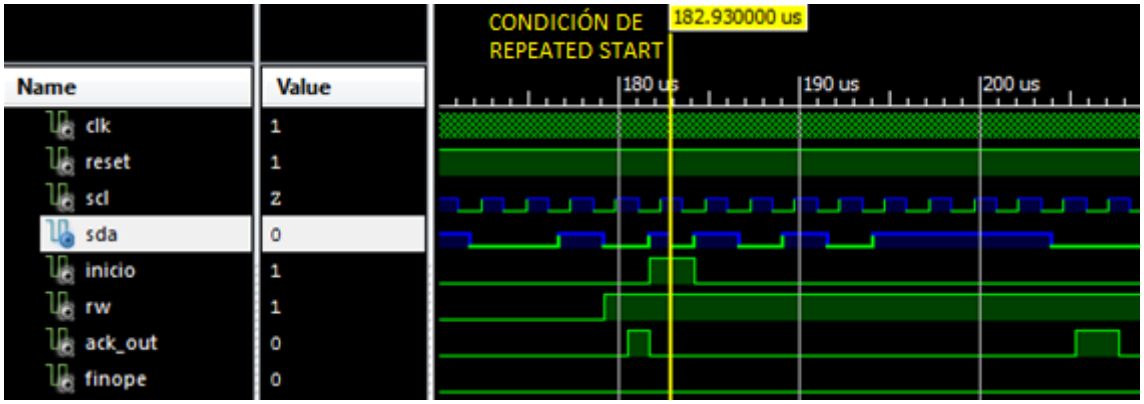


Ilustración 24 Condición de repeated start.

En la figura 25 se muestra la última trama que corresponde a datos a escribir.

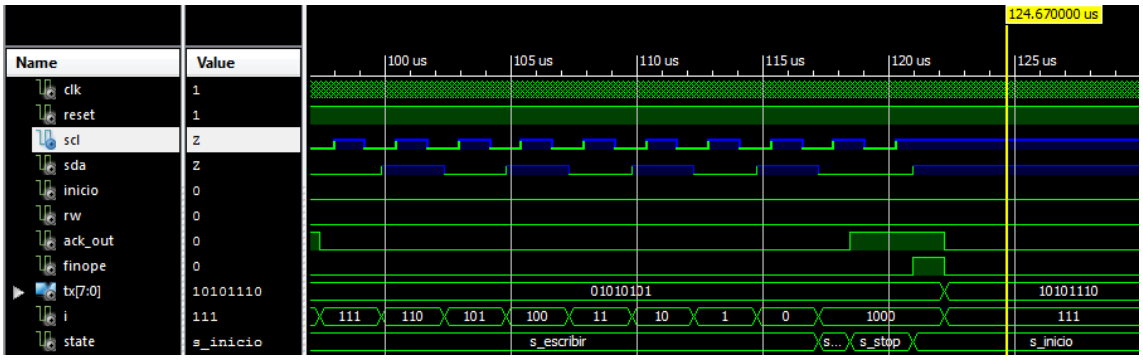


Ilustración 25 Última de datos escritos en el bus

En la figura 26 se muestra la última trama que corresponde a datos a leer y además se puede ver el procedimiento de NACK del maestro en el final de la comunicación.

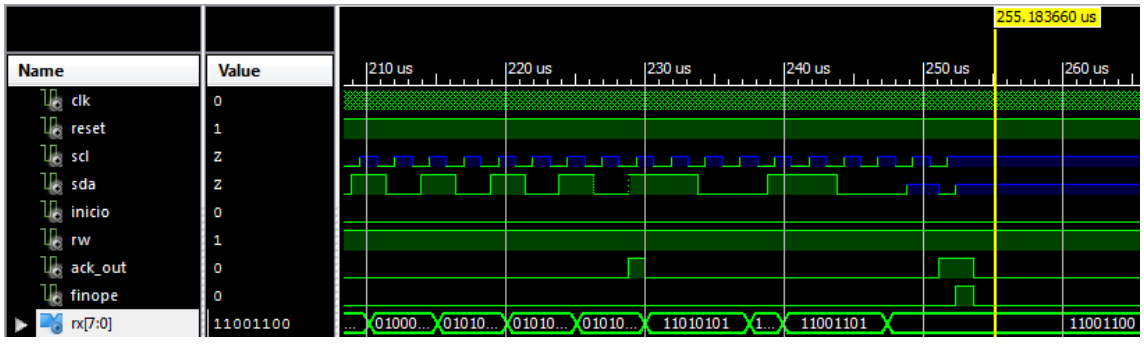


Ilustración 26 Última de datos leídos del bus

Una vez que la simulación resultó satisfactoria, se procedió a comprobar la entidad “I2C_Master” implementándola en la FPGA con un sensor menos complejo que el MAX30102 con la única finalidad de garantizar la funcionalidad de la entidad. El sensor elegido es de temperatura conocido como ADT7420 de la placa Nexys4 DDR.

Esta prueba experimental se efectuó en varios pasos para eliminar posibles fallos. Primeramente, se creó una nueva entidad la cual se encargaría de gobernar la entidad “I2C_Master”. Igualmente, la metodología seguida sería una máquina de estados. Tras este paso, se generó un testbench con los estímulos que presentaría el sensor (ver figura 27).

```
166  ----------------------------------------
167  --   TEMPERATURE MSB
168  ----------------------------------------
169  --   BIT   7 6 5 4 3 2 1 0
170  --   LEER  1 0 0 1 0 0 1 1
171  ----------------------------------------
172       sda_in <= '1';
173       wait for scl_period;
174       sda_in <= '0';
175       wait for scl_period;
176       sda_in <= '0';
177       wait for scl_period;
178       sda_in <= '1';
179       wait for scl_period;
180       sda_in <= '0';
181       wait for scl_period;
182       sda_in <= '0';
183       wait for scl_period;
184       sda_in <= '1';
185       wait for scl_period;
186       sda_in <= '1';
187       wait for scl_period;
188       ctrl_SDAsl <= '0';
189  --   ACK del maestro
190       wait for scl_period;
191
192  ----------------------------------------
193  --   TEMPERATURE LSB
194  ----------------------------------------
195  --   BIT   7 6 5 4 3 2 1 0
196  --   LEER  1 1 0 1 1 0 1 1
197  ----------------------------------------
198       sda_in <= '1';
199       ctrl_SDAsl <= '1';
200       wait for scl_period;
```

Ilustración 27 Captura de una parte del testbench.

Después de verificar la simulación funcional, se pasó a implementar y programar la entidad de control del ADT7420 en la Nexys 4 DDR mediante el software Adept de Digilent. Para visualización se utilizó los 16 leds de la placa.

A continuación, se ejecutó el primer código que constaba de una escritura en el registro de configuración, con la idea de posteriormente poder leer con el dato escrito. Una vez que este paso fue correcto, el siguiente sería repetir el paso anterior de lectura, pero con el registro de dato de temperatura.

Un detalle a destacar es el reloj de gestión de la máquina de estados que se ha originado a partir de una entidad DCM que se instancia en la entidad global como un componente. Un DCM (Digital Clock Managers) es una entidad que permite la generación de múltiples señales de reloj a partir del reloj principal de la placa. Con las ventajas de:

– Eliminan la desincronización (skew)
– Generar relojes con un desfase
– Generar relojes con frecuencias menores o mayores
– Acondicionan el reloj (50% en ‘1’ y ‘0’)
– Cambian los niveles de tensión y se integran directamente en las señales globales de reloj.

Otro aspecto importante es la adición de un fichero de restricciones al diseño (ver figura 28) para el uso de los elementos que ofrece la Nexys 4 DDR como es el sensor ADT7420, los leds, pulsadores, etc.

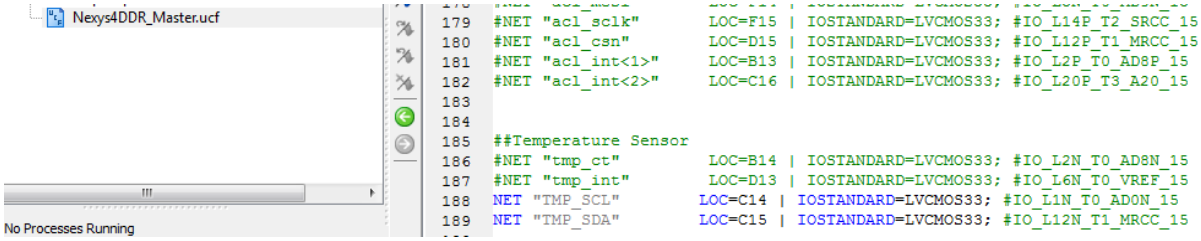


Ilustración 28 Fichero de restricciones

4 ADQUISICIÓN DE DATOS

La adquisición de datos es la parte encargada de capturar valores del sensor. Comprende desde el punto de contacto con la piel donde se encuentra el sensor hasta la comunicación con un sistema digital (FPGA, microcontroladores, pantallas visualizador de datos, etc) pasando por los subsistemas de acondicionamiento de señal, conversión de datos y almacenamiento en registros (S&H y ADC, filtros digitales, etc).

Como anteriormente se ha mencionado, el sistema MAXREFDES117 contiene estos bloques necesarios para la adquisición de datos. Además, el sensor MAX30102 posee gran flexibilidad ante el ajuste de estos elementos para conseguir una toma de datos a la altura de las exigencias del usuario. A continuación, se describen los elementos necesarios para la captura.

4.1 SUBSISTEMAS PARA LA ADQUISICIÓN DE DATOS

Entrado en detalle de los mecanismos de adquisición de datos, aparecen en primer lugar los leds. Son semiconductores que polarizados de manera directa emiten luz a una longitud de onda determinada definiendo el tipo espectro electromagnético, por lo que existe una gran cantidad de tipos de led.

Este sensor posee dos, un led rojo y un led infrarrojo (ver figura 29). El ancho de pulso es configurable y común a ambos y su amplitud independiente el uno del otro. Como es razonable, si se aumenta el ancho de pulso se tomarían más muestras, por tanto, se tendría más resolución. Como desventaja a un ancho de pulso grande, emitir ondas electromagnéticas durante más tiempo supone un aumento de la temperatura y en consecuencia un error en la medida. Este efecto es más prominente en el LED rojo. Es por ello, por lo que el fabricante proporciona un sensor de temperatura interno en el MAX30102 para compensar este efecto negativo.

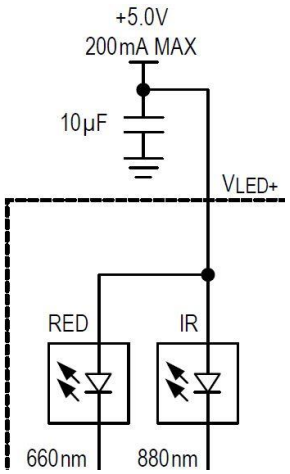


Ilustración 29 leds del MAX30102

Por otro lado, el fotodetector, fotodiodo o photodiodo es un diodo receptor del espectro electromagnético. Normalmente con un rango amplio, aunque su eficiencia cuántica está centrada en valores de interés en la medida como es el espectro visible e infrarrojo cuya longitud de onda es de 660nm y 880nm respectivamente. Según el datasheet del sensor, el fotodetector mide con una eficiencia cuántica de más del 50% para los valores de longitud de onda anteriores (ver figura 30).

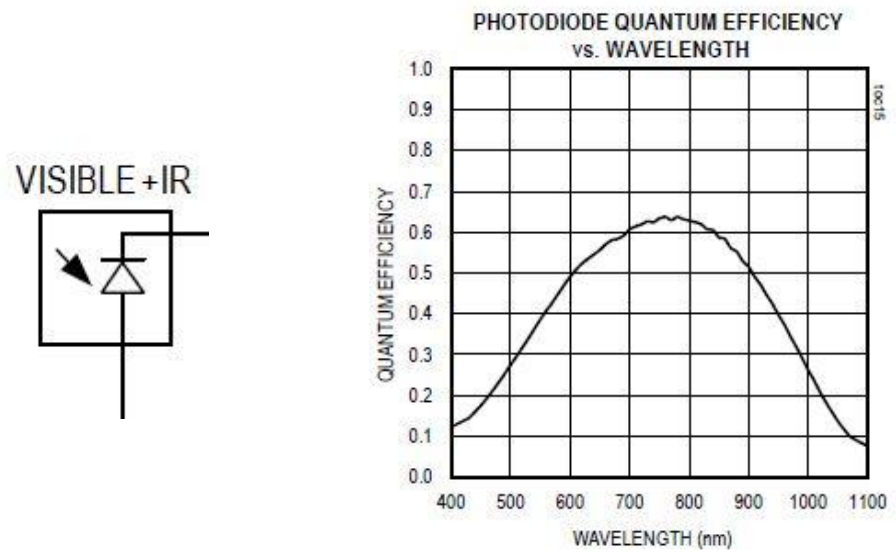


Ilustración 30 Esquema del fotodiodo y gráfica de eficiencia cuántica frente a longitud de onda

Por último, el convertidor analógico digital. Como se conoce, un convertidor analógico digital se divide en dos partes. La parte de muestreo y la parte de captura que pueden ser configuradas a gusto del usuario. Se configura a través de dos parámetros, la resolución y la frecuencia de muestreo. La placa actual tiene un ADC de 18 bits de resolución configurable desde 15 bits a 18 bits. Por otro lado, es posible configurar la frecuencia de muestreo desde un valor de 50 muestras por segundo hasta 3200

muestras por segundo. Este valor está relacionado con la resolución y ancho de pulso de los leds emisores de luz. En la figura 31 se pueden ver unas tablas con la relación existente.

SAMPLES PER SECOND	PULSE WIDTH (µs)			
	69	118	215	411
50	O	O	O	O
100	O	O	O	O
200	O	O	O	O
400	O	O	O	O
800	O	O	O	
1000	O	O		
1600	O			
3200				
Resolution (bits)	15	16	17	18

SAMPLES PER SECOND	PULSE WIDTH (µs)			
	69	118	215	411
50	O	O	O	O
100	O	O	O	O
200	O	O	O	O
400	O	O	O	O
800	O	O	O	O
1000	O	O	O	O
1600	O	O	O	
3200	O			
Resolution (bits)	15	16	17	18

Ilustración 31 Tablas de funcionamiento en el modo SpO2 y HR respectivamente

4.2 ENTIDAD DE CAPTURA DE DATOS

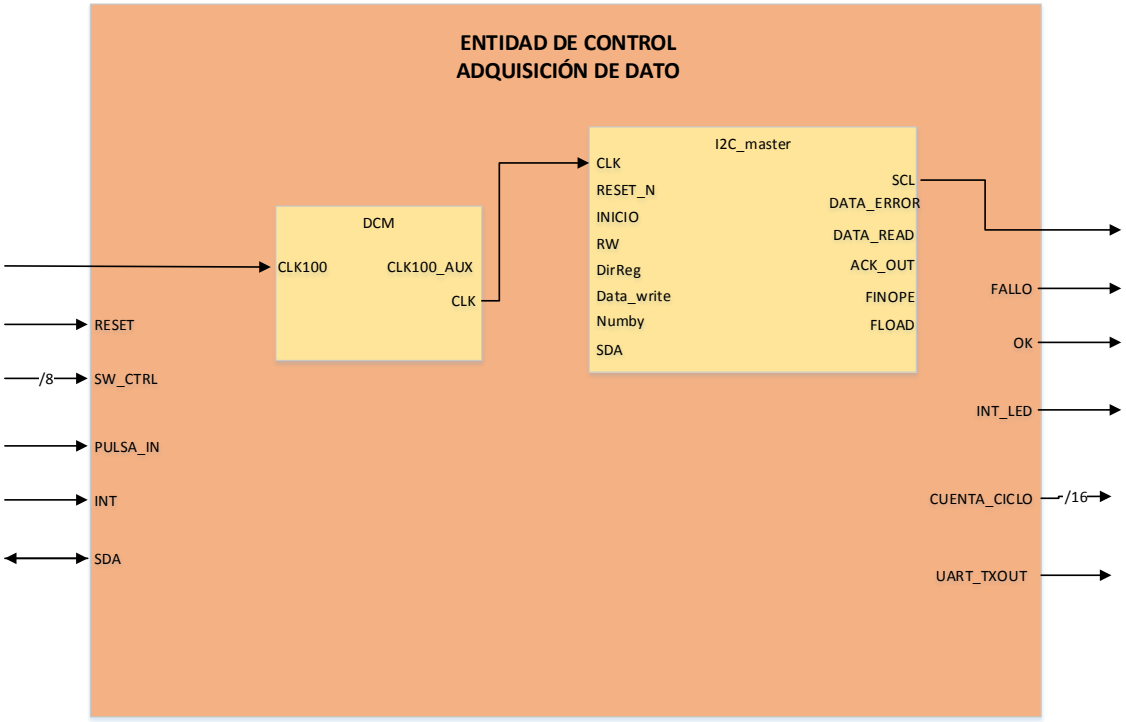


Ilustración 32 Bloque representativo de la entidad de control de la adquisición de datos.

4.2.1 FUNCIONAMIENTO

Desde un punto de vista global, la presente entidad gestiona a la entidad anteriormente mencionada de control de la interfaz “I2C_master”. Para ello se definen entradas, salidas y señales intermedias. Hay que tener en cuenta varios factores como por ejemplo la frecuencia de funcionamiento de esta nueva entidad, el uso de señales de validación similar a un protocolo Handshaking estando conectadas ambas entidades de manera que no aparezcan desincronizaciones, etc. También, es necesaria una configuración en el sensor para establecer las condiciones de funcionamiento.

La entidad se diseñó pensada para ejecutar una serie de instrucciones con la idea de realizar la configuración o la lectura de datos rápida y ordenadamente. Por esta razón se hizo un formato de instrucción determinado el cual está compuesto por la dirección del registro a acceder, el modo en el que se va a operar (lectura o escritura) y el dato a escribir o la dirección donde se almacena el dato leído (ver figura 33).

ADDR. REGISTER	**RW**	**DATO**	
8 bits	1 Bit	A escribir (8 bits)	Posición almacenada

Ilustración 33 Formato de instrucción diseñada

ADDR. Register: contiene la dirección del registro interno a leer o escribir.

RW: bit de escritura y lectura.

DATO: divida en dos partes, la primera tiene el dato a escribir y la segunda, la posición del banco de registro donde almacenar el dato leído que solo tuvo sentido en las primeras pruebas en las cuales se quería comprobar la operación de lectura y escritura.

4.2.2 TABLA DE INSTRUCCIONES

Para darle sentido a las instrucciones y ejecutarlas consecutivamente, se creó una tabla de instrucciones llamada “TABLA_INSTRU” con un tamaño de 25 bits y 32 posiciones. Se utiliza un puntero “p_instru” como mecanismo de acceso a esta tabla.

4.2.3 CONFIGURACIÓN

Como se ha señalado antes, en los registros del sensor se debe establecer unos valores determinados, es decir, una configuración de funcionamiento.

Los registros a configurar son MODE_CONFIG, INTE_ST1, INTE_EN1, INTE_EN2, Registros de la FIFO, FIFO_CONFIG, SPO2_CONFIG, LED1_PA, LED2_PA y PILOT_PA.

Primeramente, se comienza escribiendo en el registro “mode configuration” para resetear la configuración que tenía gravada el sensor y establecer unos valores de partida conocidos. A continuación, se procede a modificar una serie de registros que son los que establecen la configuración deseada. La configuración y la tabla de instrucciones del diseño se puede observar en la figura 34.

```
-----------------------------------------------------------------
-- INSTRUCCIÓN:       DIRECCIÓN    | RW |    DATO     | Posición
-----------------------------------------------------------------
-- CONFIGURACIÓN PREVIA
   TABLA_INSTRU(0) <= MODE_CONFIG & '0' & "01000000" & X"02";  --
   TABLA_INSTRU(1) <= INTE_ST1    & '1' & "00000000" & X"02";  --

-- INICIALIZACIÓN...
   TABLA_INSTRU(2) <= INTE_EN1    & '0' & "11000000" & X"03";  --
   TABLA_INSTRU(3) <= INTE_EN2    & '0' & "00000000" & X"03";
   TABLA_INSTRU(4) <= FIFO_WR_PTR & '0' & "00000000" & X"03";
   TABLA_INSTRU(5) <= OVF_COUNTER & '0' & "00000000" & X"03";
   TABLA_INSTRU(6) <= FIFO_RD_PTR & '0' & "00000000" & X"03";
   TABLA_INSTRU(7) <= FIFO_CONFIG & '0' & "01001111" & X"03";  --
   TABLA_INSTRU(8) <= MODE_CONFIG & '0' & "00000011" & X"03";  --
   TABLA_INSTRU(9) <= SPO2_CONFIG & '0' & "00101000" & X"03";  --
   TABLA_INSTRU(10)<= LED1_PA     & '0' & "00100100" & X"03";  --
   TABLA_INSTRU(11)<= LED2_PA     & '0' & "00100100" & X"03";  --
   TABLA_INSTRU(12)<= PILOT_PA    & '0' & "01111111" & X"03";  --
```

Ilustración 34 Instrucciones de configuración.

4.2.4 BANCO DE REGISTRO

Con la intención de concentrar la cantidad de datos que mide el sensor, se planteó un banco de registros de un tamaño de 8 bits y 4096 posiciones. Debe cubrir la cantidad de posiciones de memorias para 500 muestras de datos, es decir, 3000 posiciones (500 muestras por 6 bytes por muestra). Como consecuencia de esto, y estructurándolo como una memoria, se tomó 4096 posiciones (potencia de 2). Así, en la síntesis se considera este circuito como una memoria RAM distribuida y no como un conjunto de registros (ver figura 35). Las memorias distribuidas se implementan de manera que quedan repartidas por el dispositivo a través de los recursos internos que ofrece la arquitectura de la FPGA. Esto da como resultado un rutado diferente a si lo hace con bloques de memoria existentes. La función de esta memoria es recibir datos en un estado de almacenamiento de la máquina de estados que se explica en el apartado 4.2.5.

```
110  ----------------------------------------------------------------------
111  -- RAM DISTRIBUIDA (BANCO DE REGISTROS)
112  -- Tamaño de 8 bits y 4096 posiciones de memoria.
113  ----------------------------------------------------------------------
114    type mem is array(0 to 4096) of std_logic_vector(7 downto 0);
115    signal RegBank: mem;     -- Señal que representa la memoria
116    signal p_bank: integer; -- Puntero de memoria
117
```

Ilustración 35 Memoria RAM distribuida.

4.2.5 MÁQUINA DE ESTADO

El punto principal de la entidad encargada de la adquisición de datos es la máquina de estados que controla a la otra entidad "I2C_master" para la captura de datos. Esta máquina de estados finitos está sincronizada con los flancos de subida de una señal de reloj de 50 MHz. Para obtener este reloj, se utiliza un componente llamado DCM a partir del reloj de la placa Nexys de 100 Mhz. En la figura 36 se muestra este ajuste.

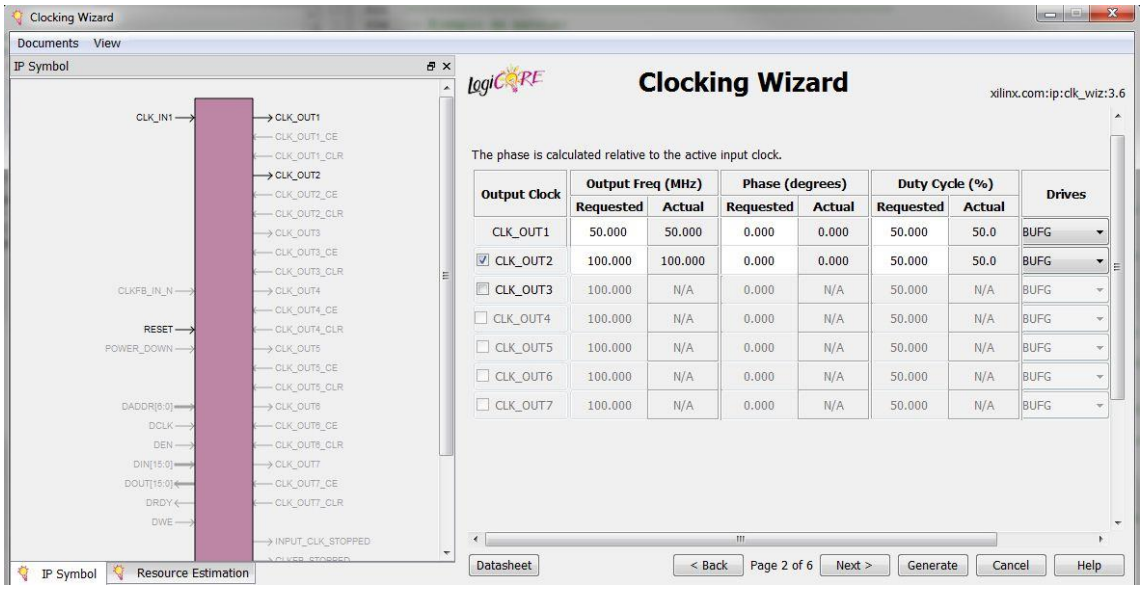


Ilustración 36 Ajuste de DCM.

Un reset asíncrono, restablece los valores de partida.

La máquina de estados está dividida en dos partes muy análogas, pero con distinta funcionalidad. La primera se centra en la configuración mientras que la segunda se ocupa de esperar a la interrupción para gestionarla y leer el dato.

La configuración comienza con una inicialización de las señales a usar. Después en el siguiente estado, almacena en un registro temporal la instrucción leída de la tabla de instrucciones. A continuación, las señales que proporcionan información al controlador del bus (“DirReg”, “RW”, etc) se cargan con los valores según la información de la instrucción y espera a la entidad “I2C_master” que confirme la carga. Una vez es confirmada, la entidad de control de la adquisición de datos da comienzo a la comunicación a través de una señal “start”. Llegando a esto, se espera a la señal “fin de operación” que envía la entidad “I2C_master” a esta para avisar de que ha finalizado la comunicación. Hay que destacar las esperas intermedias de la señal de reconocimiento para continuar con la comunicación hasta su finalización. Finalmente, se llega a un estado de parada donde se evalúa si se ha terminado la configuración. Si se ha concluido, cambia al estado de espera de interrupción. Si no ha finalizado, regresa al inicio de la máquina para proseguir con la configuración.

La última parte se diferencia de la primera por almacenar datos en la RAM distribuida y esperar la señal de interrupción “INT”. En la figura 37 se puede ver el estado donde se almacena el dato leído del sensor.

```
615         when s_STORE_PICK =>
616           start_max <= '0';
617           if fStore = '1' then  -- No entra la primera vez (Dir. Esclavo + RW)
618             RegBank(p_bank) <= data_in;
619             p_bank <= p_bank + 1;
620             fStore <= '1';
```

Ilustración 37 Estado de almacenamiento

En la figura 38 se ve el estado de espera de la interrupción.

```
527   ----- INICIO DE TOMA DE MUESTRAS (6 bytes/muestra) --------
528         when s_BEGIN_PICK =>
529         start_max <= '0';
530           if INT = '0' then   -- Interrupción activa
531             state <= s_INSTRU_PICK;
532             CE_CICLO <= '1';
533           else
534             state <= s_BEGIN_PICK;
535           end if;
```

Ilustración 38 Estado de espera de Interrupción

4.2.6 CAPTURA Y ALMACENAMIENTO DE MUESTRAS

La captura como se ha mencionado, se realiza leyendo del registro de datos del sensor a través de los subsistemas hasta llegar a la memoria RAM distribuida de entidad pasando por el bus I2C.

Según la estructura de datos que presenta la FIFO descrita en el apartado 3.1.2, los datos del canal rojo y del canal infrarrojo están localizado consecutivamente. Así pues, si se quisiera visualizar o procesar sería necesario separarlos y agrupar los datos de un mismo canal.

4.3 VERIFICACIÓN DE LA CAPTURA DE DATOS

Para comprobar el funcionamiento de la entidad de adquisición de datos, se realizaron una gran cantidad de pruebas experimentales de manera que este apartado relata la experiencia de manera resumida y práctica.

4.3.1 PRUEBA INICIAL

Una primera versión de la entidad que controla la adquisición, se basaba en la entidad de control del sensor de temperatura ADT7420 que se presenta en el capítulo 3, con el fin de probar la escritura y lectura en uno y varios registros.

El método de verificación es visual mediante el encendido de la fila de 16 leds que proporciona la placa.

Tras comprobar el funcionamiento se plantea una configuración determinada para captura de datos del registro de la FIFO que se expresa en el siguiente apartado.

4.3.2 CAPTURA DE 32 MUESTRAS

Para esta prueba se planteó modificar los registros del sensor a través de una serie de instrucciones que escribían una configuración para poner el sensor en funcionamiento y poder tomar datos de este a frecuencia de muestreo de 100 muestras por segundo.

Se diseñó una entidad que permitía verificar paso a paso las operaciones y para esto se utilizó uno de los pulsadores de la Nexys donde dicho pulsador se conectó a una señal interna de la entidad. De esta manera podía ejecutarse una instrucción detrás de otra. A esta entidad hubo que añadir un componente que eliminase los rebotes

producidos por el pulsador. Este componente se llama “debounce” y su funcionamiento se basa en un temporizador que desde que se acciona el pulsador, espera un determinado tiempo a que se estabilice la señal para detectar claramente si es “1” o “0” lógico, es decir, si se ha pulsado o no. El componente “debounce” se ha tomado de una de las fuentes.

Por último, se dio uso a los Switches (interruptores de la Nexys) como puntero de la memoria. Así pues, se podía mostrar en los leds los valores de cada posición de memoria. Para ello, harían falta tantos interruptores como tamaño en bits tuviera el puntero de memoria. En este caso, se utilizó 8 switches para direccionar las 32 muestras.

Una vez finalizada la configuración manualmente, se esperaba a la señal de interrupción “INT” que previamente se había conectado a un LED de color de la placa Nexys con el propósito de comprobar la activación de dicha interrupción. En el caso de que esté activa, quedaría accionar el pulsador para leer el registro. En todo momento de la prueba se puede ver el contenido en memoria gracias a los switches y leds que indican la posición de memoria y el dato respectivamente.

Resumen del funcionamiento de la entidad:

- Accionar pulsador para ejecutar una instrucción.
- Leer instrucción.
- Preparar datos intermedios para la comunicación
- Comprobar que el bus I2C está libre.
- Ejecutar la instrucción, activando la entidad “I2C_master”.
- Esperar a los ACK necesarios en el modo lectura para leer 1 bytes o las 32 muestras de la FIFO.
- Esperar a la señal “FinOpe” para terminar.
- Si se han leído datos almacenar en memoria.
- Comprobar si es la última instrucción, si lo es parar máquina de estado (estado STOP) y si no lo es, volvemos al primer paso.

4.3.3 TRANSFERENCIA DE DATOS CON UART

En este apartado se presenta la comunicación de la Nexys 4 DDR con un hiperterminal (PC) para la transferencia de datos mediante la comunicación UART.

El puerto UART (Universal Asynchronous Receiver-Transmitter) que dispone la placa Nexys 4 DDR es un puerto de comunicación serie asíncrona para la transmisión y recepción de datos de la placa a un PC u otro sistema de procesado. Este se apoya en el bus USB (Universal serie bus) que hace de intermediario para unir el puerto UART de la placa con el puerto COM de un ordenador. Con el cable USB se programa la Nexys y también se transmite la información. Hay que tener en cuenta que las

transmisiones tienen un tamaño de 8 bits por trama y que la velocidad es de 9600 baudios (símbolos por segundo).

Se generó un proceso “process_UART” dentro de la arquitectura de la entidad adquisición de datos con el fin de gestionar la comunicación UART y poder trasmitir los datos almacenados en la memoria al PC. Diligent aporta un demo de la UART con los recursos de la Nexys 4 DDR donde se probó el software del hiperterminal que más abajo se describe.

Este proceso está basado también en una máquina de estados finitos cuya señal de reloj es 100 MHz.

Una vez que la máquina de estado encargada de la captura de datos tiene almacenada los datos se procede a iniciar la máquina de estados que controla la comunicación UART cuyo funcionamiento de este proceso es el siguiente.

Se empieza con una inicialización de las señales intermedias a utilizar. Después se toman 3 bytes de memoria RAM. A continuación, se agrupan los 3 bytes en un registro temporal de 24 bits y se le aplica una máscara (ver figura 39). Esta máscara elimina los bits no deseables de los 3 bytes. Los 6 bits del primer byte se eliminan porque no tienen uso y los 3 bits menos significativos del tercer byte se eliminan o se mantienen en función de la resolución elegida.

	Primer byte recibido								Segundo byte								Tercer byte							
BIT							14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	-	-	-	-	-	-	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-	-	-

Ilustración 39 Máscara aplicada para 15 bits de resolución.

Se buscó una forma de codificar los datos almacenados para que al enviarlos por la UART los interpretara como caracteres ASCII y no comandos. De este modo, se podían enviar tanto datos (caracteres y números) como comandos ASCII (nueva línea, retorno de carro, etc). Para ello, se recurrió al agrupamiento en paquetes de 4 bits de los datos enmascarados (en este caso, 15 bits) en un registro temporal “temp_ASCII”. Entonces, se diseñó un circuito combinacional que diera como resultado los correspondientes caracteres ASCII en función del valor que toma “temp_ASCII”. Para luego más tarde poder representar en el PC.

Estos pasos se realizan tantas veces como datos se quieran enviar. Según la resolución impuesta para cada muestra sería necesaria 4 transmisiones de 8 bits. En conclusión, es una comunicación lenta (9600 baudios) pero de gran utilidad. En la figura 40 se presenta el inicio de la comunicación UART.

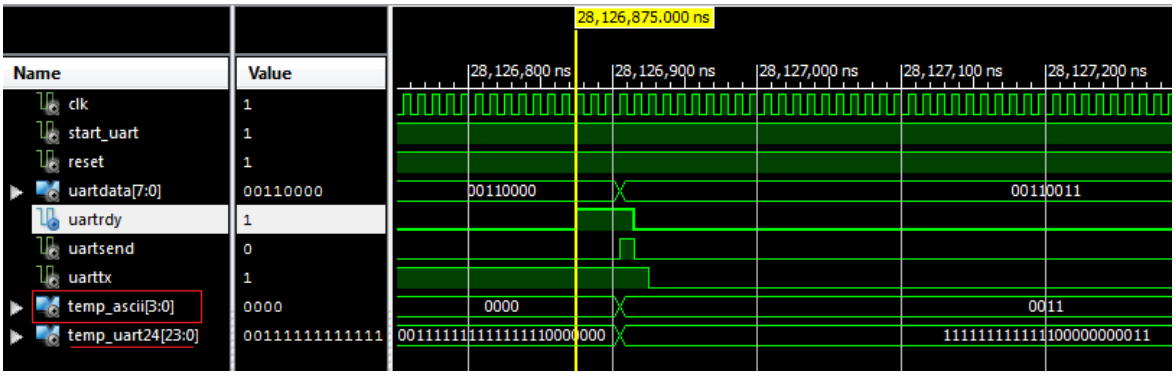


Ilustración 40 Captura de la simulación de la comunicación UART.

Para recibir los datos es necesario el uso de un programa en Windows u otro sistema operativo. Hay muchos programas para la adquisición de datos, pero se eligió uno sencillo para agilizar las pruebas. En la figura 41 puede verse algunos resultados de la adquisición. También hay que señalar, que es necesario conocer y configurar la aplicación con las características del puerto UART a usar como velocidad, paridad, bits a trasmitir o recibir, etc.

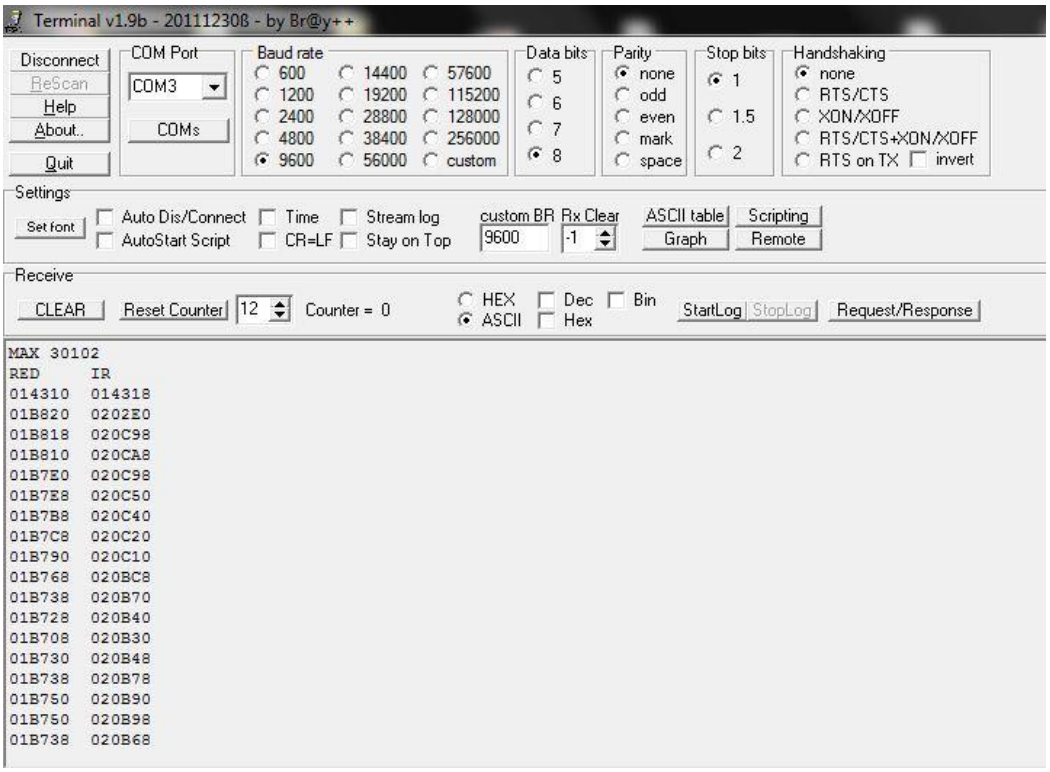


Ilustración 41 Software para la comunicación USB-UART.

4.3.4 CAPTURA DE 500 MUESTRAS Y TRANSFERENCIA AL PC

Llegado a este punto, se amplió la cantidad de muestras a medir ya que con 32 muestras sería difícil visualizar un pulso cardíaco. El objetivo de esta prueba es visualizar una gran cantidad de muestras y analizar los resultados para distintas frecuencias de muestreo y resoluciones mediante la misma comunicación UART que se explicó en el apartado anterior.

Lo primero, es modificar la entidad que capturaba 32 muestras. Aprovechando la estructura de esta última, se ha rediseñado de tal manera que no se lee la FIFO completa (32 muestras) una vez sino que se lea muestra a muestra por cada interrupción. Este aspecto supone automatizar las ejecuciones de instrucciones eliminando la opción del pulsador como único uso en el arranque.

Esta entidad ha sufrido muchos cambios de mejoras gracias a la cantidad de pruebas que se han llevado a cabo.

La primera prueba para conseguir procesar las 500 muestras a una frecuencia de 100 muestras por segundo fue la de automatizar el pulsador. Entonces, se empezó esperando la señal de interrupción “INT” pero nunca se desactivaba y por consiguiente bloqueaba el funcionamiento. Por este motivo, se intentó resolver la toma de datos por otro camino alternativo al apartado 4.3.4.

La solución fue configurar el sensor de manera semiautomática, activando un pulsador de la placa cuando se encendiera el led de indicación de interrupción de manera que se fue tomando de 32 en 32 muestras. Repitiendo lo mismo para las muestras restantes, es decir, 96 muestras en total.

Al funcionar correctamente se pasó a reducir el número de muestras a capturar con la idea de llegar un flujo de 1 muestra por interrupción. La prueba fue exitosa y se eliminó la acción del pulsador esperando la interrupción para leer muestra.

Más tarde, se modificó la entidad de la adquisición de datos para que leyera 300 muestras al ritmo de 1 muestra por interrupción. Para ello, se incorporó al diseño un contador de ciclos de reloj. Este contaría el número de ciclos que pasan desde la primera interrupción hasta la última y cuyo valor se representaría en los leds de la placa. Además, se amplió el número de captura a 500 muestras.

Con el contador de ciclos se pudo calcular un tiempo de aproximadamente 10 segundos cada 500 muestras capturadas. Esto quería decir que el flujo de datos era de 50 muestras por segundos. Por lo tanto, con esta referencia de tiempo se podría

conocer la frecuencia cardíaca de los datos capturados. A partir de aquí ya se pudo analizar los resultados con una base temporal fiable.

RESULTADOS

Hay que recordar que los datos con los que se ha trabajado son los trasmitidos al PC mediante la comunicación UART y el software anteriormente relatado.

La prueba está estructurada de la siguiente manera en base a 3 parámetros relacionados. La frecuencia de muestreo, la resolución y el ancho de pulso, siendo estos dos últimos dependientes. El objetivo de la prueba no es más que seleccionar la mejor configuración, es decir, la que contenga menor componentes de alta frecuencia (ruido o vibración), menores picos erróneos y más parecida a una fotoplestimografía.

En primer lugar, se hizo un barrido a una frecuencia de 100 muestras por segundo y ancho de pulso desde 69 us (15 bits de resolución) hasta 411 us (18 bits). En la figura 42 y 43 puede verse algunos de los resultados. A continuación, se configuró a una frecuencia de 200 muestras por segundo y tomando resolución de 15 bits (ver figuras 44 y 45). Más tarde, se configuró a una frecuencia de 400 muestras por segundo y tomando resolución de 18 bits (ver figuras 46 y 47). Por último, se configuró a una frecuencia de 400 muestras por segundo y tomando resoluciones de 15 y 18 bits (ver figuras 48 y 49).

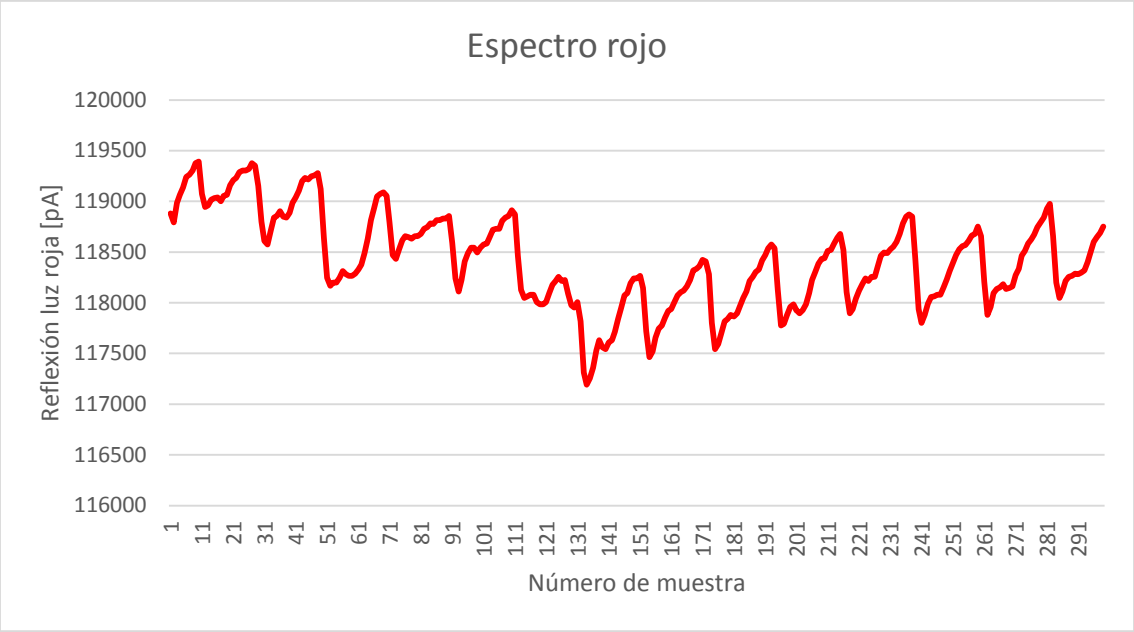


Ilustración 42 Espectro rojo a 15 bits resolución 100 muestras/s frecuencia de muestreo.

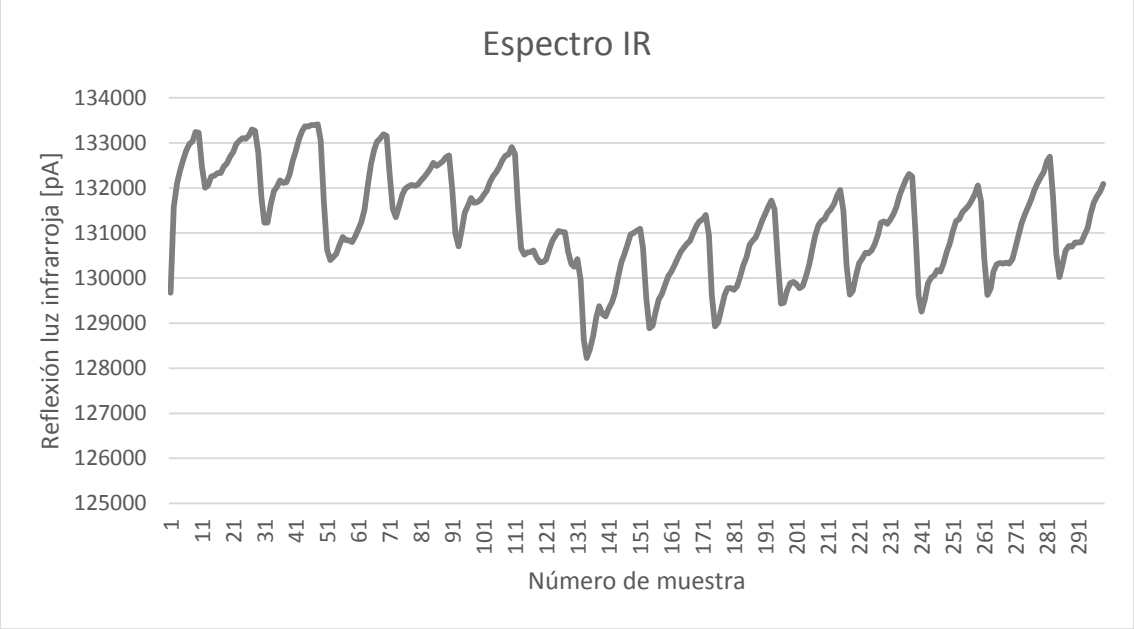


Ilustración 43 Espectro infrarrojo (IR) a 15 bits resolución 100 muestras/s frecuencia de muestreo.

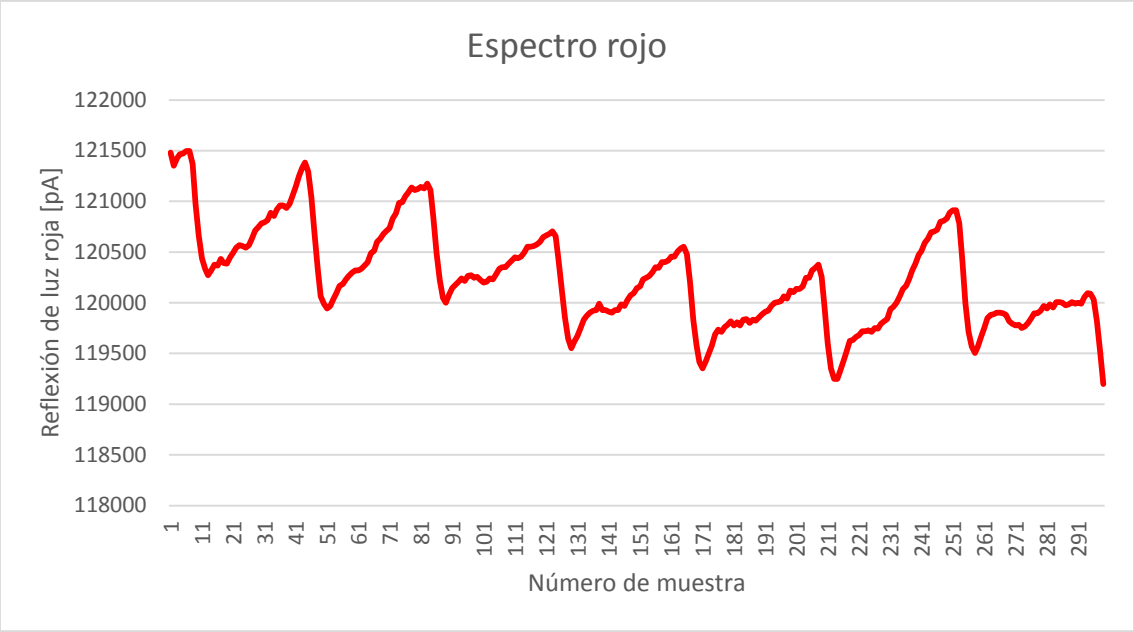


Ilustración 44 Espectro rojo a 15 bits resolución 200 muestras/s frecuencia de muestreo.

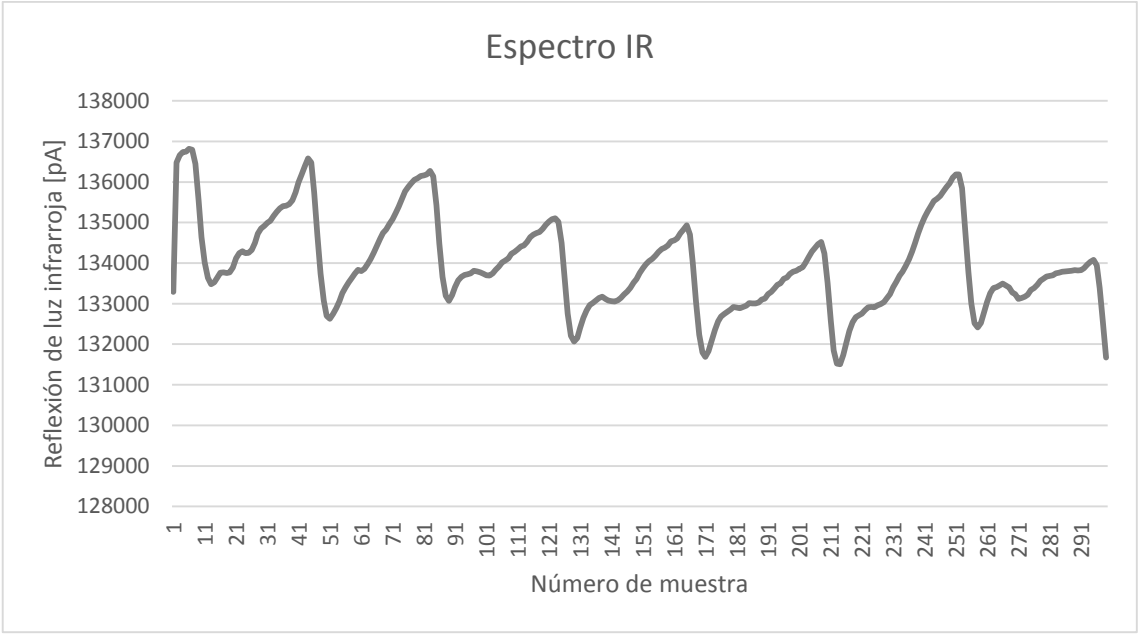


Ilustración 45 Espectro infrarrojo a 15 bits resolución 200 muestras/s frecuencia de muestreo.

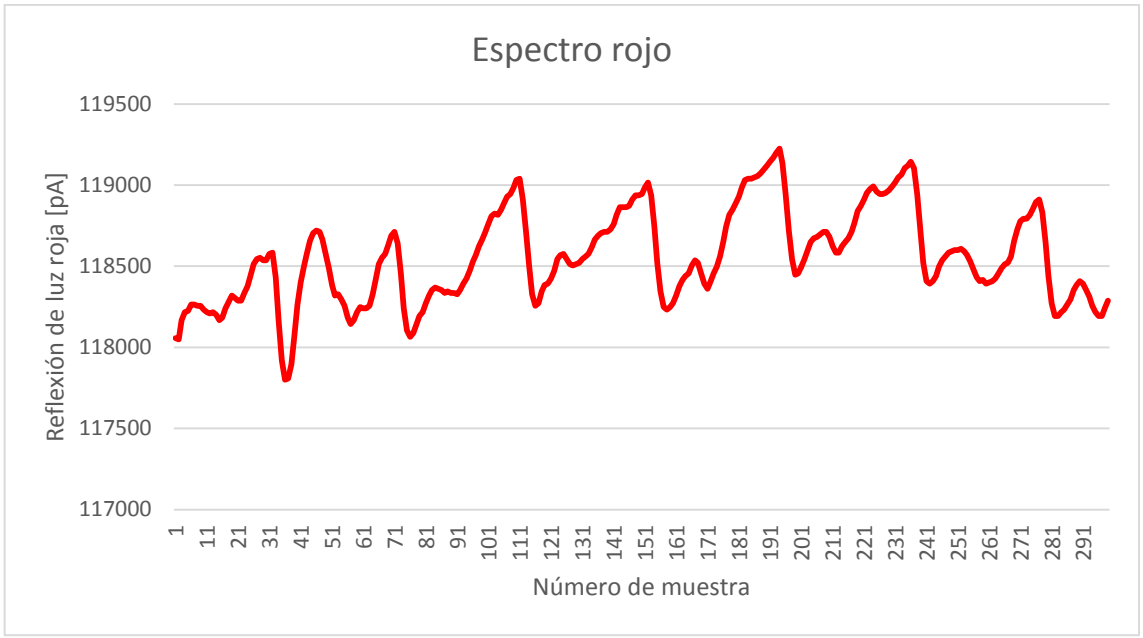


Ilustración 46 Espectro rojo a 18 bits resolución 200 muestras/s frecuencia de muestreo.

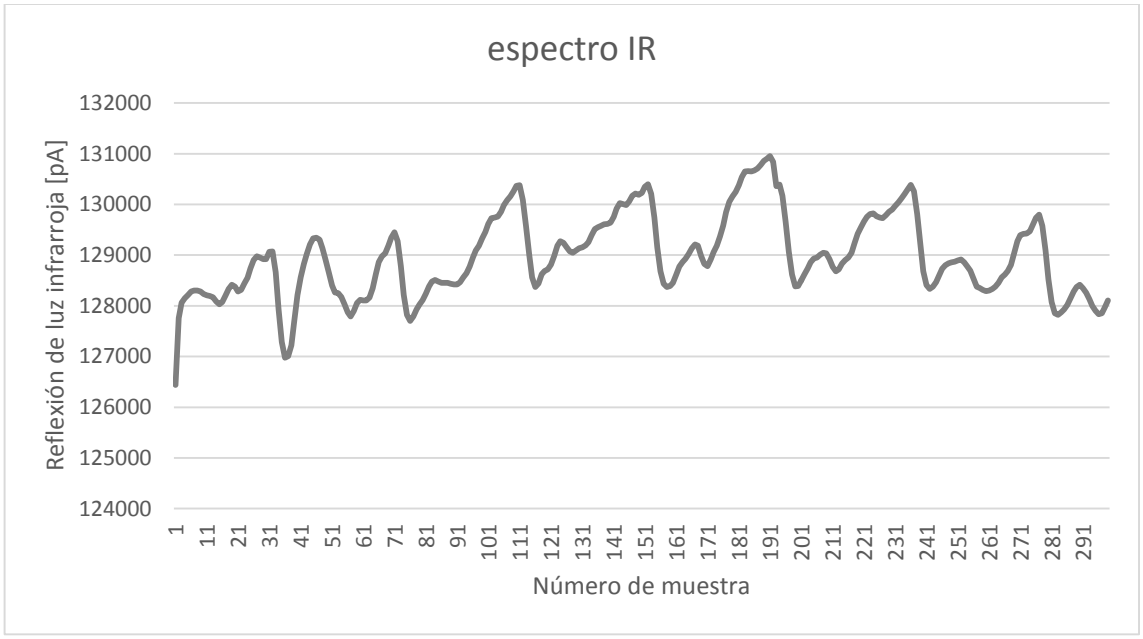


Ilustración 47 Espectro infrarrojo a 18 bits resolución 200 muestras/s frecuencia de muestreo.

Ilustración 48 Espectro rojo a 15 bits resolución 400 muestras/s frecuencia de muestreo.

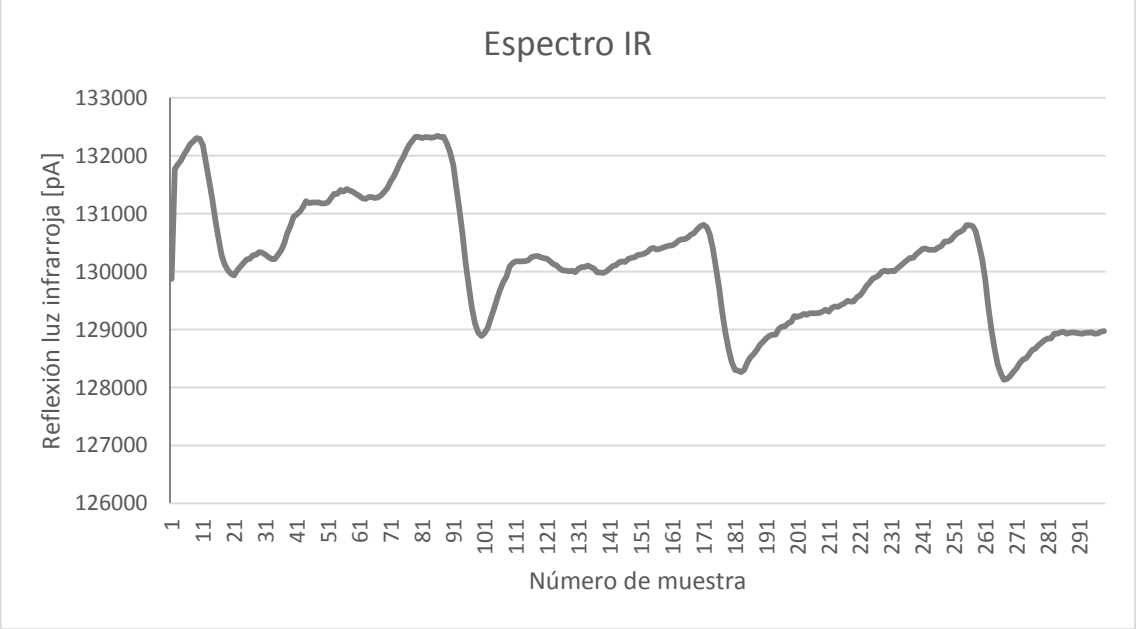


Ilustración 49 Espectro infrarrojo a 15 bits resolución 400 muestras/s frecuencia de muestreo.

Hay que señalar que el tiempo proporcionado por el contador de ciclos era aproximadamente de 10 segundos para todas las pruebas.

Después de esta variedad de pruebas, se puede observar que los resultados son similares con algunas diferencias. Con el fin de tener una gran cantidad de muestras para ritmos cardíacos altos, se concluye que la frecuencia de muestreo a establecer es de 200 muestras por segundo con una resolución de 15 bits ya que la resolución de 18 bits contenía falsos picos que pueden dificultar el cálculo del ritmo cardíaco (ver figura 50 y 51).

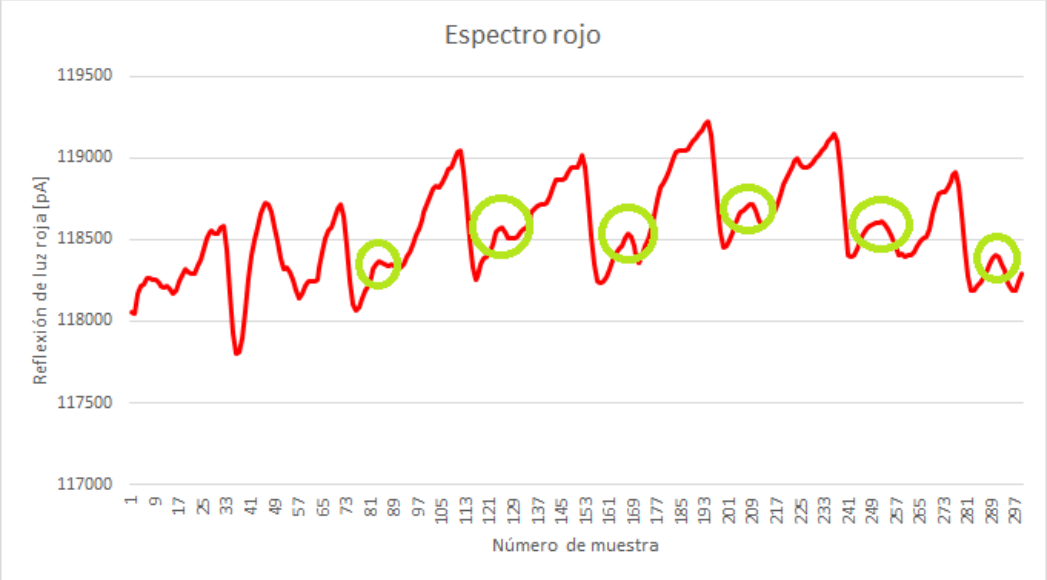


Ilustración 50 Falsos picos

SAMPLES PER SECOND	PULSE WIDTH (µs)			
	69	118	215	411
50	O	O	O	O
100	●	●	●	●
200	●	O	O	●
400	●	O	O	●
800	O	O	O	
1000	O	O		
1600	O			
3200				
Resolution (bits)	15	16	17	18

Ilustración 51 Punto de trabajado elegido

Respecto al tipo de canal a usar, se ha seleccionado el canal infrarrojo ya que contienen menor ruido que el rojo, como se puede observar en las figuras anteriores.

Uno de los aspectos a tener en cuenta cuando se realice una medida es no presionar demasiado el sensor. En términos fisiológicos supone no dejar pasar nuevo flujo sanguíneo y como consecuencia de esto, no trasmite la parte pulsátil. Este hecho implica una señal casi plana sin elevados y continuos picos necesarios para la medida del ritmo cardíaco (ver figuras 52 y 53). La medida en estas condiciones es errónea.

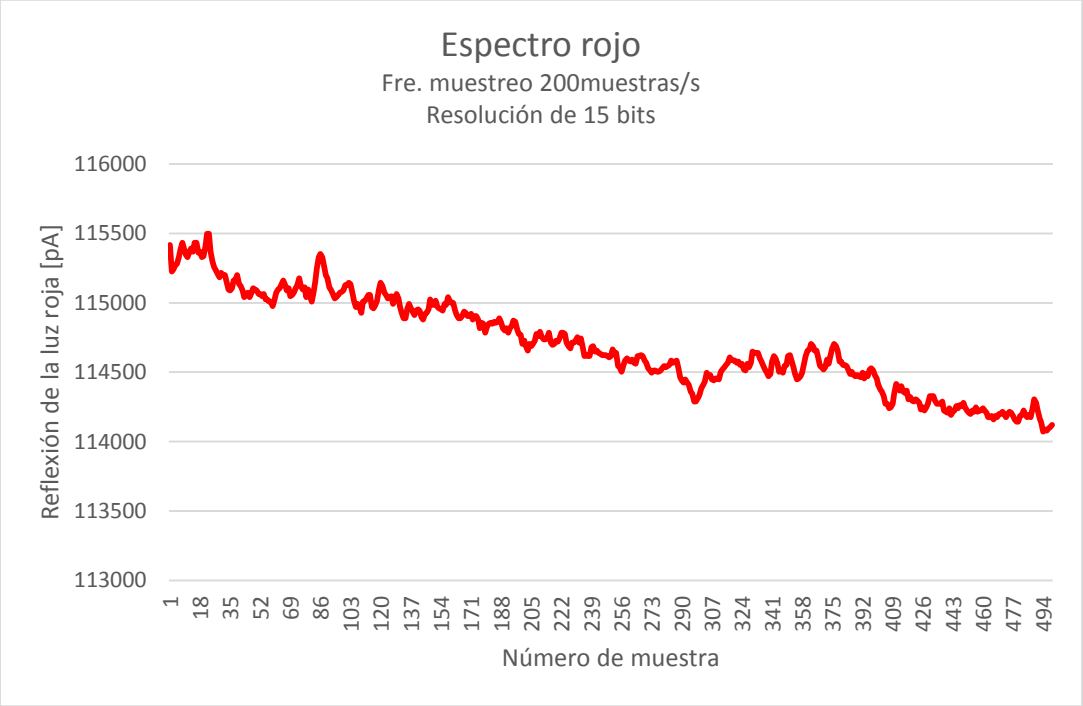


Ilustración 52 Deficiencia en la medida por presión considerable, espectro rojo.

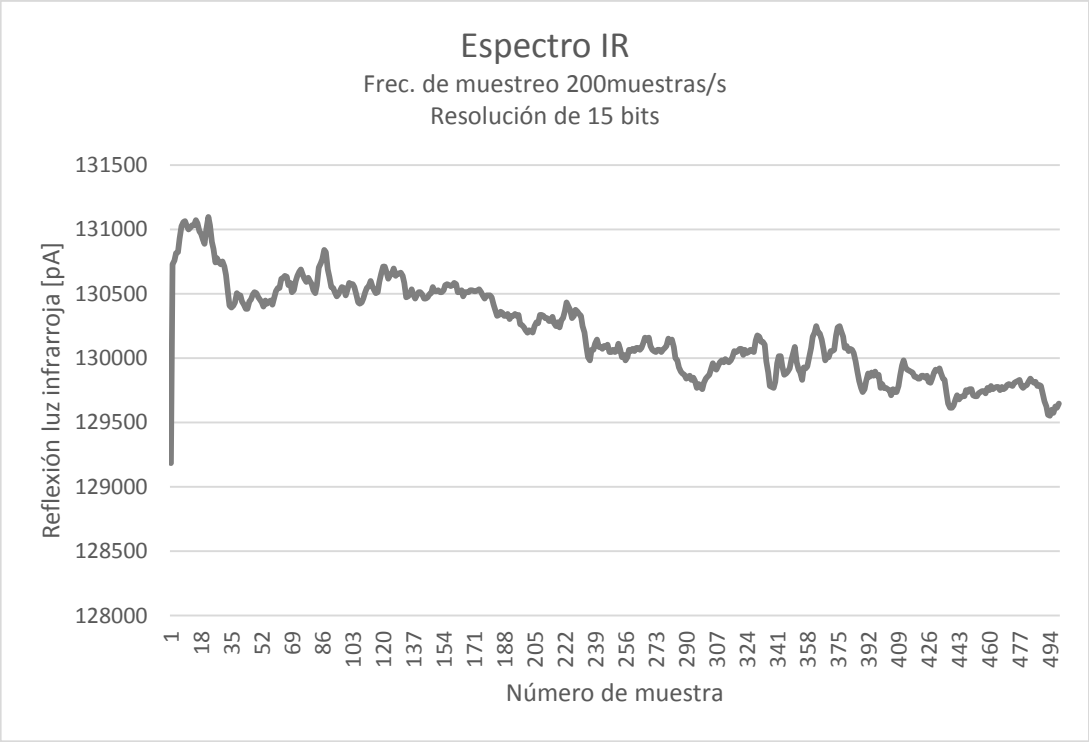


Ilustración 53 Deficiencia en la medida por presión considerable, espectro infrarrojo.

Otro detalle a tener en cuenta es la posición en la que se apoya la huella del dedo en el sensor. Es muy importante que se sitúe la yema del dedo y no el extremo final.

En definitiva, con la configuración final establecida, se realizó 10 medidas de 500 muestras. Los datos adquiridos se emplearon como datos reales para las simulaciones del procesado que se explicará más adelante.

5 CÁLCULO DEL PULSO CARDÍACO

Todo sistema que tenga que hallar un valor determinado a partir de una señal, memoria o datos conocidos requiere de un procesado de datos. Un procesado de datos se centra en el cálculo a nivel de registros, puertas lógicas, biestables y todo elemento digital necesario para resolver una o varias operaciones. Estos elementos son utilizados a través de un código con una herramienta de programación como es ISE Design de Xilinx. En este caso, se llegará a conocer el ritmo cardíaco mediante los valores o datos que proporciona el sensor MAX30102 con la configuración oportuna.

5.1 ALGORITMO

La palabra algoritmo tiene origen árabe durante la época de la Edad Media con ámbito de la matemática y del cálculo. Desde el punto de vista de los sistemas digitales, un algoritmo es un conjunto de instrucciones consecutivas donde las cuales, utilizando los recursos de computación del sistema, hallan un valor final a partir de uno o varios valores iniciales conocidos.

Si se observa las gráficas resultantes del capítulo 4, es intuitivo conocer el ritmo cardíaco ya que cada pico o máximo de la señal corresponde con un pulso del corazón. Los máximos son los puntos de presión sistólica mientras que los valles o mínimos son entendidos como presión diastólica. Entonces, si se tiene información temporal de la señal, es sencillo a simple vista hallar de manera aproximada el ritmo cardíaco. O dicho de otro modo, si se conoce el periodo de la señal (tiempo que tarda en recorrer un ciclo de señal), la frecuencia no es más que la inversa del periodo.

Con esta premisa, se plantea diseñar un circuito encargado de detectar los máximos que presenta la señal fotoplestimográfica proporcionada por el sensor y llevar una cuenta de las muestras que existen entre picos. Conociendo el número de muestras entre picos y sabiendo que tenemos una adquisición de datos de 50 muestras por segundos, la frecuencia cardíaca se hallaría tal y como se expresa en la ecuación 1.

Ecuación 1 Frecuencia cardíaca en unidades BPM

$$Frec.cardíaca = \frac{1}{T} = \frac{1}{\frac{nº\ de\ muestras\ entre\ pico}{50\frac{muestras}{s} * 60\frac{s}{min}}}\ [BPM]$$

En términos computacionales o de cálculo a nivel de hardware, una división requiere de una gran cantidad de recursos que engrandece y encarece el sistema. Más adelante se describe una alternativa a este cálculo.

5.2 ENTIDAD MAXIMO

La entidad encargada de detectar los pulsos y medir la cantidad de muestras que hay entre ellos es “maximo”.

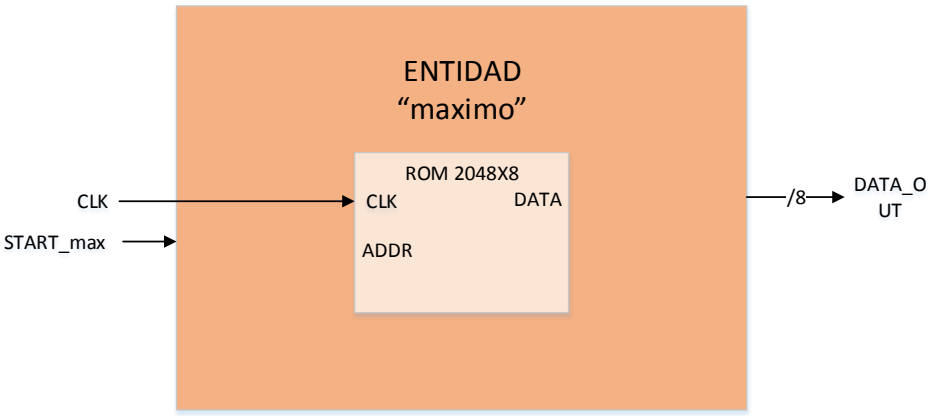


Ilustración 54 Bloque representativo de la entidad "MAXIMO"

“maximo” se ha desarrollado a través de un proceso que está basado en una máquina de estados finitos con dos posibles estados finales, gobernada por una señal de reloj y una señal intermedia de comienzo “start_max”.

La máquina de estados está estructurada en 3 partes. La primera parte solo se ejecuta una vez al iniciar la entidad y detecta el primer máximo. La segunda y la tercera localizan los siguientes máximos de manera alternativa. Además, hallarán la cantidad de muestras que hay entre dichos máximos. Esta cantidad se lo asignará a la señal de salida “DATA_OUT”.

Lo primero, en el estado “INICIO”, se inicializan los valores de las señales a usar. A continuación, se leen 3 bytes del canal infrarrojo situados en la memoria RAM que se ha mencionado en el capítulo anterior. Al tratarse de una memoria, cada lectura de un byte requiere de un ciclo de reloj y para ello se utiliza un contador “ConByte” que cuenta el número de lecturas.

Segundo, se aplica una máscara como se realizó en el apartado 4.3.3para quedarse solamente con los bits de interés. En el caso de la configuración adoptada, serían 15 bits. Con una operación AND se puede resolver.

Después, se compara el valor leído recientemente con el último más grande almacenado. Si es mayor lo remplaza sino es así lo desechamos. Esta operación se repite tantas veces como sea necesario hasta conseguir una serie de datos consecutivos que sean menores que el mayor almacenado. Entonces, llegado a este punto se da por valido el valor almacenado y detectado un máximo.

Para contar el número de muestras que hay entre picos se utiliza contadores. Entonces, una vez que se ha detectado un máximo, un contador (“contador1”) es puesto a cero y comienza a contar hasta encontrar el siguiente máximo (“maximo2”). Cuando se haya dado por valido ese segundo máximo, se almacena la cantidad de muestras contadas en un registro “ContMax1”. De la misma manera ocurre en la búsqueda del siguiente máximo “maximo1” con otro contador “contador2” y almacenando el número de muestras entre picos en “ContMax2”. Este proceso continúa alternando los contadores como se ve en la figura 55. También se puede observar el proceso de la máquina de estados en la figura 56. Por último, si se ha llegado al final de la memoria, se finaliza en un estado final u otro dependiente de si se está calculando “ContMax1” o “ContMax2”.

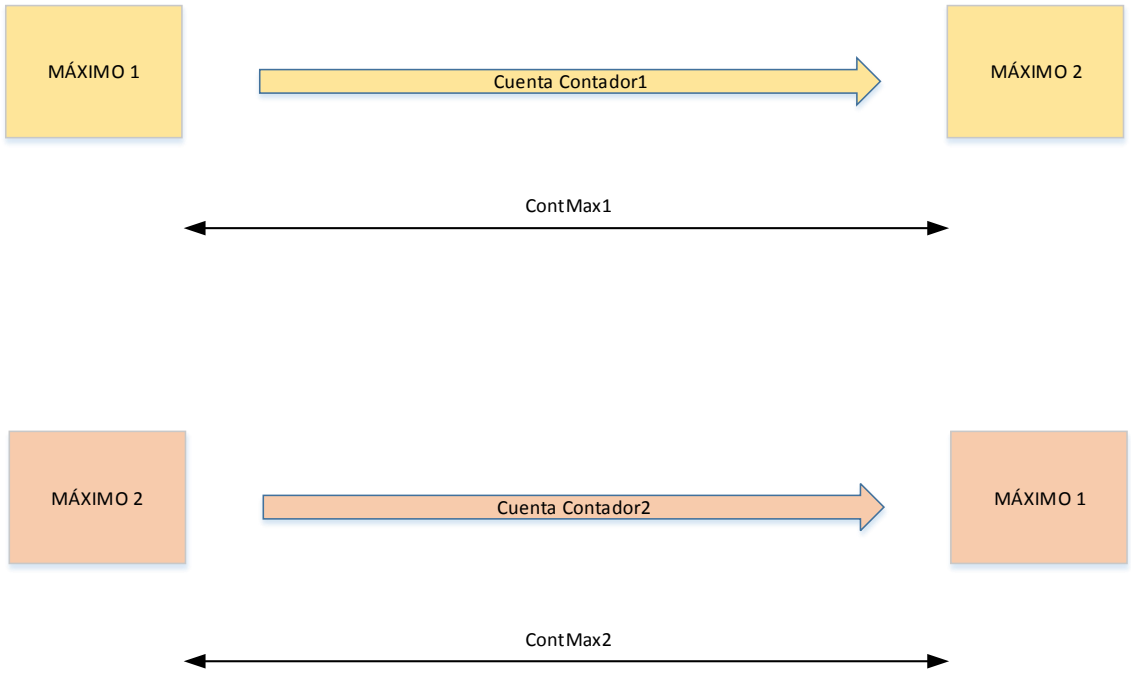


Ilustración 55 Diagrama de bloque que representa el detector de máximos

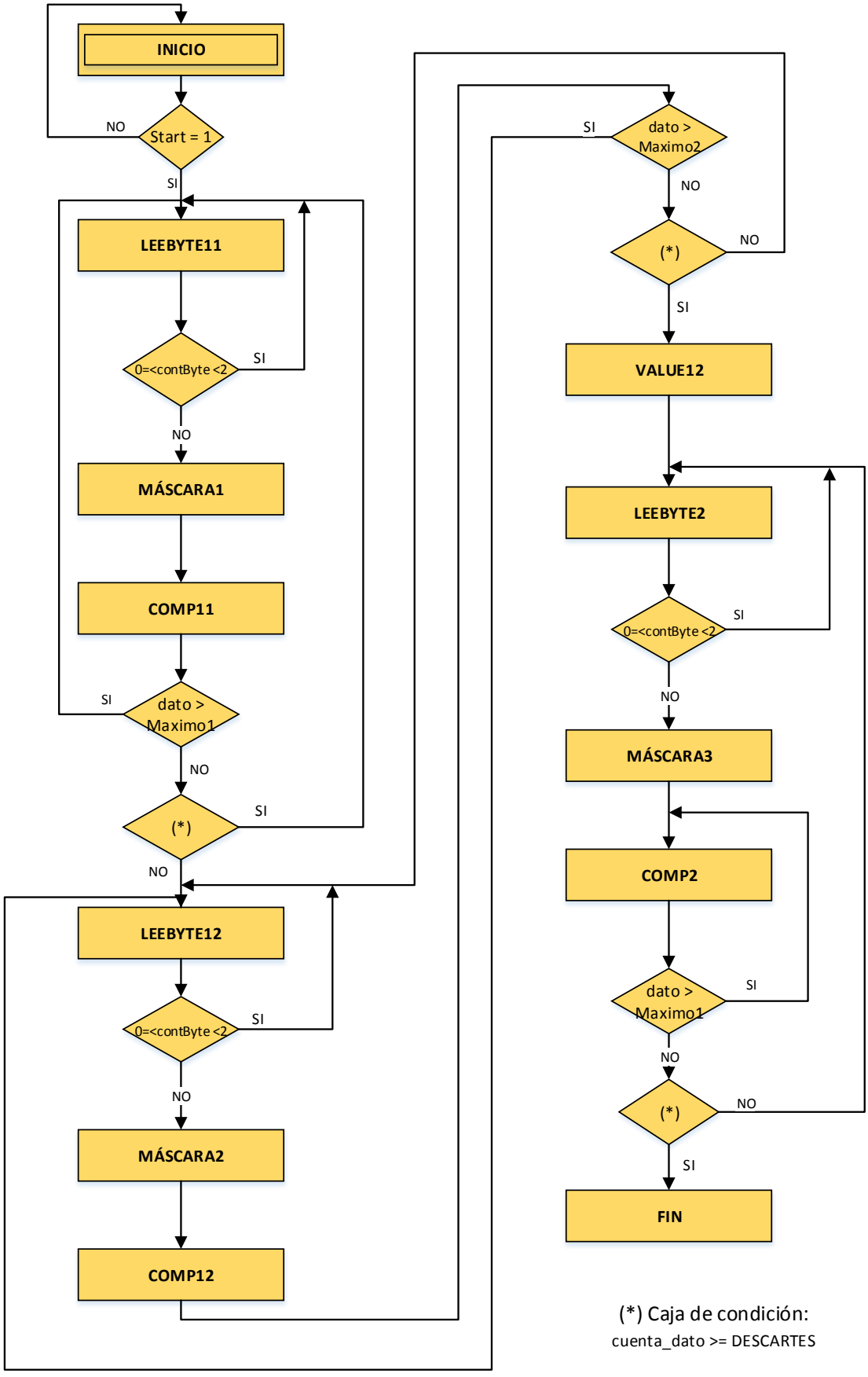


Ilustración 56 Carta ASM de la máquina de estado de la entidad "maximo".

5.3 VERIFICACIÓN

Como todas las entidades que se han diseñado, necesitan una verificación funcional. Igualmente, el método es la simulación.

El testbench generado para esta entidad simula todo el procesado que abarca desde que lee byte a byte de la ROM hasta el estado que devuelve el valor final del número de muestras entre picos. Por otra parte, se analiza con Excel los valores entre picos para corroborar el correcto funcionamiento de la entidad.

Para simular el comportamiento de la memoria se diseña una entidad llamada "ROM2048x8" que contiene los datos registrados de 10 medidas (cada medida de 500 muestras) realizadas con la entidad de adquisición de datos en el capítulo anterior. Los datos están almacenados en un fichero de texto y la entidad ROM2048x8 se ocupa de abrirlo, leerlos y enviárselo a la entidad "maximo" para procesarlos.

En un principio se había diseñado la entidad "máximo" para que descartara solo 3 o 4 muestras, si ya se había encontrado un posible máximo. Pero esto suponía detectar picos que no correspondían con un pulso cardíaco por consiguiente un error en la medida (ver cifras rojas en figura 57).

TABLA DE VALORES DADOS EN SIMULACIÓN										
PRUEBA	**1**	**2(*)**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Pulso 1-2	39	39	40	32	43	49	49	45	14	45
Pulso2-3	40	40	59	35	18	49	53	46	28	41
Pulso 3-4	43	16	45	36	29	44	51	43	28	39
Pulso 4-5	41	20	49	40	39	46	48	43	14	42
Pulso 5-6	44	39	48	41	41	46	41	44	43	45
Pulso 6-7	44	38	44	46	39	42	46	42	46	42
Pulso 7-8	44	36	48	46	40	44	44	44	46	42
Pulso 8-9	45	14	44	46	46	47	49	43	48	48
Pulso 9-10	41	19	46	14	46	48	49	46	49	45
Pulso 10-11	53	37	47	31	52	49	45	47	46	48
Pulso 11-12	47	38		45	51			15	49	47
Pulso 12-13		40		44	35				30	
Pulso 13-14		46		25					16	
Pulso 14-15		47								
MEDIA	44	34	47	37	40	46	48	42	35	44
FREC. CARDIACA	69	90	64	81	75	63	63	72	85	68

Ilustración 57 Falsos picos en la simulación.

Para solucionar este problema y tener la posibilidad de medir señales de hasta 200 bpm, se fue probando con un número más grande de muestras de descartes hasta llegar a una cantidad de 13 muestras y dar por válido el máximo correspondiente. Los resultados pueden observarse en la señal “data_out” en la figura 58.

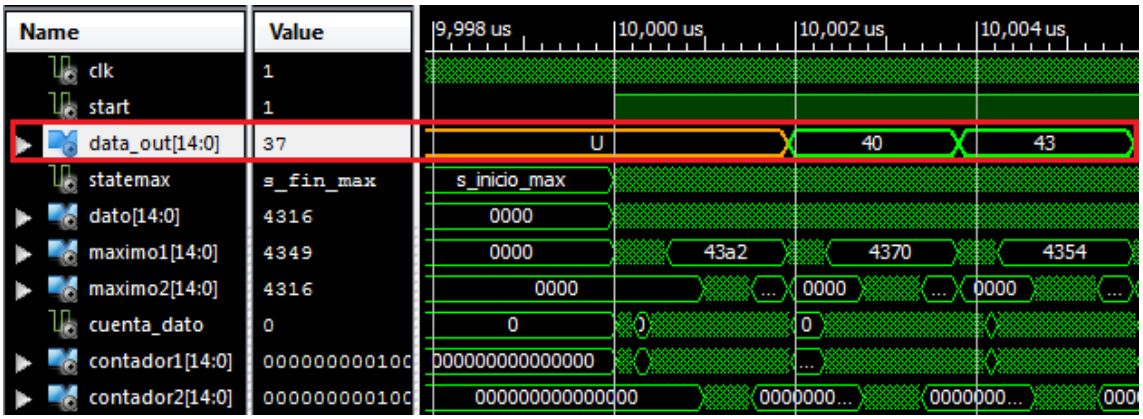


Ilustración 58Resultados de simulación de la entidad "maximo".

5.4 INTEGRACIÓN CON LA ENTIDAD FINAL

La integración de todas las entidades supone modificaciones de entidades ya existentes y la adición de algunos componentes (ver figura 59).

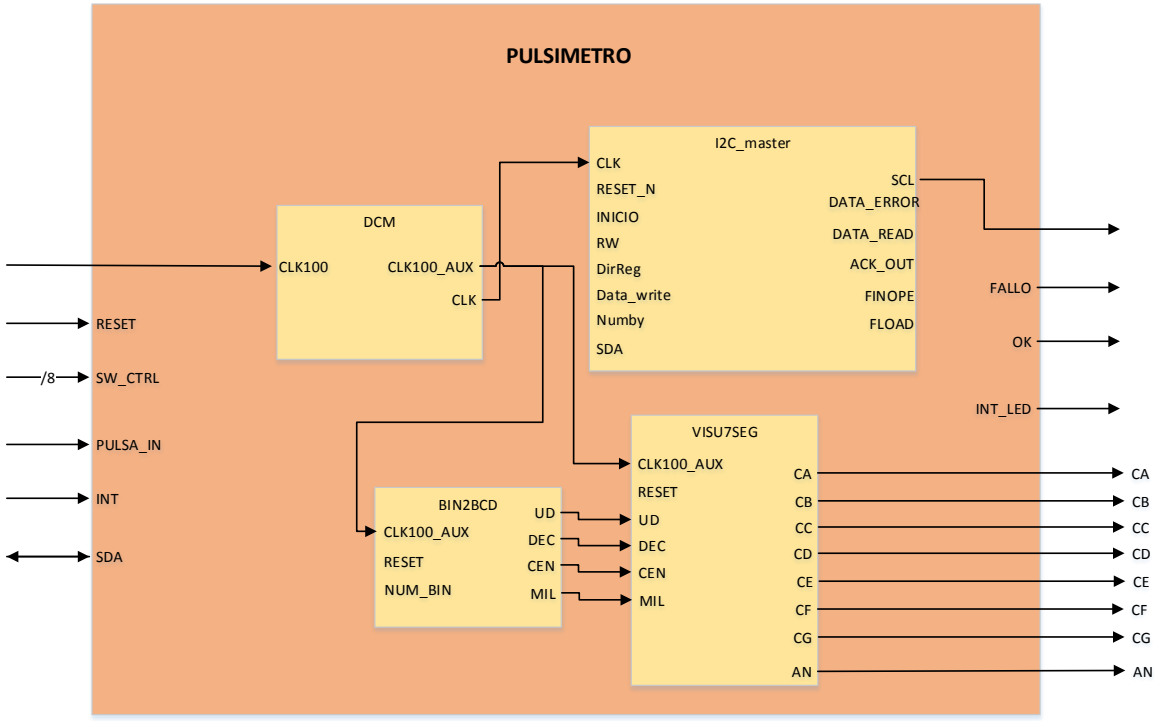


Ilustración 59 Bloque representativo de la entidad principal

La entidad principal llamada “pulsímetro” consta de 4 componentes que son “DCM”, “I2C_master”, “BIN2BCD” y VISU7SEG”, tres procesos que corresponden con la adquisición de datos, el procesado de datos y el operador de media aritmética. Además de una memoria ROM la cual devuelve el valor de la frecuencia cardíaca en función de la distancia entre máximos.

Como en el apartado 5.1 se comentó, la división es un problema para hardware ya que requiere circuitos de gran tamaño. Pero hay soluciones muy útiles para este problema.

Una solución es una memoria ROM que tabule los valores a calcular, es decir, que contenga los valores de la frecuencia cardíaca en función de un puntero de memoria. Este puntero es el valor del registro “ContMax1” o “ContMax2” de la entidad “maximo” que posee la cantidad de muestras que hay desde un máximo a otro. Entonces, esta ROM asigna a su salida el valor aproximado de la frecuencia cardíaca.

A modo de ejemplo, para un valor de entrada “ContMax1” o “ContMax2” de 00101110 (46 en decimal), la memoria asigna a la salida “num_bin” el valor de 001000001 (65 en decimal). O dicho de otro modo, para una cantidad de 46 muestras entre picos se obtiene 65 pulsaciones por minuto. Contrastando con la ecuación 1, para un valor de 46 muestras entre picos se obtendría aproximadamente 65,21 bpm. En definitiva, el error de linealidad supone un error en la medida despreciable.

En la figura 60 se puede observar la estructura de la memoria ROM en el ISE Design.

```
929   -- ROM que contiene el valor de los BPM para un valor de muestras entre pulsos
930   with data_out_max select
931     num_bin(8 downto 0) <=
932           "000000000"  when  "00000000",
933           "110111000"  when  "00000001",
934           "111011100"  when  "00000010",
935           "111101000"  when  "00000011",
936           "011101110"  when  "00000100",
937           "001011000"  when  "00000101",
938           "111110100"  when  "00000110",
939           "110101100"  when  "00000111",
940           "101110111"  when  "00001000",
941           "101001101"  when  "00001001",
942           "100101100"  when  "00001010",
943           "100010000"  when  "00001011",
944           "011111010"  when  "00001100",
945           "011100110"  when  "00001101",
946           "011010110"  when  "00001110",
```

Ilustración 60 Memoria ROM que contiene los datos para el cálculo de la frecuencia cardíaca

Por otra parte, la entidad “BIN2BDC” es un componente que convierte el valor binario de la salida de la memoria ROM “num_bin” en valores BDC para posteriormente enviarlos a la entidad de visualización del display 7 segmentos “visu7seg”. Esta entidad está gobernada por la señal de reloj de 100Mhz y una señal reset.

Una vez que el dato está en formato BCD, por cada flanco de reloj, la entidad “visu7seg” toma los dato en BCD y lo convierte en salidas lógicas (“ca”, “cb”, “cc”, etc) para los cátodos del visualizador 7 segmentos de la placa y así iluminar el o los respectivos segmentos.

Por otro lado, el proceso de la media aritmética nace con la idea de suavizar los resultados. Para ello, cada vez que se encuentra un máximo se hace la media de los 4 últimos o 2 últimos si se está procesado 500 o 250 muestras respectivamente.

El funcionamiento es el siguiente. Los registros “temp_r1”, “temp_r2”, “temp_r3” y “temp_r4” se suman y se alojan en un registro de 10 bits “temp_suma”. Finalmente, para dividir entre cuatro es tan sencillo como desplazar “temp_suma” dos posiciones hacía la izquierda, pero aún más fácil es tomar solo los 8 bits más significativos desechando los dos menos significativos.

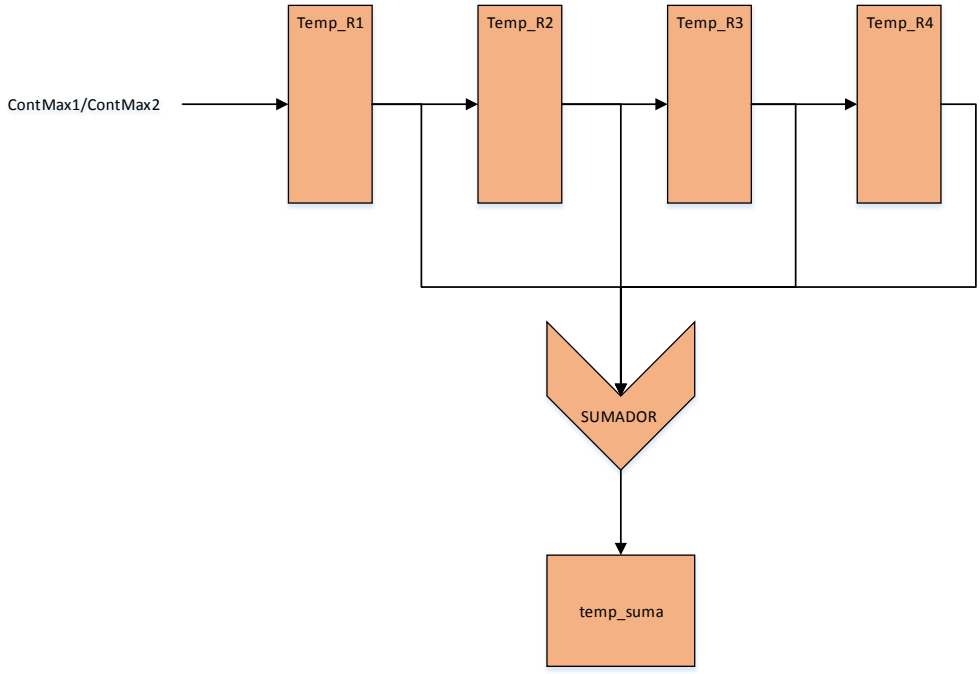


Ilustración 61 Suma aritmética

Por último, la entidad “máximo” expuesta en el subcapítulo 5.2 se ha modificado para que los valores de “ContMax1” o “ContMax2” se almacenen en el registro temporal “temp_r1” donde posteriormente sirve para hallar la media aritmética.

5.4.1 PRUEBAS EXPERIMENTALES

Tras todas las etapas de verificaciones que se han realizado en los apartados anteriores, se ha llegado a un sistema que es capaz de medir el ritmo cardíaco de forma muy aproximada, simplemente situando la yema del dedo en el sensor y accionando el pulsador central de la placa Nexys. En la figura 62 puede verse que tras realizar una medida satisfactoria, el resultado de la medida es de 76 bpm y lo muestra en los visualizadores 7 segmentos. También puede observarse el encendido del led rojo del sensor que demuestra que el MAX30102 está capturando valores. Por último, en la figura 63, se puede ver una medida en directo de la frecuencia cardíaco y su correspondiente visualización.

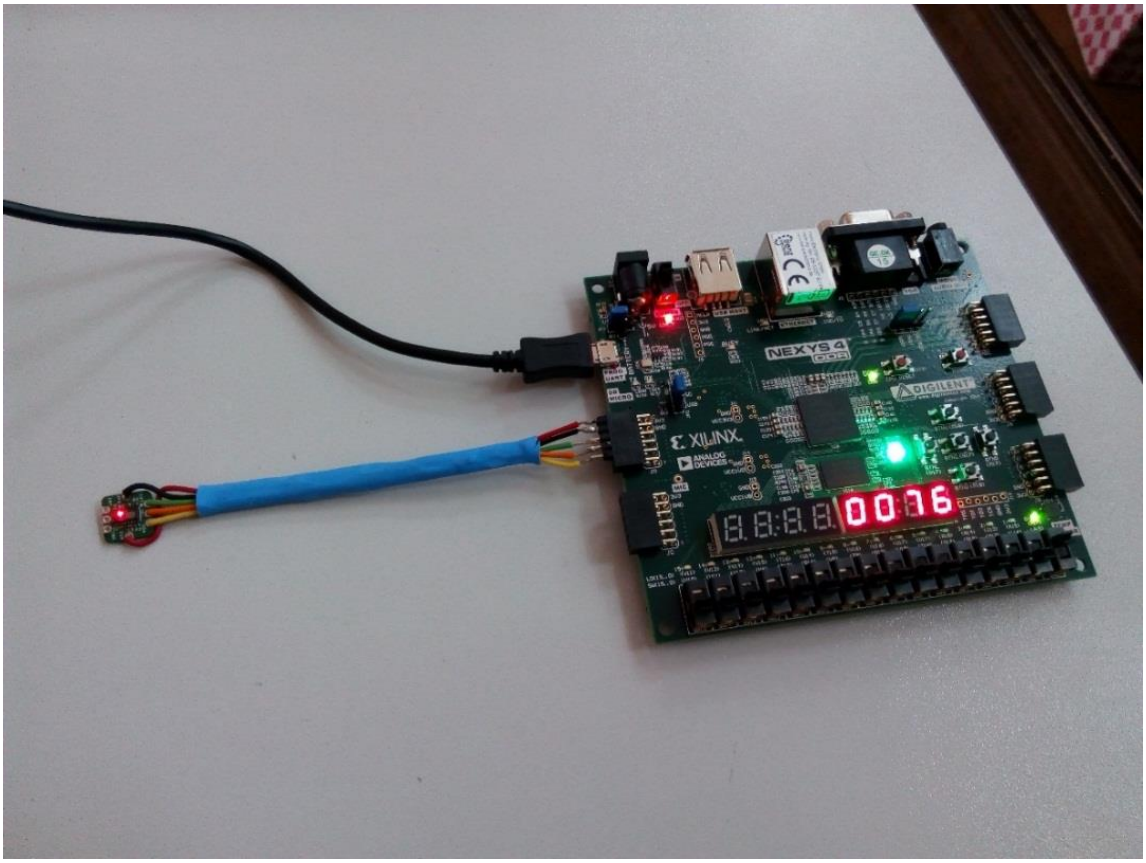


Ilustración 62 Sistema final funcionando correctamente en el laboratorio.

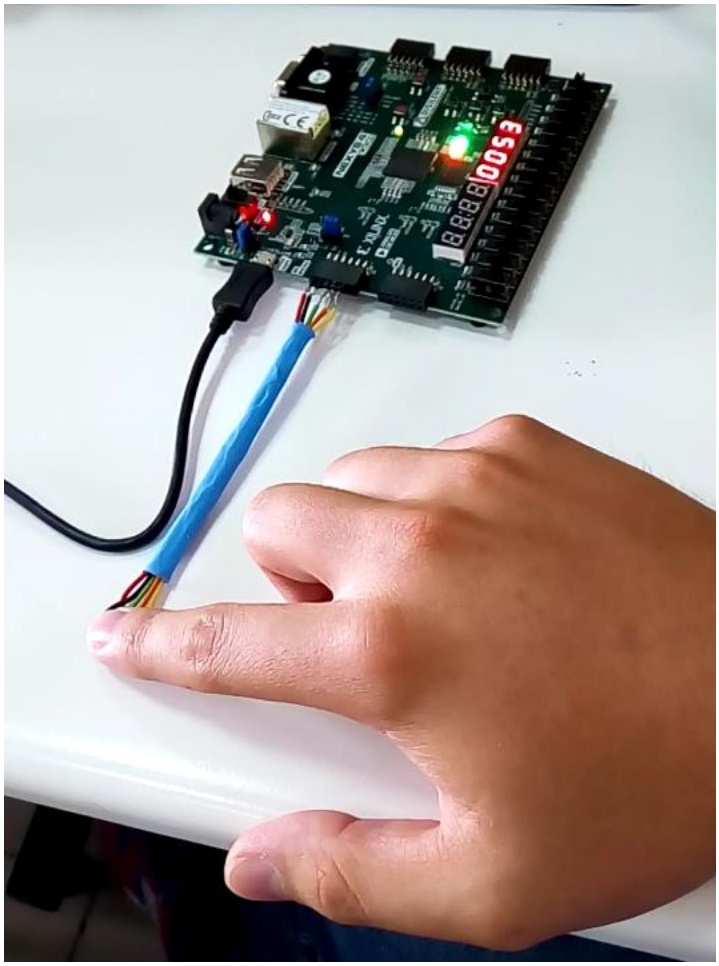


Ilustración 63 Sistema final en funcionamiento.

6 CONCLUSIONES

El objetivo de este proyecto es construir un prototipo de un pulsómetro utilizando un sistema digital basado en una FPGA. El sistema toma los datos proporcionados por un sensor de pulso cardíaco, los procesa y muestra el resultado en un visualizador. Para alcanzar este objetivo se han seguido los siguientes pasos:

Se ha analizado los sensores de pulso cardíaco existentes en el mercado, seleccionándose el sensor de Maxim MAX30102. Este sensor consta de dos fotodiodos y dos fotoreceptores, uno de luz roja y otro de luz infrarroja, con los que se mide la reflexión de la luz sobre la piel del cuerpo humano. A través de la intensidad de esta reflexión se puede obtener el ritmo cardíaco. Este sensor tiene una interfaz I2C para transmitir los datos al exterior.

Se ha seleccionado a Xilinx como el fabricante de los dispositivos FPGA que se van a utilizar, Xilinx ISE como la herramienta de CAD y la placa Nexys4 DDR de Digilent como el soporte concreto sobre el que se van a realizar los diseños.

Se ha diseñado una interfaz I2C en lenguaje VHDL para la comunicación con el sensor de pulso. Se han realizado pruebas para corroborar el correcto funcionamiento de esta interfaz.

Se ha realizado un estudio del sensor para conocer los comandos de configuración y la forma de transmisión de los datos capturados. Se ha realizado un segundo programa que, utilizando la interfaz I2C, realiza la configuración del sensor y realiza una captura de datos.

Para comprobar el funcionamiento de este circuito se ha modificado el diseño para que pueda transmitir los datos capturados a un ordenador. El mecanismo utilizado es la comunicación de la FPGA con el ordenador a través del mismo cable USB que se utiliza para la programación de la placa desde el ordenador. En esta conexión USB se crea un puerto serie virtual que se comunica con la FPGA mediante un circuito FDTI incorporado en la misma placa.

Los datos trasladados al ordenador han sido utilizados con varios propósitos: en primer lugar, han servido para comprobar que los datos recibidos del sensor eran los que se debían recibir, presentando la forma mostrada por el fabricante del sensor. Con estos datos se ha realizado una primera estimación del pulso cardíaco para verificar que las medidas eran correctas. También han servido para analizar el comportamiento del sensor con varios modos de funcionamiento, y así poder seleccionar el modo más adecuado para el circuito desarrollado.

Pero el uso más interesante de estos datos ha sido el de servir de patrones de test para el desarrollo de un circuito que calcule el tiempo entre dos máximos. Como el sensor no proporciona de forma directa la medida el ritmo cardíaco, sino sólo el valor de reflexión de diodo sobre el cuerpo humano, el cálculo del pulso cardíaco ha de hacerse calculando el tiempo que pasa entre dos máximos (o dos mínimos) en el valor de la reflexión. El desarrollo del diseño que calcula este tiempo ha contado con el punto de partida de los datos concretos medidos por el sensor y trasladados al ordenador.

Finalmente se ha montado el sistema completo, que consta de la interfaz I2C con el sensor, el diseño que realiza la configuración y transmisión de datos, un circuito que calcula el tiempo que pasa entre dos máximos y un último circuito que muestra el valor del ritmo cardíaco en los visualizadores de la placa.

Los resultados muestran que el pulsímetro ofrece resultados correctos. El sistema diseñado es capaz de medir un ritmo cardíaco de hasta 200 pulsaciones, siendo válido tanto en aplicaciones médicas como en aplicaciones de salud personal, ocio y deporte.

A partir de los diseños realizados en este proyecto, muchas son las líneas de continuación que se abren. Una continuación es, realizar la medida de la saturación de oxígeno en sangre, la monitorización del pulso cardíaco durante largos periodos de tiempo, monitorización nocturna durante el sueño, la generación de alarmas por detección de pulsos cardíacos peligrosamente altos o bajos, etc.

7 REFERENCIAS

[1] SENSOR MAX30102
https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/MAX30102.html

[2] PLACA MAXREFDES117
https://www.maximintegrated.com/en/design/reference-design-center/system-board/6300.html

[3] PLACA ENTRENADORA NEXYS 4 DDR
https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/start?redirect=1

[4] CÓDIGO VHDL EWIKI
https://eewiki.net/pages/viewpage.action?pageId=10125324