ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
Departamento de Tecnología Electrónica

Tesis doctoral

# Técnicas de Implementación de Circuitos Integrados Digitales CMOS de Alta Velocidad de Operación y Bajo Consumo de Potencia

Realizada por

David Guerrero Martos

Ingeniero en Informática

Dirigida por

Dr. Manuel Jesús Bellido Díaz — Catedrático de Universidad

Dr. Jorge Juan Chico — Prof. Titular de Universidad

Sevilla, febrero de 2012

Escuela Técnica Superior de Ingeniería Informática
Departamento de Tecnología Electrónica

Tesis doctoral

# Técnicas de Implementación de Circuitos Integrados Digitales CMOS de Alta Velocidad de Operación y Bajo Consumo de Potencia

Realizada por

David Guerrero Martos
Ingeniero en Informática

Dirigida por

Dr. Manuel Jesús Bellido Díaz
Catedrático de Universidad

Dr. Jorge Juan Chico
Prof. Titular de Universidad

Sevilla, febrero de 2012

*Este trabajo no hubiese sido posible sin el apoyo de mis compañeros, mentores y amigos. Doy las gracias a todos ellos y a mi familia.*

# Índice general

# Capítulo 1

# Introducción

El *modus vivendi* de muchas culturas actuales depende de aparatos conocidos como *computadoras*. Lo que en esencia caracteriza a estas máquinas es que pueden ser instruidas o *programadas* para realizar una función. A pesar de eso, hay fuentes que las definen basándose en una aplicación determinada, lo que resulta desafortunado. Así, [1] las define como máquinas capaces de *resolver problemas aritméticos y lógicos*, mientras que según [2] son capaces de *procesar datos*, sin tener en cuenta que en nuestro mundo se usan además para muchas otras tareas. Así, muchas computadoras de propósito específico, como sistemas de control de señales de tráfico o consolas de videojuegos, quedan fuera de estas definiciones. Otra aplicación notable de las computadores es la implementación de redes de comunicación: la red de comunicación mundial conocida como Internet está constituida, además de por líneas de comunicación, por un conjunto de nodos interconectados que pueden considerarse como computadores dedicados.

Las disciplinas que estudian el diseño de computadoras se basan en la implementación de un tipo de funciones denominadas *booleanas* (o de forma más específica, *funciones booleanas binarias*). Existen muchas formas de implementar dichas funciones, algunas tan pintorescas como el uso de tecnologías mecánicas. No obstante, la más utilizada hoy en día es la tecnología electrónica que produce los llamados circuitos y sistemas digitales y, entre ellos, las computadoras digitales electrónicas. Los sistemas digitales

actuales se caracterizan por su alta velocidad de operación y su elevada miniaturización (podemos encontrar del orden de miles de millones de componentes en un solo circuito integrado). La evolución de esta tecnología a lo largo de las últimas décadas ha permitido desarrollar los sistemas digitales y las comunicaciones de forma espectacular y, a medio plazo, no parece que vaya a ser sustituida por otra tecnología. El diseño de circuitos y sistemas digitales se enfrenta continuamente con nuevos retos a la hora de mejorar sus prestaciones, sobre todo a la hora de aumentar su velocidad de operación y reducir su consumo de energía. Esta tesis aborda estos dos problemas estudiando y proponiendo mejoras en los dispositivos digitales actuales de cara a aumentar su velocidad de operación y reducir su consumo. En concreto nos centraremos en sistemas digitales construidos en tecnología *MOS* (*Metal-Oxide-Semiconductor*, Semiconductor-Óxido-Metal), que es la más ampliamente utilizada. Así mismo, se estudian los esquemas de sincronización empleados en los circuitos digitales y se plantean alternativas que permitan mejorar su rendimiento.

## 1.1. Introducción al diseño de circuitos digitales

El diseño de sistemas digitales persigue, entre otros, los siguientes objetivos:

- Optimizar la velocidad, es decir, reducir el tiempo que el sistema tarda en dar una respuesta válida.
- Reducir el espacio físico ocupado por el sistema.
- Reducir el coste de producción.
- Reducir el consumo de energía del circuito.

Hoy en día el área ocupada no resulta crítico en la mayoría de las aplicaciones gracias a la disminución exponencial del tamaño de los componentes electrónicos. Por otro lado se está dedicando un gran esfuerzo en la mejora

de la velocidad y en la eficiencia energética. Esto último es especialmente importante en equipos portables alimentados por baterías de cara a aumentar su autonomía. Suelen distinguirse dos tipos de consumo:

- Consumo *dinámico*: se define como el que se produce cuando el sistema conmuta, es decir, cuando cambia su estado lógico.

- Consumo *estático*: se define como el que se produce cuando el sistema se encuentra en un estado lógico estable.

Para realizar funciones booleanas, los circuitos electrónicos usan componentes denominados *puertas lógicas*, que implementan operadores booleanos sencillos. Además de la implementación de las funciones booleanas, los diseñadores deben ocuparse de aspectos de sincronización cuando implementan *circuitos secuenciales*. Este tipo de circuitos van calculando funciones booleanas de forma iterativa, en ocasiones tomando como entrada de dichas funciones valores previamente calculados. El subcomponente encargado que implementa las funciones booleanas se denomina *circuito combinacional*. El diseñador debe ocuparse de que el cómputo se lleve a cabo únicamente cuando el dato de entrada haya sido ya calculado. En la mayoría de los circuitos actuales esta temporización se lleva a cabo empleando señales de control periódicas llamadas *señales de reloj*. Los sistemas dotados de estas señales se denominan *síncronos*. El esquema general de un circuito secuencial síncrono (CSS) se muestra en la figura 1.1. Incluye los siguientes bloques:

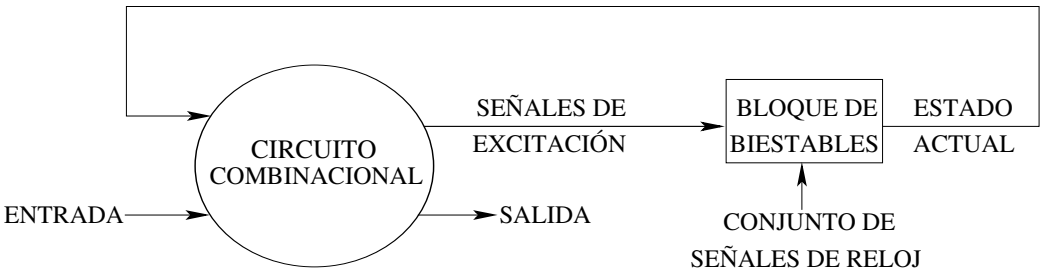



Figura 1.1: Circuito secuencial síncrono genérico.

- Un conjunto de elementos de memoria (biestables) que almacenan el estado del sistema. Para que operen correctamente es necesario que sus entradas (señales de excitación) cumplan determinadas restricciones temporales. Para simplificar la tarea de diseño suelen usarse biestables tipo D, con lo que las señales de excitación coinciden con el próximo estado del biestable.

- Un circuito combinacional que calcula las señales de salida y las señales de excitación de los biestables en función del estado actual y de las entradas. Este circuito está construido a partir de puertas lógicas. Un cambio en su entrada puede provocar, pasado un retraso mayor que cero, un cambio en la salida. Para garantizar el correcto funcionamiento del sistema es necesario calcular una cota superior para el tiempo que tarda el circuito en mostrar la salida correspondiente a la nueva entrada (retraso máximo). También es necesario calcular una cota inferior para el tiempo que seguirá mostrando la salida correspondiente a la entrada previa cuando dicha entrada haya cambiado, lo que se denomina *retraso de contaminación* [3].

- Un conjunto de señales de reloj que controlan los instantes en los que cambian de estado los biestables. Normalmente son señales periódicas.

- Un conjunto de líneas conductoras que propagan las señales de control y la información entre los distintos componentes.

Hasta hace pocos años el retraso de las líneas de interconexión era despreciable frente al de los componentes secuenciales y combinacionales, lo que facilitaba la labor de diseño de CSS. Irónicamente, las continuas mejoras en la tecnología de fabricación han complicado la labor del diseñador: las puertas lógicas son cada vez más pequeñas, lo que permite una alta densidad de integración, pero integrar más componentes en cada chip implica, además de aumentar su consumo, incrementar la longitud de las líneas de interconexión en relación con el tamaño y capacidad de carga de las puertas que alimentan dichas líneas. Esto hace que el retraso de estas líneas sea comparable al de los elementos que interconectan. Cada vez se estrechan más

los intervalos de tiempo que tienen las señales para operar adecuadamente, con lo que la dimensión temporal ha ido ganando en importancia en las cuestiones de diseño. En el apartado siguiente veremos con más detalle los componentes empleados para implementar la parte combinacional de los CSS. La importancia de los aspectos de sincronización en los circuitos secuenciales se detallará en el apartado 1.3.

## 1.2. Puertas lógicas CMOS

Esta tesis se encuadra en los sistemas digitales construidos en tecnología MOS. Más concretamente nos centraremos en la tecnología *CMOS* (*Complementary MOS*) que es la más ampliamente utilizada en la actualidad. Esto es debido a las excelentes prestaciones de las puertas CMOS fruto de la investigación en entornos académicos e industriales en materia de diseño y de procesos de fabricación. Algunas de las bondades de esta tecnología se listan a continuación:

- Reducido espacio: la disminución exponencial del tamaño de los componentes electrónicos nos permite integrar en una sola pastilla de silicio complejos sistemas que hace poco resultaban muy voluminosos y difíciles de transportar.

- Bajo coste: uno de los efectos de la continua disminución del tamaño de los componentes electrónicos es que permiten aumentar el número de chips que se incluyen en cada oblea fabricada, lo que ayuda a reducir el coste por chip. Hoy en día, el reducido coste de los sistemas electrónicos ha permitido integrarlos en cualquier electrodoméstico o artículo de consumo.

- Reducido consumo: hoy en día la tecnología CMOS hace posible el uso de ordenadores portátiles, teléfonos móviles o minúsculos reproductores digitales de música con horas de autonomía.

- Alta velocidad: las mejoras en tecnología CMOS han permitido aumentar la velocidad de cómputo en varios órdenes de magnitud en los

últimos 20 años. Aunque en la actualidad existen otras tecnologías más rápidas, estas no pueden competir en coste y consumo con la CMOS y se relegan a aplicaciones muy específicas.

Explicar la causa de estas excelentes prestaciones requiere estudiar los transistores MOS que componen las puertas lógicas. A continuación describiremos brevemente dichos transistores y como se interconectan para implementar puertas lógicas.

### 1.2.1. Estructura de los transistores MOS

En la actualidad la mayoría de los circuitos digitales emplean transistores MOS por sus buenas características eléctricas y su bajo coste. Se trata de un dispositivo electrónico pasivo cuyo funcionamiento se basa en un fenómeno denominado *efecto campo*. Aunque tiene importantes aplicaciones analógicas tales como amplificación de señales, nosotros nos centraremos en su uso en circuitos digitales, en los que funciona como un conmutador. La figura 1.2 muestra el corte transversal de un transistor MOS. Como puede verse, está dotado de cuatro terminales:

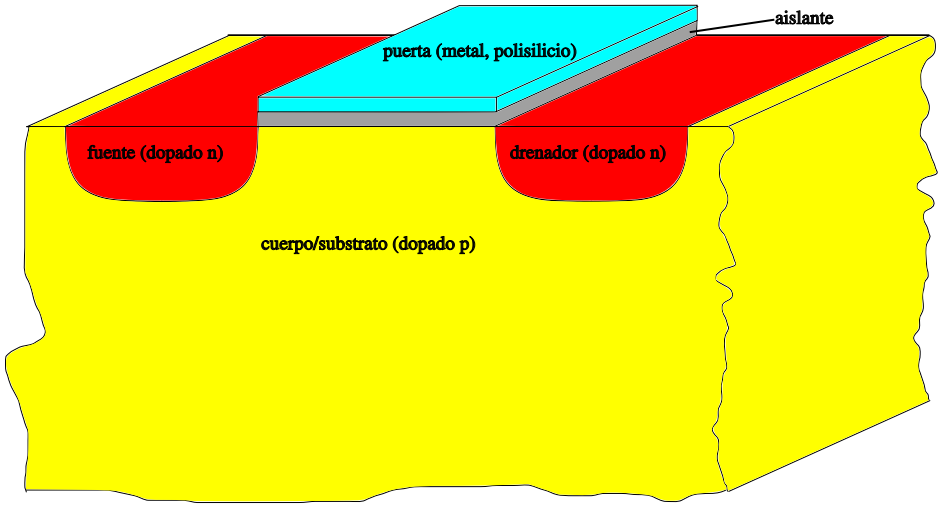



Figura 1.2: Corte transversal de un transistor *bulk*-NMOS
(se han omitido los contactos a los terminales).

- puerta (*gate*): Es una capa de material altamente conductor, originalmente un metal. Actualmente se usa más a menudo polisilicio, aunque la mayor conductividad de los metales hace que se este volviendo a usarse en circuitos de altas prestaciones [4, 5].

- cuerpo (*body*): Se trata de una capa de material semiconductor (originalmente silicio) paralela a la puerta. En la gran mayoría de procesos de fabricación se dopa con impurezas que alteran la densidad de sus portadores de carga. Una fina capa de material dieléctrico separa el terminal *body* de la puerta, de modo que, idealmente, no circula corriente entre ambos. Tradicionalmente este aislante es óxido de silicio, pero actualmente hay tecnologías de integración tan pequeñas que requieren materiales con una constante dieléctrica más alta [4, 6, 7].

- drenador (*drain*) y fuente (*source*): Estas terminales son anexas al terminal *body* y están separadas entre sí por este. Están formadas por el mismo material semiconductor del terminal *body*, pero tienen un dopado opuesto al de este (P si el terminal *body* tiene un dopado tipo N y N si tiene un dopado tipo P). En la gran mayoría de aplicaciones los terminales drenador y fuente son estructuralmente idénticos y se diferencian únicamente por el nivel de tensión que se les aplica.

Los transistores MOS se clasifican según el tipo de impureza que dopa sus terminales drenador/fuente. Así, los transistores MOS tipo N (transistores NMOS) presentan impurezas tipo N en sus terminales drenador/fuente e impurezas tipo P en el terminal *body*, mientras que los transistores MOS tipo P (transistores PMOS) presentan un dopado tipo P en sus terminales drenador/fuente y tipo N en el terminal *body*. Así, el transistor mostrado en la figura 1.2 es de tipo NMOS. En concreto se trata de un transistor NMOS construido en tecnología *bulk* [8], lo que significa que el propio sustrato sobre el que se construyen los transistores constituye el terminal *body*. Esto implica que, en principio, todos los transistores comparten ese terminal. Las primeras tecnologías de fabricación de transistores MOS eran de tipo *bulk* de modo que, al establecerse un dopado común para todo el

*body*/sustrato, solo permitían un tipo de transistor (P o N) en cada chip. Con el tiempo surgieron tecnologías *bulk-CMOS* (*Complementary MOS*), esto es, que permitían implementar en el mismo chip ambos tipos de transistores. En estas tecnologías los transistores de un tipo se construyen directamente sobre el sustrato, mientras que los del tipo complementario se construyen sobre los llamados *pozos* (*wells*). Estos pozos son partes de la superficie de la oblea con un dopado opuesto al del sustrato. La figura 1.3 muestra transistores complementarios construidos en un proceso *bulk*-CMOS de pozo N. Hoy en día, la mayor parte de los circuitos electrónicos se fabrican utilizando tecnología *bulk*-CMOS al ser muy madura y económica.

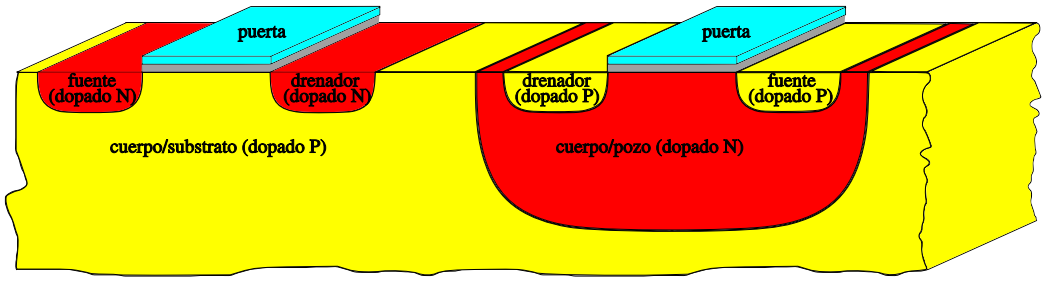



Figura 1.3: Corte transversal de transistores complementarios construidos en un proceso *bulk*-CMOS de pozo N (se han omitido los contactos a los terminales).

Las figuras 1.4 y 1.5 muestran representaciones esquemáticas de los transistores NMOS y PMOS respectivamente. Nótese que hay representaciones en las que se omite el terminal *body*. Normalmente, en los transistores NMOS este terminal se conecta a la tensión más baja del circuito (la tensión de tierra) mientras que en los transistores PMOS se polariza a la tensión más alta (la tensión de alimentación). De esta forma las uniones PN correspondientes a los contactos drenador/*body* y fuente/*body* están siempre polarizadas inversamente. En principio esto puede hacernos pensar que la intensidad que atraviesa esas uniones debe ser nula, sin embargo esto depende de la tensión que se aplique a la puerta. Para ilustrarlo obsérvese la distribución de cargas en el transistor NMOS de la figura 1.6 (el caso PMOS es análogo): una diferencia de potencial positiva entre la puerta y el terminal *body* crea una zona de vaciamiento sobre la superficie de este


Figura 1.4: Representaciones esquemáticas de un transistor NMOS incluyendo el terminal *body* (izquierda) u omitiéndolo (derecha).


Figura 1.5: Representaciones esquemáticas de un transistor PMOS incluyendo el terminal *body* (izquierda) u omitiéndolo (derecha).

último. Si dicho potencial se aumenta hasta llegar a la tensión umbral $V_t$, los electrones se atraen hasta la superficie creando una zona de inversión (el canal) que permite a la corriente fluir de drenador a fuente. De esta forma un transistor MOS puede modelarse como un interruptor controlado por la tensión de puerta, lo que permite utilizarlo para implementar puertas lógicas como veremos a continuación.

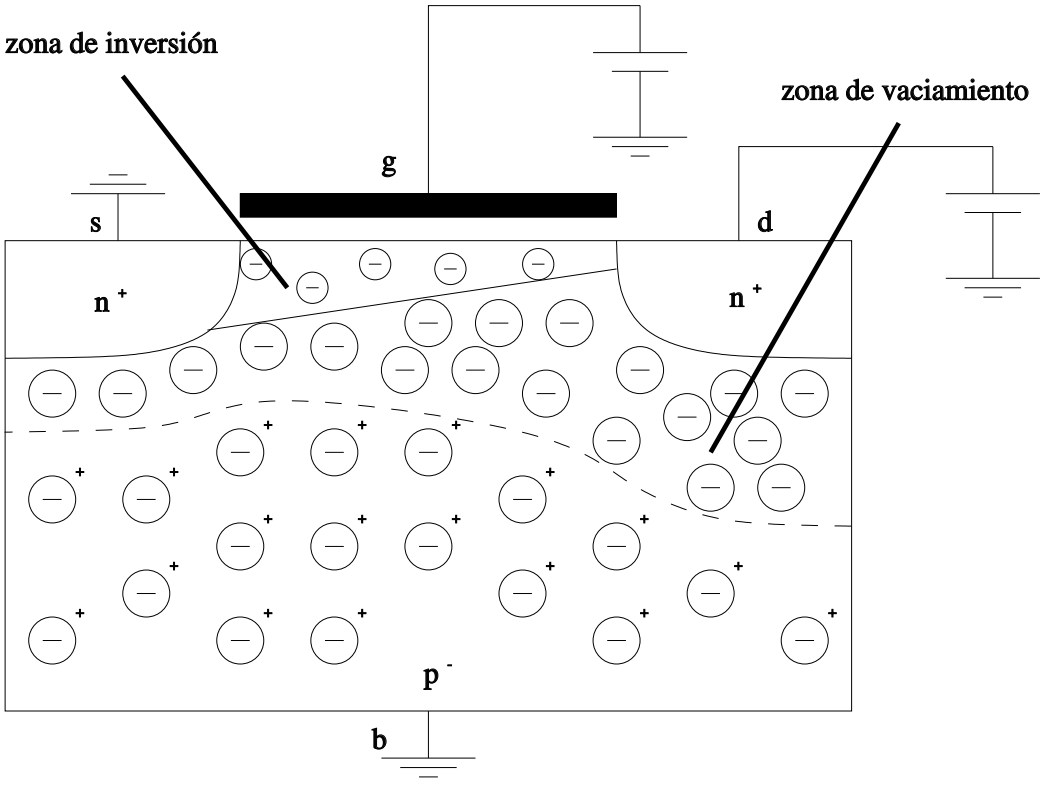



Figura 1.6: Formación del canal en un transistor NMOS.

### 1.2.2. Estructura de las puertas MOS complementarias

La aparición de tecnologías de fabricación CMOS supuso una auténtica revolución. Estas tecnologías permitían diseñar un tipo de puertas denominadas *complementarias* que presentan una tolerancia al ruido y un consumo extremadamente buenos, especialmente cuando no conmutan sus entradas. Debe tenerse en cuenta, no obstante, que no existe un estilo de diseño óptimo para cualquier entorno pues cada uno tiene sus ventajas e inconvenientes [9]. Por ejemplo, las puertas pseudo-NMOS requieren menos área que las CMOS complementarias, pero tienen menor margen de ruido y mayor consumo; las puertas dinámicas son más rápidas, pero son muy sensibles al ruido y su uso requiere un importante esfuerzo de diseño; la lógica de paso puede reducir el área en determinadas aplicaciones, pero suele requerir entradas en doble raíl y/o lógica adicional que restaure los

valores lógicos [10]. Los requerimientos del sistema o subsistema (tiempo y recursos disponibles para el diseño, tolerancia al ruido, área, velocidad, consumo, etc.) establecerán cual es la mejor alternativa. Sin perjuicio de lo anterior, las puertas CMOS complementarias son las más ampliamente usadas [9, 11] debido en gran medida a su robustez y a la existencia de herramientas de diseño que automatizan y reducen enormemente el tiempo y esfuerzo de diseño de circuitos que emplean este tipo de puertas. Por ello esta tesis se ha centrado en el diseño de puertas MOS complementarias.

Tal y como se ilustra en la figura 1.7, una puerta CMOS complementaria está compuesta por dos redes de transistores: la red *PUN* (*pull-up network*) y la red *PDN* (*pull-down network*). La primera se encarga de conectar el nodo de salida con la tensión de alimentación cuando dicho nodo deba tener un valor lógico alto. De forma complementaria, la red PDN se encarga de conectar a tierra la salida cuando esta deba tener un valor lógico bajo. La topología de estas redes se elige de forma que, para cualquier entrada con niveles lógicos válidos, una y solo una de ambas redes está conduciendo en estado estacionario.

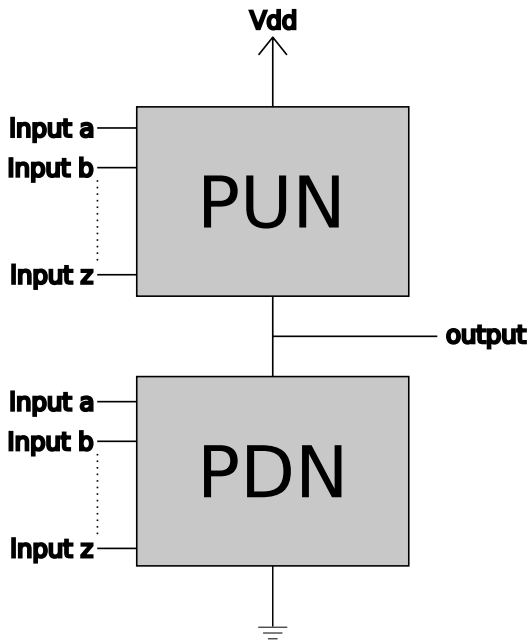



Figura 1.7: Estructura de una puerta complementaria CMOS.

A la hora de diseñar las redes PUN y PDN debe tenerse en cuenta lo siguiente [9]:

- Como se mencionó en el apartado 1.2.1, los transistores MOS funcionan como interruptores. Un transistor NMOS se activa cuando su puerta tiene una tensión correspondiente al nivel lógico alto, mientras que un PMOS se activa si su puerta se encuentra a un nivel lógico bajo.

- La red PDN debe usar exclusivamente transistores NMOS mientras que la red PUN debe usar únicamente transistores PMOS. De no ser así, el *efecto sustrato* produciría degradación de los niveles lógicos. Este efecto se explicará en el apartado 2.1.1.

- Debido a la restricción anterior, las puertas complementarias CMOS solo pueden implementar *funciones inversoras* (NAND, NOR, etc.), de modo que una única transición en una entrada solo puede provocar transiciones de sentido opuesto en la salida. La implementación de funciones no inversoras requiere por tanto dos o más niveles de puertas.

La topología de las redes para puertas NAND y NOR es muy sencilla: La red PUN de una NAND de N entradas está constituida por N transistores en paralelo que conectan la salida con la tensión de alimentación, mientras que la red PDN la forman N transistores en serie que conectan la tierra con el nodo de salida. La puerta de cada transistor de la red PUN se conecta a una entrada distinta, y lo mismo ocurre con la red PDN. De forma complementaria, la red PUN de una puerta NOR está constituida por transistores en serie y la red PDN por transistores en paralelo. A modo de ejemplo, el esquemático de una puerta NAND de dos entradas implementada en lógica CMOS complementaria se ilustra en la figura 1.8. El diseño complementario CMOS permite implementar funciones más complejas que simples operaciones NAND/NOR en una sola etapa, aunque deben tratarse siempre de funciones inversoras. Las puertas que implementan estas funciones suelen denominarse *puertas complejas*.

Las puertas diseñadas de acuerdo con la metodología CMOS complementaria presentan las siguientes ventajas:

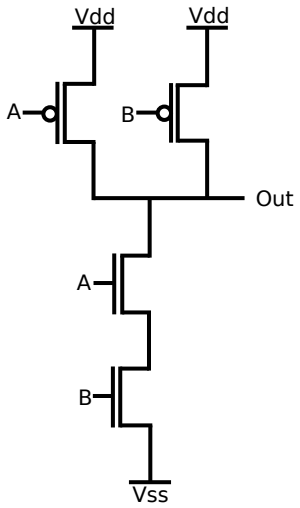



Figura 1.8: Esquemático de una puerta NAND de dos entradas implementada en lógica CMOS complementaria.

- El valor de la tensión umbral puede ajustarse en el proceso de fabricación lo que, al contrario de lo que sucede en otras tecnologías electrónicas, permite elegir la tensión de alimentación dentro de un rango muy amplio: puede elegirse una tensión de alimentación alta para mejorar la tolerancia a ruido, pero también es posible diseñar circuitos con tensiones de alimentación muy bajas para reducir su consumo.

- Dado que, en estado estacionario, solo una de las redes (PUN o PDN) de cada puerta está activa, no existe dentro de la puerta un camino que permita el flujo de corriente de alimentación a tierra (salvo por las corrientes de fuga). Esto, unido a la alta impedancia de la puerta de los transistores MOS, hace que el consumo estático de circuitos lógicos construidos en esta tecnología sea muy reducido.

- Cuando un cambio en la entrada de una puerta provoca un cambio en la salida, las redes PUN y PDN solo están activas simultáneamente durante un periodo muy reducido. Transcurrido ese pequeño lapso no hay contienda entre ambas redes y la red activa puede conmutar la salida de forma muy rápida.

- El reducido consumo estático de las puertas CMOS hace que generen mucho menos calor que otras tecnologías electrónicas, lo que permite integrar multitud de componentes en el mismo chip. Esto es beneficioso porque reduce los costes de producción y porque la comunicación entre componentes dentro del mismo chip es más rápida.

A pesar de todas estas bondades, ciertos fenómenos como acoplamientos capacitivos y el ya mencionado efecto sustrato limitan las prestaciones de estas puertas. En particular, una puerta CMOS no puede tener un número arbitrariamente alto de entradas como veremos en más detalle en el apartado 2.1.

## 1.3. Esquemas de sincronización

Como se indicó en el apartado 1.1, los circuitos secuenciales calculan iterativamente funciones booleanas, a menudo tomando como entrada de dichas funciones valores previamente calculados. Esta característica se denomina *realimentación*. Para que el sistema funcione correctamente el diseñador debe resolver el problema de la *sincronización*, es decir, debe asegurarse de que cada cómputo se lleve a cabo únicamente cuando el dato de entrada haya sido ya calculado. La forma más habitual de implementar circuitos secuenciales es usar señales que sincronicen los elementos de cómputo (señales de reloj). Es más, algunos tipos de puerta, como las puertas dinámicas, son inherentemente síncronas, aunque en este trabajo nos centraremos únicamente en esquemas en los que las señales de reloj controlan únicamente elementos de memoria. Normalmente el bloque de memoria del CSS se construye a partir de biestables. Estos son elementos de memoria básicos que en condiciones normales de funcionamiento pueden encontrarse en dos estados eléctricos estables permitiendo, por tanto, almacenar un bit de información. Como es lógico, los estados de los biestables destinados a almacenar bits de información se etiquetan 0 y 1. La forma en la que se sincronizan los elementos de un circuito secuencial síncrono está estrechamente vinculada al tipo de biestables que utiliza [12], por lo que

en el apartado siguiente estudiaremos dichos elementos. En particular nos centraremos en biestables tipo D, que son los utilizados entre las etapas de los sistemas *segmentados* [13] para almacenar resultados intermedios. En el apartado 1.3.2 veremos distintas formas de sincronizar el funcionamiento de los biestables mientras que en el apartado 4.4.2 se describirán los sistemas segmentados.

### 1.3.1. Elementos de memoria

Desde el punto de vista externo, un biestable D dispone de las siguientes señales de entrada y salida:

- Una salida de estado $q$: El valor de esta salida indica el estado en el que se encuentra el biestable, es decir, el valor del bit que el biestable tiene almacenado. A menudo los biestables disponen además de otra salida con el valor complementario a q.

- Una entrada de excitación $D$: Cuando el biestable cambie de estado lo hará al indicado por esta entrada. Si se utiliza el biestable para almacenar información puede interpretarse que, cuando el biestable cambia de estado, el valor de esta entrada es escrito en el biestable.

- Una señal que controla el instante en el que el biestable cambia de estado, es decir, cuando es escrito. En los CSS suelen conectarse a una fuente de señal cuadrada común a todo el sistema (la señal de reloj), etiquetada usualmente con la letra '$\Phi$'.

La señal de reloj es la que establece cuándo se van a producir los cambios de estado del biestable. Podemos clasificar los biestables en dos grupos atendiendo a los instantes en los que pueden producirse estos cambios:

- Biestables D sensibles a nivel o *latches*: Estos biestables solo pueden cambiar de estado cuando su señal de control de carga (load) tiene un valor lógico determinado denominado *valor activo*. En los CSS esta señal de control suele conectarse a una señal de reloj. De esta forma, cuando el reloj tiene el valor activo el estado del biestable se

hace igual al de la entrada D y cambia con esta con un determinado retraso denominado *retraso de propagación*. En tal caso se dice que el biestable es *transparente*. Salvo que se indique lo contrario, en adelante consideraremos que el nivel activo del biestable será el nivel alto (las consideraciones para el nivel bajo son idénticas). En estos biestables se definen los instantes de captura como aquellos en los que la señal de reloj pasa del valor activo al no activo. La figura 1.9a muestra este comportamiento para un *latch* tipo D.

- Biestables D sensibles a flanco o *flip-flops*: Estos biestables solo pueden cambiar de estado cuando se produce un cambio determinado (flanco activo) en la señal de reloj. En estos biestables la entrada de reloj se etiqueta con el símbolo '^', y los instantes de captura se definen como aquellos en los que se producen flancos activos en la señal de reloj. En estos flancos el estado del biestable se hace igual al de la entrada D. De nuevo, entre la ocurrencia del flanco activo y el cambio de estado transcurre un retraso de propagación. Dependiendo del biestable, los flancos activos pueden ser los de subida (cambio de 0 a 1), de bajada (cambio de 1 a 0) o ambos. Salvo que se indique lo contrario, en adelante consideraremos que los flancos activos son los de subida.

En la figura 1.9 se compara el comportamiento de biestables D sensibles a nivel y sensibles a flanco. Se observa que el *flip-flop* permite aislar mejor la salida de la entrada que el *latch*. En el caso del *latch* se observa que tiene un comportamiento combinacional durante el nivel activo del reloj. Esta propiedad de transparencia va a establecer las particularidades de temporización que deben considerarse cuando se diseña con uno u otro tipo de dispositivo. A parte de los tiempos de propagación, hay que tener en cuenta parámetros temporales típicos de cada biestable: Para asegurar que el biestable opera de forma correcta en cada instante de captura, la excitación de entrada debe permanecer a un valor lógico válido y fijo durante un intervalo de tiempo. A este intervalo se le denomina *ventana de metaestabilidad* [14]. El *tiempo de setup* es una cota del tiempo que debe transcurrir desde el inicio de la ventana hasta el instante de captura. El *tiempo de hold* es una cota del

tiempo que debe transcurrir desde el instante de captura hasta el fin de la ventana. Estos parámetros se ilustran en la figura 1.9.

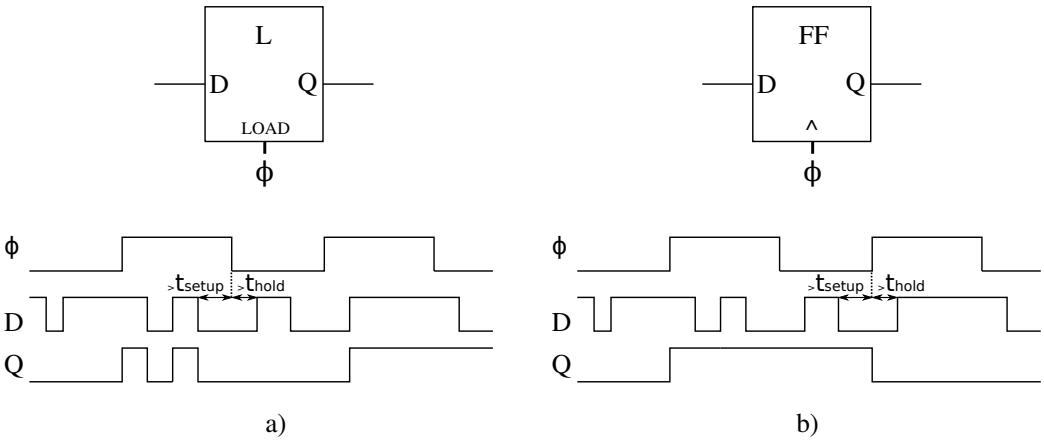



Figura 1.9: Comportamiento de biestables: a) *latch*, b) *flip-flop.*

### 1.3.2. Esquemas de sincronización existentes

Según el número de señales de reloj y la forma en la que controlan los distintos elementos de memoria, podemos distinguir varios esquemas de sincronización o temporización. Algunos de ellos son los siguientes:

- Esquema de temporización de una fase de reloj basado en *latches* controlados por pulso: En este esquema se tienen tantos *latches* como señales de estado tenga el CSS. Todos los *latches* están controlados por la misma señal de reloj, y cada uno genera una señal de estado y es alimentado por una señal de próximo estado distinta. Un pulso activo en la señal de reloj hace que se actualice el bloque de registros.

- Esquema de temporización de una fase de reloj basado en *flip-flops*: Este esquema es similar al anterior salvo que usa *flip-flops* en lugar de *latches*. En cada flanco activo de la señal de reloj el estado del bloque de registros se actualiza.

- Esquema de temporización *máster-slave*: En este esquema se usa un par de *latches* por cada variable de estado, el *latch máster* y el *latch slave*. La salida del primero se conecta a la señal de excitación del segundo. Los *latches slave* generan las señales de estado, y las señales de próximo estado alimentan las señales de excitación de los *latches máster*. Los *latches máster* y *slave* están controlados por señales de reloj distintas, aunque de la misma frecuencia.

Cada uno de estos esquemas tiene sus ventajas e inconvenientes: El primero permite realizar CSS muy rápidos, pero requiere afinar de forma muy precisa el retraso de los componentes y un gran esfuerzo de diseño, por lo que su uso se limita a aplicaciones muy específicas y no será tratado en esta tesis. El segundo es más sencillo conceptualmente, pero no es tan rápido y, al igual que el primero, es tremendamente sensible a un fenómeno denominado *clock skew* (desplazamiento de reloj) en virtud del cual la señal de reloj no llega simultáneamente a todos los componentes del sistema. El tercero requiere más elementos de memoria y señales de reloj, pero permite ajustar la forma de onda de las mismas para hacer que el CSS funcione independientemente del retraso de los componentes y del *clock skew*. En esta tesis nos centraremos en esquemas de sincronización que, como el *máster-slave*, presenten tolerancia a un *clock skew* arbitrario. Comparado con los esquemas no tolerantes al *clock skew* mencionados, el esquema *máster-slave* presenta los siguientes inconvenientes:

- El consumo debido al conjunto de señales de reloj en cada ciclo de cómputo es más elevado. Al igual que en los esquemas de una fase, las señales de reloj cambian dos veces por cada ciclo de cómputo, pero hay dos señales de reloj en lugar de una.
- El retraso de la parte combinacional es de dos biestables en lugar de uno, lo que afecta negativamente a la velocidad del sistema.

Esto limita las prestaciones de los esquemas *máster-slave*, lo que debe tenerse en cuenta en aplicaciones de alta velocidad y/o bajo consumo.

## 1.4. Objetivos y organización de la tesis

A lo largo de los apartados anteriores hemos realizado un análisis introductorio a las partes secuenciales y combinacionales de los circuitos digitales. Se pretende aportar avances en la tarea de diseño de estos circuitos en dos vertientes, a saber, mejora de las prestaciones de puertas lógicas CMOS y mejora de las prestaciones de los esquemas de reloj. En particular, los objetivos principales de esta tesis son los siguientes:

1. Presentar y analizar nuevos estilos de diseño de puertas lógicas en tecnología CMOS y comparar sus prestaciones con el estilo de diseño CMOS complementario habitual. Esto requerirá llevar a cabo las siguientes tareas:

   *a*) Analizar los efectos no deseados que afectan a las puertas *bulk*-CMOS complementarias.

   *b*) Proponer y analizar una nueva forma de implementación que alivie algunos de estos problemas.

   *c*) Comparar cuantitativamente las prestaciones de la forma de implementación propuesta con la implementación habitual.

2. Proponer nuevos esquemas de sincronización tolerantes a imprecisiones en la señal de reloj y comparar sus prestaciones con esquemas habituales. Esto requerirá llevar a cabo las siguientes tareas:

*a*) Analizar los problemas de sincronización asociados al aumento de la densidad de integración y los esquemas de reloj empleados para resolverlos.

*b*) Proponer nuevos esquemas de sincronización que mejoren las prestaciones respecto a los tradicionales.

*c*) Estudiar la descripción de diseños que usen los nuevos esquemas mediante herramientas de diseño automático: lenguajes de descripción de hardware y software de síntesis automática.

*d*) Comparar cuantitativamente estos esquemas.

La primera parte de la tesis se dedica al primer objetivo, a saber, diseño de puertas lógicas CMOS de altas prestaciones. En el capítulo 2 se realizará el estudio de los efectos no deseados que afectan a las puertas, se analizará como afectan a las implementaciones tradicionales y se propondrá una nueva forma de implementación. En el capítulo 3 se compararán las prestaciones de la implementación propuesta con la tradicional.

La segunda parte de la tesis se dedicará al objetivo de desarrollar esquemas de temporización tolerantes a imprecisiones en la señal de reloj. En el capítulo 4 se describirá la fuente de estas imprecisiones y sus efectos, se propondrán nuevos esquemas de sincronización y se indicará como describir los nuevos esquemas planteados en lenguajes de descripción de hardware. En el capítulo 5 se compararán las prestaciones de los esquemas de sincronización propuestos con los tradicionales. Para finalizar se presentarán resumidas las principales conclusiones de esta tesis en el capítulo 6.

# Capítulo 2

# Diseño de puertas lógicas *bulk*-CMOS complementarias

Este capítulo está dedicado al estudio de puertas lógicas CMOS complementarias en tecnologías de fabricación tipo *bulk*, que son las más utilizadas como vimos en el apartado 1.2.1. En el capítulo se analizará las limitaciones de la implementación tradicional de estas puertas, lo que requerirá conocer algunos efectos no deseados que se dan en los transistores MOS. También se propondrá una forma alternativa de implementación que pretende aliviar estas limitaciones.

## 2.1. Efectos indeseados en puertas lógicas *bulk*-CMOS complementarias

En este apartado revisaremos varios fenómenos que pueden afectar negativamente al funcionamiento de las puertas lógicas complementarias construidas en tecnología *bulk*-CMOS. En concreto se describirá el efecto sustrato, la conducción subumbral, las corrientes de polarización inversa y las capacidades parásitas.

### 2.1.1. El efecto sustrato

Como se comentó en el apartado 1.2.1, los transistores MOS funcionan como interruptores controlados por la tensión de puerta. En el caso de un transistor NMOS (siendo el caso PMOS análogo) una diferencia de potencial positiva entre la puerta y el terminal *body* crea una zona de vaciamiento sobre la superficie de este último. Si dicho potencial se aumenta hasta llegar a la tensión umbral $V_t$, los electrones se atraen hasta la superficie creando una zona de inversión (el canal) que permite a la corriente fluir de drenador a fuente. Ahora bien, si se aumenta mucho la diferencia de potencial entre la fuente y el sustrato $V_{sb}$, la densidad de carga espacial en la zona de vaciamiento aumenta y la carga en el canal disminuye, lo que se traduce en una reducción de la conductividad del transistor. Este fenómeno se denomina *efecto sustrato*, y se ilustra en la figura 2.1.

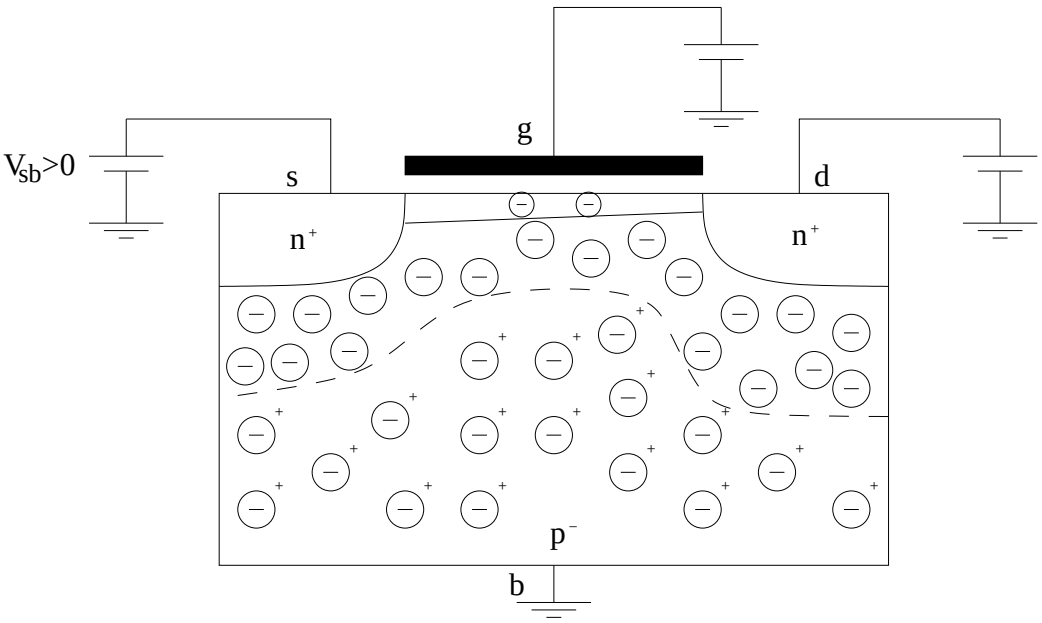



Figura 2.1: Ilustración del efecto sustrato.

Un incremento en la tensión puerta sustrato $V_{gs}$ puede restablecer la conductividad del canal, de modo que este efecto puede modelarse como una dependencia de la tensión umbral $V_t$ sobre $V_{sb}$ tal y como ilustra la fórmula siguiente [15, 16]:

$$V_t = V_{t0} + \gamma(\sqrt{2\phi_f + V_{sb}} - \sqrt{2\phi_f}) \tag{2.1.1}$$

donde $V_{t0}$ es la tensión umbral para $V_{sb} = 0$, $\gamma$ es un parámetro dependiente del proceso de fabricación denominado *coeficiente de efecto sustrato* y $\phi_f$ es el potencial de Fermi.

### 2.1.2. Conducción subumbral

Idealmente, la resistencia que presenta un transistor inactivo es infinita. En el mundo real, por el contrario, la conductividad de un transistor en el que la tensión de la puerta no alcanza $V_t$ no es exactamente igual a cero. Esto permite que fluya una pequeña corriente subumbral que contribuye de forma importante al consumo en estática [17, 18]. La magnitud de esta corriente disminuye al aumentar $V_t$, por lo que en algunas técnicas de reducción de consumo estático se aumenta de forma selectiva la tensión umbral. Por ejemplo, en la técnica CMOS de umbral variable (*Variable-Threshold* CMOS o VTCMOS) cuando un circuito CMOS está en modo *stand-by* el sustrato NMOS se polariza por debajo de GND, mientras que el PMOS se polariza por encima de VDD [19] lo que incrementa $V_{sb}$ y, en consecuencia, $V_t$. De este modo la corriente subumbral se reduce. Naturalmente emplear esta técnica implica cierta penalización en área dado que requiere un circuito independiente para controlar la polarización de los sustratos así como fuentes de tensión adicionales [20]. No se han realizado tantos estudios sobre la variación dinámica de la tensión del sustrato cuando el transistor está en funcionamiento, aunque sus resultados parecen muy prometedores [17].

### 2.1.3. Corrientes de polarización inversa

Otra fuente importante de consumo en estática en circuitos CMOS proviene de los diodos parásitos que aparecen en las uniones PN internas de los transistores. Esto se ilustra en el inversor *bulk*-CMOS de la figura 2.2.

Como puede observarse, cada unión PN constituye un diodo parásito, de modo que hay uno por cada terminal fuente o drenador y por cada pozo.

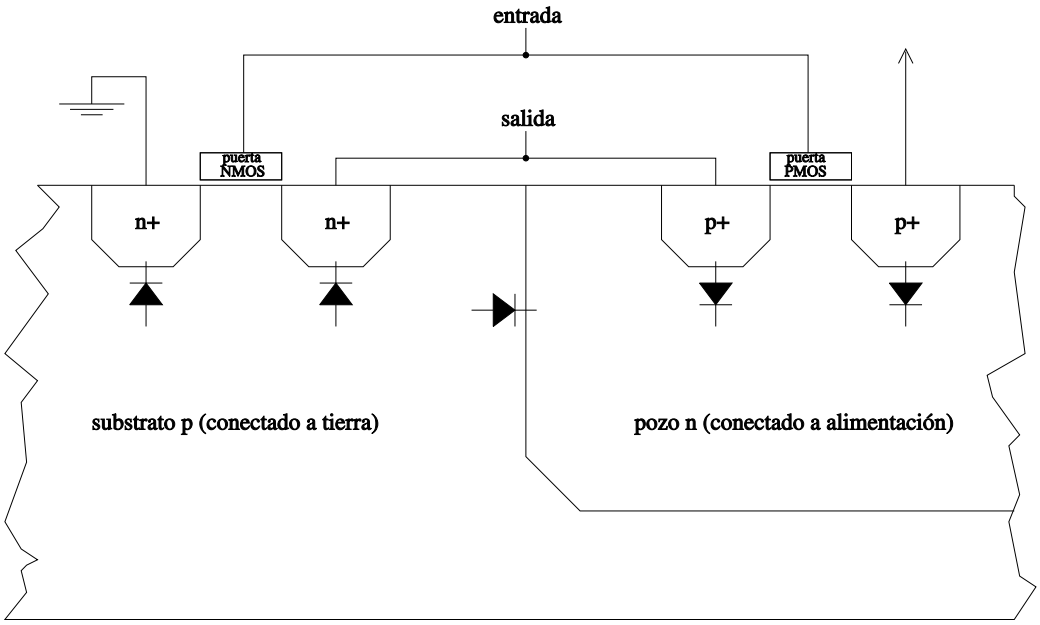



Figura 2.2: Diodos parásitos en un inversor CMOS.

Dependiendo del valor de entrada, los diodos correspondientes a terminales fuente o drenador pueden estar polarizados en inversa, conduciendo una pequeña corriente que contribuye al consumo estático. Las corrientes de polarización inversa son, junto con la conducción subumbral, una de las mayores fuentes de consumo estático en circuitos CMOS [17, 18].

### 2.1.4. Capacidades parásitas

Existen fenómenos capacitivos internos al transistor MOS que afectan notablemente a su funcionamiento. Las capacidades internas pueden clasificarse en dos tipos básicos [15, 21, 22]:

- Capacidades asociadas al terminal de puerta: La estructura Metal-Óxido-Semiconductor se corresponde con la de un condensador de placas conductoras paralelas donde la capa de óxido cumple el papel de dieléctrico. El movimiento de portadores de carga que se produce en la formación y desaparición de las zonas de inversión se corresponde con la carga y descarga de este condensador.

- Capacidades de unión: La zona de contacto entre el sustrato y los terminales fuente/drenador son uniones tipo PN polarizadas en inversa, lo que acarrea efectos capacitivos.

Estos efectos pueden modelarse añadiendo condensadores parásitos entre los distintos terminales del transistor. Evidentemente, la capacidad asociada a la puerta es proporcional a su superficie. El dispositivo que genere la tensión del terminal de puerta deberá encargarse de cargar y descargar dicha capacidad. Es por eso que la estructura y polarización del transistor afecta tanto a las prestaciones de la puerta a la que pertenece como a las prestaciones de la puerta que genera su entrada. Esto debe tenerse en cuenta al comparar distintas alternativas de implementación de puertas.

## 2.2. Comportamiento de la implementación con terminal *body* común

Como se comentó en el apartado 1.2.1, en las primeras tecnologías *bulk*-CMOS todos los transistores de uno de los tipos debían compartir un terminal *body* común. La tecnología empleada en la implementación del inversor de la figura 2.2, por ejemplo, usa un sustrato de tipo P, por lo que todos los transistores N comparten el terminal *body*. Los transistores del tipo complementario se construyen sobre pozos de semiconductor de dopado opuesto al del sustrato. En la implementación tradicional, si estos transistores pertenecen a la misma puerta se construyen sobre el mismo pozo.

El usar un mismo terminal *body* para los transistores puede afectar negativamente al retraso de las puertas. La figura 2.3, por ejemplo, muestra la implementación estándar de una puerta NAND estática de cuatro entradas. Si inicialmente las entradas 1, 2 y 3 están en nivel lógico alto y se produce una transición de subida en la entrada 0, la salida cambiará de 1 a 0 y se descargarán las capacidades de carga de los nodos A, B y C a través de la cadena de transistores NMOS. La conductividad del transistor NMOS asociado a la entrada 0 no se ve afectada por el efecto sustrato, pero sí la del resto de transistores en el árbol serie. La capacidad extra que debe ser

descargada de los nodos A, B y C así como la reducida conductividad de los transistores NMOS debido al efecto sustrato hace que el retraso de la entrada 0 aumente respecto al de otras entradas (en las que habría menos capacidades a ser descargadas). El efecto global es que el retraso de propagación aumenta y la pendiente de salida se degrada para las entradas conectadas a los transistores más próximos a tierra. El mismo fenómeno ocurre en puertas NOR para transiciones de entrada de bajada, así como en puertas complejas. Esta desigualdad en el retraso de las entradas limita el número total de las mismas a la hora de diseñar las puertas.

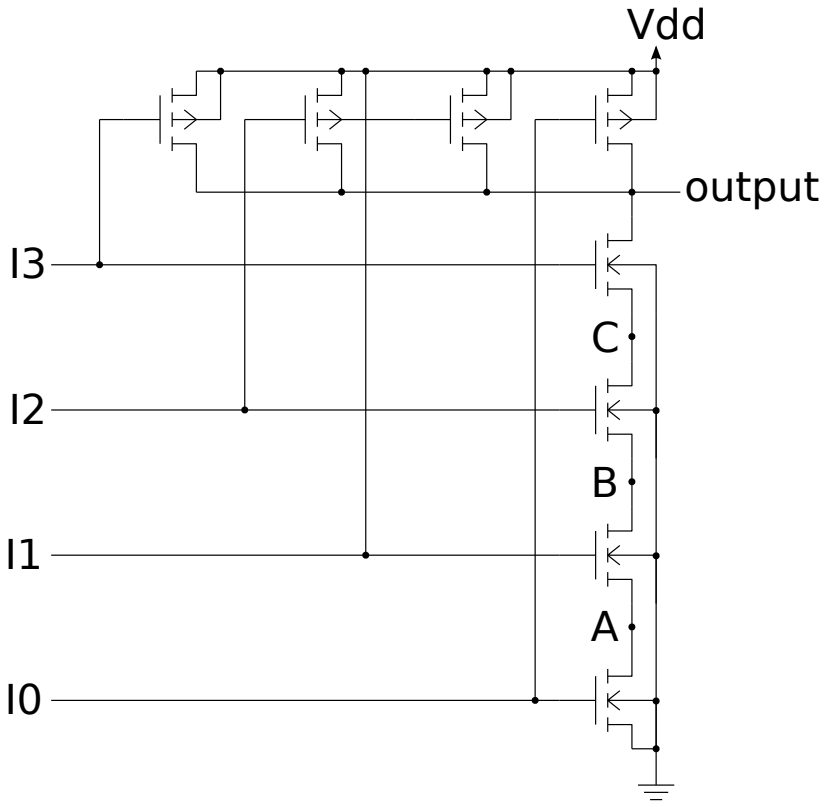



Figura 2.3: Implementación tradicional de una puerta NAND de cuatro entradas.

Hay que hacer notar que el efecto sustrato hace que la intensidad de la corriente subumbral que circula a través de una puerta dependa fuertemente del patrón de entrada. Esta intensidad puede variar drásticamente incluso para patrones de entrada que generan la misma salida. Por ejemplo, si la puerta de la figura 2.3 tiene todas sus entradas a uno salvo la entrada 0, entonces los nodos A, B y C se cargarán a un nivel alto y los transistores NMOS asociados a las entradas 1, 2 y 3 tendrán una tensión umbral mayor que la nominal y una conductividad reducida. Por el contrario, si la única entrada a nivel lógico bajo es la 3, entonces ningún transistor verá reducida su conductividad y la conducción subumbral será mayor. Este fenómeno se conoce como *efecto apilado* [23].

## 2.3. Implementación con fuente y pozo cortocircuitado

Como se comentó en el subapartado 1.2.1, la mayor parte de los circuitos electrónicos fabricados hoy en día utilizan procesos *bulk*-CMOS al ser una tecnología muy madura y económica [24]. No obstante, otros procesos de fabricación permiten construir transistores sobre materiales no semiconductores. Un ejemplo notable son las tecnologías SOI (*Silicon On Insulator*). En los procesos de fabricación SOI los transistores se forman sobre una capa de material aislante [25]. Pese a la mayor complejidad (y por tanto mayor coste) de los procesos de fabricación SOI frente a los *bulk*-MOS, la tecnología SOI presenta importantes ventajas. Por ejemplo, un subtipo de tecnología SOI denominada SOS (*Silicon on Sapphire*, silicio sobre zafiro) presenta mayor tolerancia a los rayos cósmicos [26, 27], por lo que es la tecnología predominante en la industria aeroespacial. Otra ventaja de las tecnologías SOI es que permiten independizar el terminal *body* tanto de los transistores NMOS como el de los transistores PMOS, evitando por tanto el efecto sustrato [8]. Esto último no puede decirse de todos los procesos *bulk*-CMOS. Tomemos como ejemplo la tecnología de fabricación de la figura 1.3: A los transistores PMOS se les puede dotar de terminales

*body* independientes construyéndolos sobre pozos N separados, pero todos los transistores NMOS deberán compartirlo al estar construidos directamente sobre el propio sustrato. No obstante, las tecnologías *bulk*-CMOS *de pozo múltiple* permiten dotar de terminales *body* independientes a ambos tipos de transistores. En estas tecnologías se crean pozos sobre el sustrato como ocurre en cualquier proceso *bulk*-CMOS pero, además de eso, dentro de estos mismos pozos es posible a su vez crear pozos de dopado opuesto sobre los que colocar transistores. Esto se ilustra en la figura 2.4.

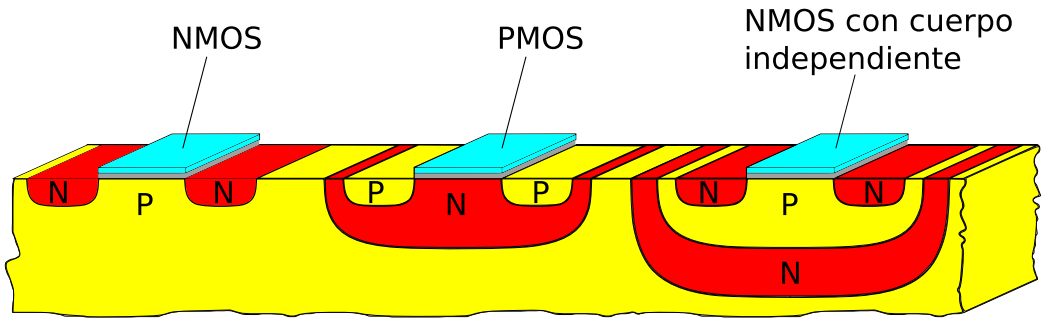



Figura 2.4: Corte transversal de transistores complementarios construidos en un proceso *bulk*-CMOS de pozo múltiple (se han omitido los contactos a los terminales).

Al igual que en tecnologías SOI, los procesos de fabricación *bulk*-CMOS de múltiple pozo hacen posible dotar a cada transistor NMOS y PMOS de cada puerta de un terminal *body* dedicado. De esta forma es posible realizar una implementación alternativa a la tradicional de la figura 2.3 en la que los transistores del árbol serie tengan conectados sus terminales de fuente y *body*. Esta alternativa, que en adelante llamaremos INBO, del inglés "*INdependent BOdies*" se ilustra en la figura 2.5 para el caso de una puerta NAND de cuatro entradas. El uso de pozos independientes para cada transistor del árbol serie implica penalización en área respecto a la implementación tradicional a la que llamaremos COBO, del inglés "*COmmon BOdies*". Por esa razón el uso de terminales *body* independientes no suele considerarse en diseño lógico *bulk*-CMOS. No obstante, los beneficios que pueden reportar la alternativa INBO justificarían su uso en aplicaciones concretas y/o en partes específicas de los circuitos lógicos. Estas ventajas potenciales se resumen a continuación:

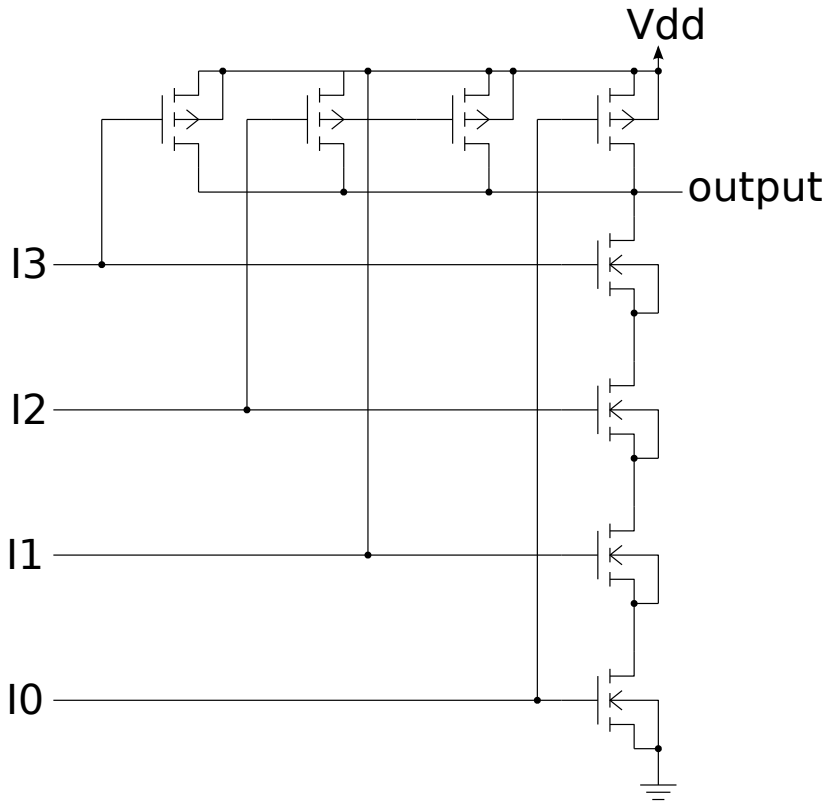



Figura 2.5: Implementación de una puerta NAND en la que se anula el efecto sustrato.

- Tal y como se vio en el apartado 2.1.1, la conductividad de un transistor disminuye al aumentar $V_{sb}$ en virtud del efecto sustrato. Al conectarse el terminal *body* de cada transistor serie con su fuente correspondiente tenemos $V_{sb} = 0$, por lo que este efecto se elimina. La conductancia de los transistores durante las transiciones se mejora, por lo que la respuesta de la puerta será más rápida. La eliminación del efecto sustrato también tiene su lado negativo ya que reduce el efecto apilado y, en consecuencia, hace que aumente la conducción subumbral.

- Al no encontrarse polarizados los pozos en la implementación INBO, las corrientes de polarización inversa que circulan a través de los diodos parásitos formados por las uniones PN en el drenador de los transistores serie pueden anularse. Esto ayudaría a reducir el

consumo estático respecto a la implementación tradicional. La mejora en consumo depende del patrón de entrada de la puerta ya que, en la implementación tradicional, este patrón determina cuantos diodos parásitos se encuentran en polarización inversa.

- Es evidente que, en esta implementación, no existe capacidad parásita entre el terminal *body* y la fuente de los transistores serie. Por otro lado, al dejar sin polarizar el terminal *body* se tiene que la capacidad de unión entre el terminal *body* y el drenador se carga a una menor diferencia de potencial, acumulando menos carga. Lo mismo puede decirse de la capacidad entre la puerta y el terminal *body*. Todo esto debería mejorar la velocidad de conmutación y el consumo dinámico. No obstante, la capacidad entre el pozo y el sustrato puede afectar negativamente a ambos parámetros. A primera vista, podría parecer que esta última capacidad es la mayor y más relevante, pero esto no es necesariamente cierto: la capacitancia depende del dopado, y el dopado del sustrato puede diferir del de las terminales fuente/drenador, aunque sea del mismo tipo.

## 2.4. Resumen

En este capítulo se ha revisado la implementación tradicional de puertas lógicas CMOS complementarias construidas en procesos de fabricación tipo *bulk* analizando los aspectos más importantes que limitan sus prestaciones. Se ha puesto de manifiesto como el uso de los actuales procesos de fabricación de múltiple pozo permite aliviar algunas de estas limitaciones, en especial las debidas al efecto sustrato, al hacer posible conectar los terminales de fuente y terminal *body* en cada uno de los transistores.

Aunque esta alternativa de diseño supone una clara penalización en el área ocupada por la puerta, las potenciales ventajas pueden justificar su uso en aplicaciones o subcircuitos específicos. Por este motivo, en el capítulo siguiente haremos un análisis comparativo entre la implementación tradicional (COBO) y la alternativa (INBO).

# Capítulo 3

# Prestaciones de las puertas CMOS complementarias

El objetivo de este capítulo es comparar las prestaciones de puertas CMOS complementarias implementadas de la forma tradicional (COBO) con las de puertas homólogas implementadas de acuerdo a la topología INBO descrita en el capítulo 2. La comparación se realiza en términos de retraso, consumo estático, consumo dinámico y área ocupada. En particular se compararán puertas NAND y NOR de cuatro entradas, habitualmente disponibles en bibliotecas de celdas, así como puertas NAND y NOR de un número de entradas tan inusualmente alto como ocho para ver si la topología propuesta las hace viables. Además, las comparaciones serán de dos tipos: en el primero las puertas COBO e INBO a comparar tendrán la misma geometría de transistores; en el segundo las implementaciones COBO e INBO ocuparán la misma área.

En el apartado siguiente se describirán los detalles de las puertas comparadas y las condiciones de test. En el apartado 3.2 se compararán puertas con la misma geometría de transistores, mientras que en el 3.3 la comparación se realizará sobre puertas de la misma área. Por último, los resultados más relevantes se resumirán en el apartado 3.4.

## 3.1. Banco de pruebas

Para poder dotar tanto a los transistores NMOS como a los PMOS de pozos independientes se ha utilizado una tecnología de fabricación *bulk*-MOS de múltiple pozo. En concreto el proceso CMOS de 0.18 µm de United Microelectronics Corporation. Como hemos comentado se han implementado puertas NAND y NOR de cuatro y ocho entradas para comparar las prestaciones de las implementaciones COBO e INBO. Los *layouts* de las puertas diseñadas pueden observarse en el apéndice B. Dado que la implementación de una puerta INBO requiere más área que la de una puerta COBO análoga con la misma geometría de transistores, se harán dos tipos de comparaciones:

- Usando transistores con la misma geometría en ambas implementaciones: Cada transistor tiene longitud mínima y los transistores de la rama en paralelo tendrán ancho mínimo. Los transistores de la rama serie podrán ser más anchos para balancear los tiempos de subida y bajada de la salida. Se comparan puertas de cuatro entradas, que es un número usual (figuras B.1, B.2, B.3 y B.4), y de un número de entradas inusualmente grande, concretamente ocho (figuras B.5, B.6, B.7 y B.8), para comprobar si la implementación INBO las hace viables. Es fácil apreciar en estas figuras como, para la misma geometría, la implementación INBO requiere un área notablemente mayor dado que cada transistor serie está construido sobre un pozo independiente.

- Usando transistores sobredimensionados en la implementación COBO para que ambas implementaciones ocupen el mismo área: A modo de ejemplo, la puerta NOR de cuatro entradas con implementación INBO de la figura B.4 se comparará con una puerta COBO análoga con transistores sobredimensionados (figura B.9).

Las entradas de cada puerta se han etiquetado de forma que aquellas que están conectadas a transistores serie más alejados del nodo correspondiente a la salida tienen un índice menor como ejemplifican las figuras 2.5 y B.3.

En esta última figura puede observarse que toda la puerta está contenida en un pozo de material tipo N, de forma que los transistores NMOS pueden construirse dentro de pozos de material P contenidos en el primero (laterales del *layout*). Los transistores PMOS se construyen directamente sobre el pozo N (centro del *layout*). Se ha realizado el *layout* de forma rectangular para ahorrar área y para simplificar la colocación y rutado si se usa una metodología de celdas estándar.

Para evaluar las prestaciones de las puertas se someterá cada diseño a simulación eléctrica con la herramienta HSPICE tras la extracción *post-layout* de parásitos. En la simulación se usará una tensión de alimentación nominal de 1.8 V. La salida de cada puerta se cargará con entradas de puertas idénticas tal y como se ilustra en la figura 3.1. Además, las formas de onda de entrada de cada puerta serán generadas por una cadena de puertas similares para hacerla más parecida a una entrada real. Para cada puerta se medirá el retraso pin a pin, su consumo dinámico, su consumo estático, su producto energía-retraso y el área que ocupa. La medición del retraso y del consumo

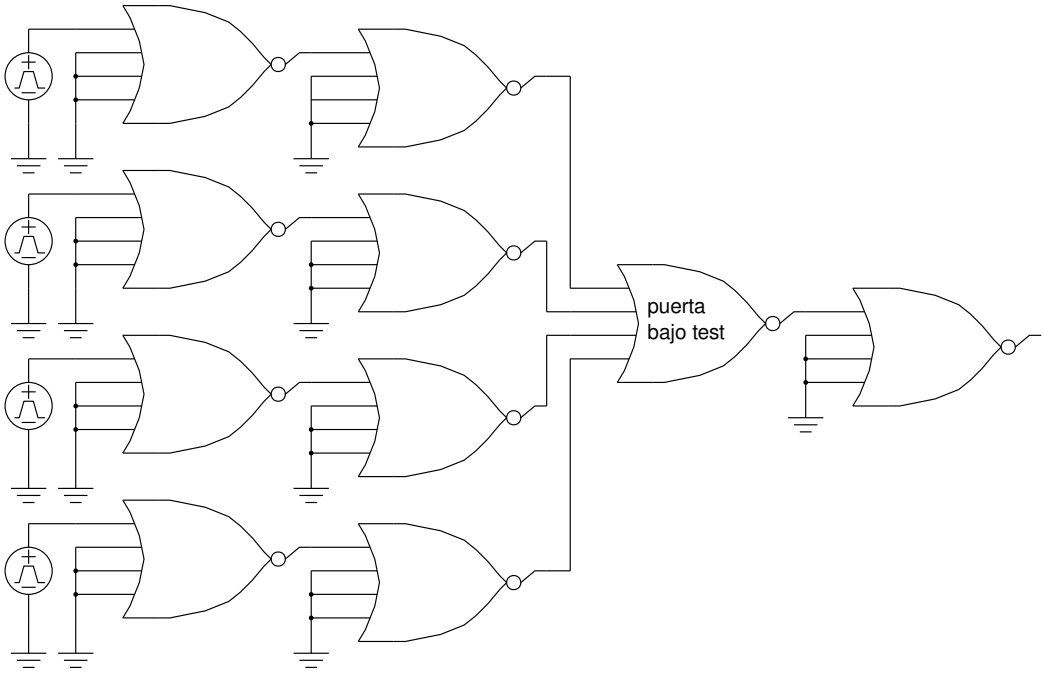



Figura 3.1: Circuito empleado para evaluar las prestaciones de una puerta NOR de cuatro entradas.

dinámico asociado a la entrada de una puerta requiere cambiar el valor lógico de dicha entrada y que dicho cambio provoque un cambio en el valor lógico de la salida de la puerta. Cuando un cambio en una entrada determinada provoca un cambio en la salida se dice que dicha entrada está *sensibilizada*. Si, por ejemplo, todas las entradas de una puerta NAND tienen un valor lógico 1, un cambio de cualquier entrada provocará que la salida cambie de valor lógico, de modo que todas las entradas están sensibilizadas.

## 3.2. Comparación de puertas con la misma geometría

En este apartado se compararán pares de puertas COBO e INBO con la misma geometría de transistores, es decir el que cada transistor de cada puerta COBO tendrá la misma anchura y longitud que el transistor homólogo de la puerta INBO.

### 3.2.1. Puertas de cuatro entradas

En primer lugar analizaremos las prestaciones de las puertas de cuatro entradas correspondientes a las figuras B.1, B.2, B.3 y B.4.

#### 3.2.1.1. Área

Todos los transistores se han dimensionado con longitud mínima y un ancho de 240 nm a excepción de los transistores P de las puertas NOR. Estos últimos tienen mayor anchura, 1440 nm, para mantener similares sus retrasos de subida y bajada. El área ocupada por cada una de las puertas se muestra en la tabla 3.1. Como se comentó, hay un gran incremento de área al usar el estilo INBO al requerirse que cada transistor serie tenga su propio pozo.

#### 3.2.1.2. Retraso pin a pin

Como se comentó en la sección 3.1, el cambio de estado lógico de una entrada sensibilizada de una puerta provoca un cambio en el estado lógico

| | COBO | INBO | AUMENTO |
|---|---|---|---|
| NAND | 14.2 $\mu m^2$ | 43.7 $\mu m^2$ | 208 % |
| NOR | 19.6 $\mu m^2$ | 48.5 $\mu m^2$ | 147 % |

Tabla 3.1: Área ocupada por las puertas de cuatro entradas.

de la salida de dicha puerta. Por ejemplo, para que una transición en una entrada de la puerta bajo test de la figura 3.1 provoque una transición en la salida es necesario que las demás entradas tengan un valor lógico 0. Cuando un cambio en el valor lógico de una entrada hace que cambie el valor lógico de salida se define el retraso pin a pin asociado a esa entrada como el tiempo que transcurre desde que se produce la transición de entrada hasta que se produce la transición de salida. Esta definición es problemática, ya que las señales reales son de naturaleza analógica y el cambio entre niveles lógicos no es instantáneo. Un convenio ampliamente utilizado es definir el instante de cambio como aquel en el que la señal pasa por el valor medio de las tensiones nominales correspondientes a los valores lógicos alto y bajo. Si, como es habitual, el valor lógico bajo se corresponde con una tensión 0 y el valor lógico alto con la tensión de alimentación, entonces el instante de cambio será aquel en el que la señal pase por el 50 % de la alimentación. En nuestras mediciones de retraso usaremos este convenio. Además, por cada entrada distinguimos dos tipos de retraso: los retrasos de propagación de subida son aquellos en los que un cambio en la entrada hace que la salida cambie de valor lógico bajo a valor lógico alto; los retrasos de propagación de bajada son aquellos en los que un cambio en la entrada hace que la salida cambie de valor lógico alto a valor lógico bajo.

Los retrasos de propagación medidos en función de la carga de salida para ambas implementaciones se ilustran en las figuras 3.2, 3.3, 3.4 y 3.5 (el valor numérico se encuentra en las tablas del apéndice C.1.1). Puede observarse una notable reducción del retraso con la implementación INBO en casi todos los casos. No obstante, el retraso de la entrada 3 (es decir, aquella conectada al transistor serie conectado directamente al nodo de salida) es similar en las implementaciones COBO e INBO dado que las capacidades parásitas de los nodos intermedios no influyen cuando esta entrada está sensibilizada: tal

y como aclaran las figuras 2.3 y 2.5, las capacidades ya están descargadas cuando la entrada sube en las puertas NAND (o cuando baja en las puertas NOR) y son desconectadas del nodo de salida cuando la entrada baja (o sube en las puertas NOR). Para el resto de las entradas, la influencia de las capacidades parásitas hace que el estilo INBO presente mejores prestaciones de acuerdo a lo previsto en la sección 2.3. Nótese que las mejoras más notables en velocidad se producen en las entradas de respuesta más lenta, por lo que el retraso es más homogéneo para todas las entradas en la implementación INBO. Se observa que la mejora es más pronunciada en la puerta NOR, que obtiene una mayor reducción del retraso en todos los casos: si comparamos el retraso de subida de la entrada más lenta (la 0) para carga de salida unitaria en ambas implementaciones vemos una mejora de un 19 % en la puerta NAND (véase la figura 3.4 y los valores numéricos en las tablas C.7 y C.5), mientras que la reducción es del 23 % en la puerta NOR (véase la figura 3.2 y los valores numéricos en las tablas C.3 y C.1).

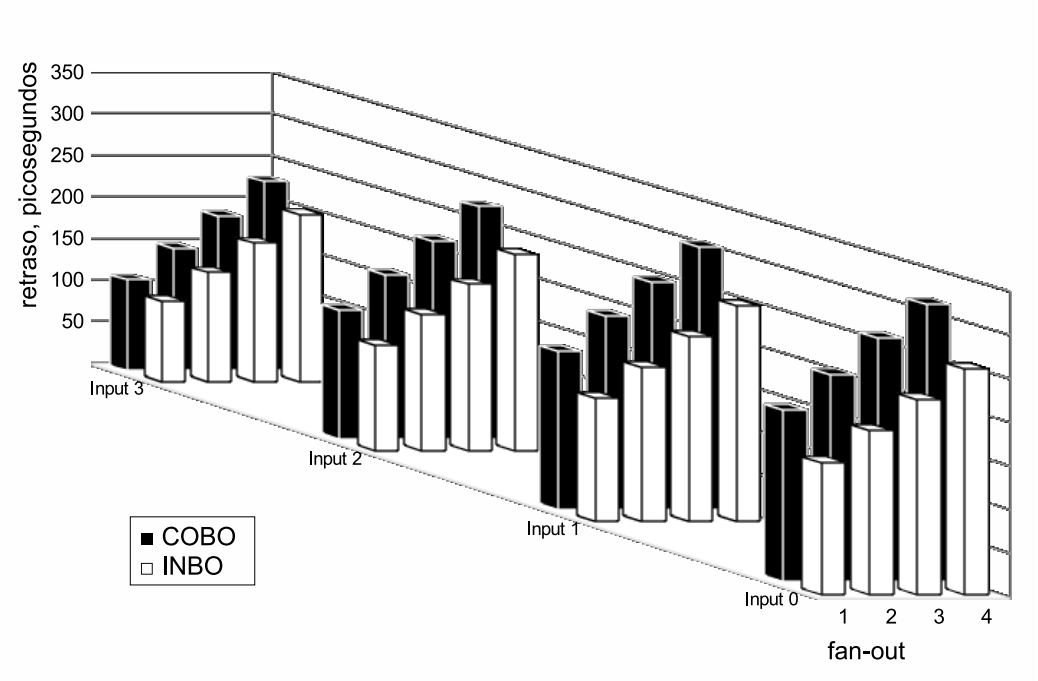



Figura 3.2: Retrasos de subida de las puertas NOR de cuatro entradas.

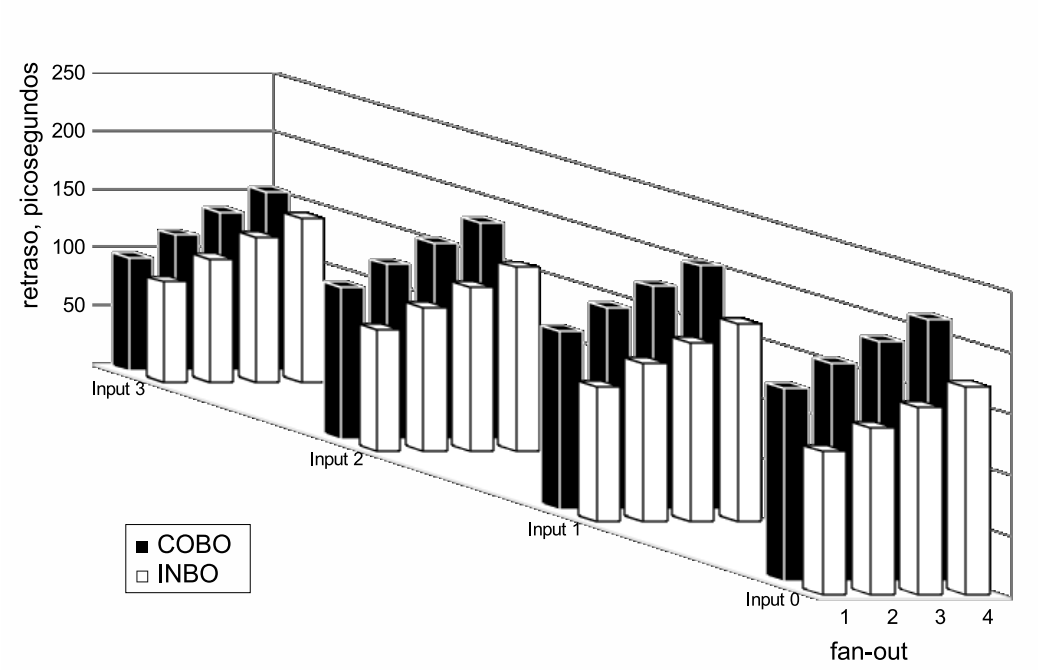



Figura 3.3: Retrasos de bajada de las puertas NOR de cuatro entradas.

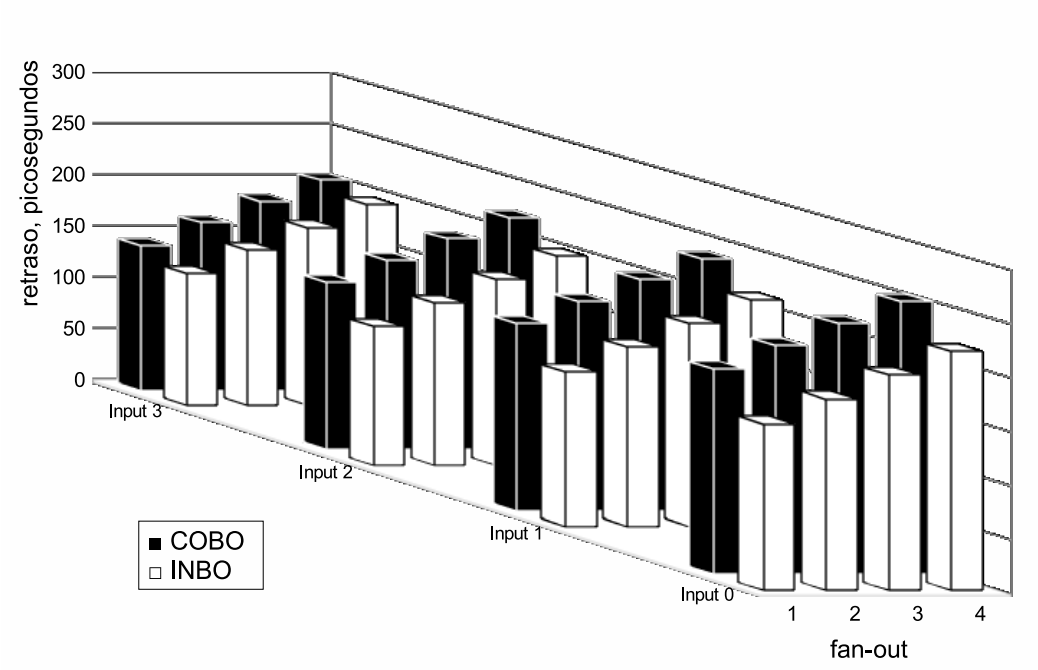



Figura 3.4: Retrasos de subida de las puertas NAND de cuatro entradas.

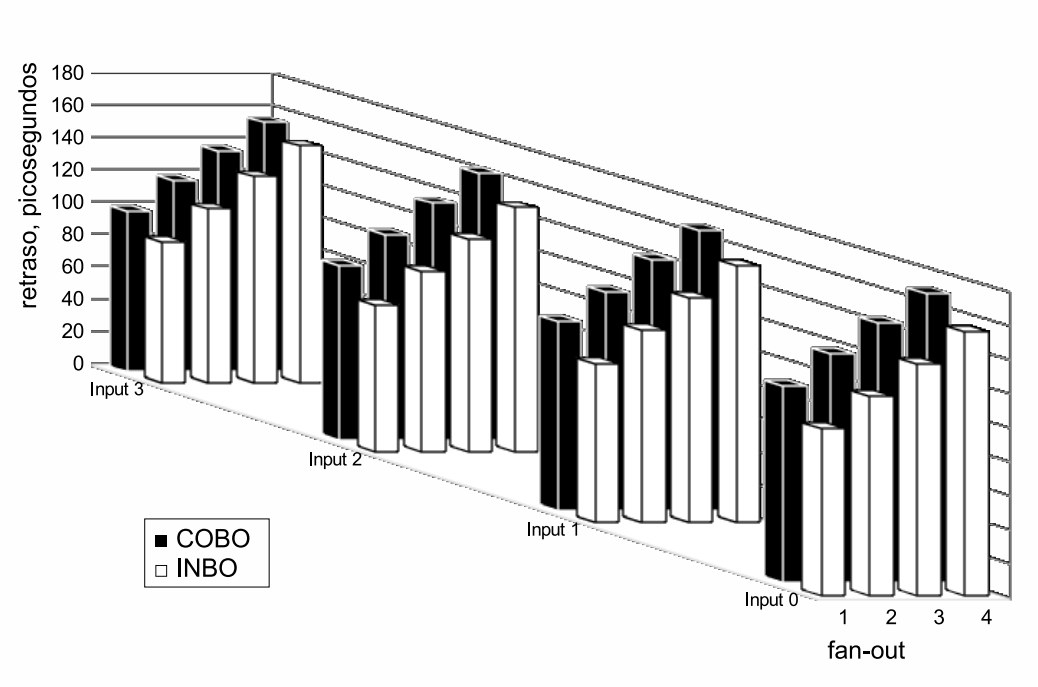



Figura 3.5: Retrasos de bajada de las puertas NAND de cuatro entradas.

#### 3.2.1.3. Consumo dinámico

Como se indicó en la sección 1.1, el consumo dinámico de una puerta es el que se produce cuando cambia su estado lógico. Por tanto, para medirlo es necesario sensibilizar una entrada y aplicar a dicha entrada transiciones tal y como se hizo en la sección 3.2.1.2. La energía consumida por la puerta al recibir un pulso en una entrada sensibilizada para ambas implementaciones se ilustra en las figuras 3.6 y 3.7 (el valor numérico se encuentra en las tablas del apéndice C.1.2). Se observa una notable mejora en el consumo dinámico de la puerta NOR usando la implementación INBO en casi todas las entradas. El ahorro es mayor en las entradas con mayor consumo. Así, la energía consumida cuando conmuta la entrada 0 se reduce hasta un 19 % para carga de salida unitaria (véanse las tablas C.9 y C.10). La importancia relativa de las capacidades parásitas en los nodos intermedios del árbol serie disminuye al aumentar la carga en la salida, de modo que para mayores *fan-out* el porcentaje de ahorro es menor (sobre un 10 % para *fan-out* 4).

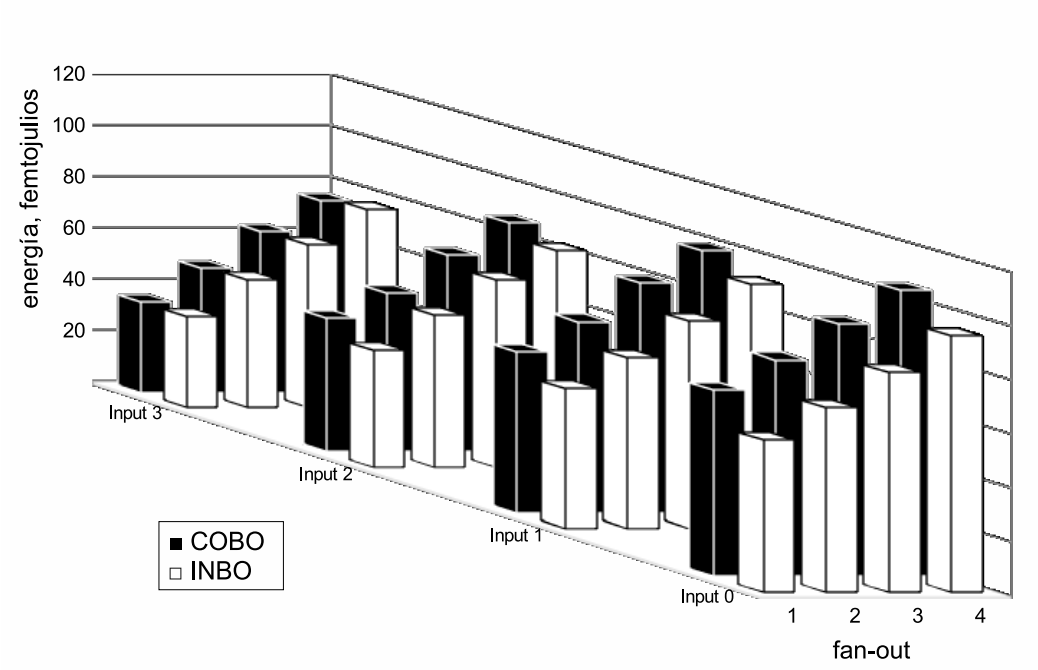



Figura 3.6: Consumo dinámico de las puertas NOR de cuatro entradas.

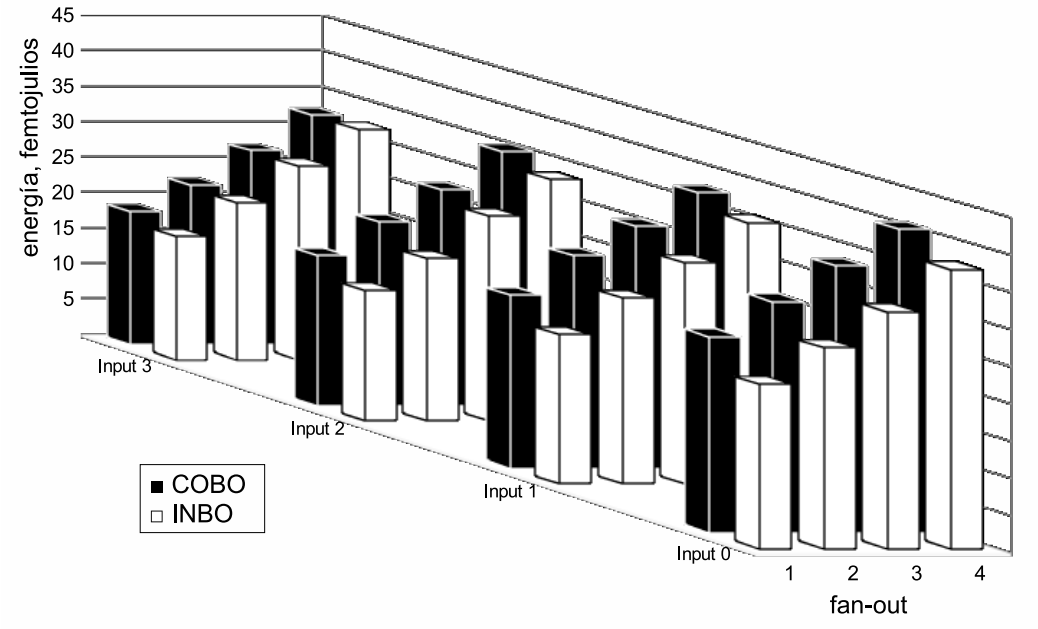



Figura 3.7: Consumo dinámico de las puertas NAND de cuatro entradas.

#### 3.2.1.4. Producto energía-retraso

El producto energía-retraso o EDP (*Energy-Delay Product*) para ambas implementaciones se ilustra en las figuras 3.8 y 3.9 (el valor numérico se encuentra en las tablas del apéndice C.1.3). Se ha calculado eligiendo, para cada entrada, su peor retraso (de subida o bajada) y multiplicándolo por la energía consumida cuando se aplica un pulso a dicha entrada. El uso de la implementación INBO reduce el EDP en casi todos los casos, siendo mayor la reducción en las entradas de índice menor. La entrada 0 de la puerta NOR, por ejemplo, ve reducido su EDP entre un 28 % y un 38 % dependiendo del *fan-out*.

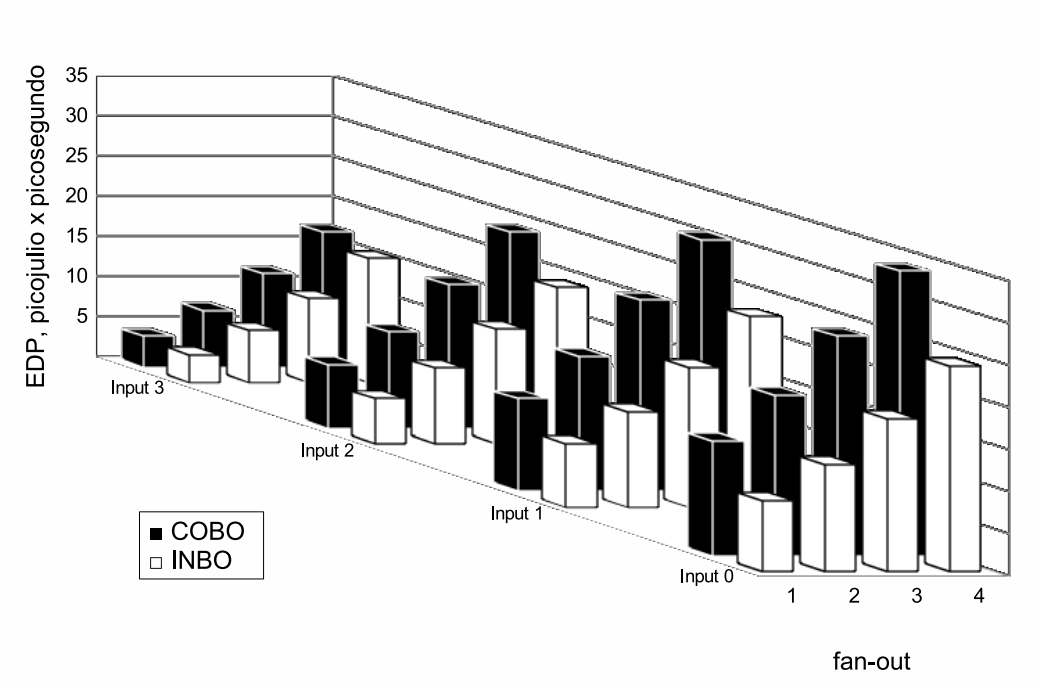



Figura 3.8: Producto energía retraso de las puertas NOR de cuatro entradas.

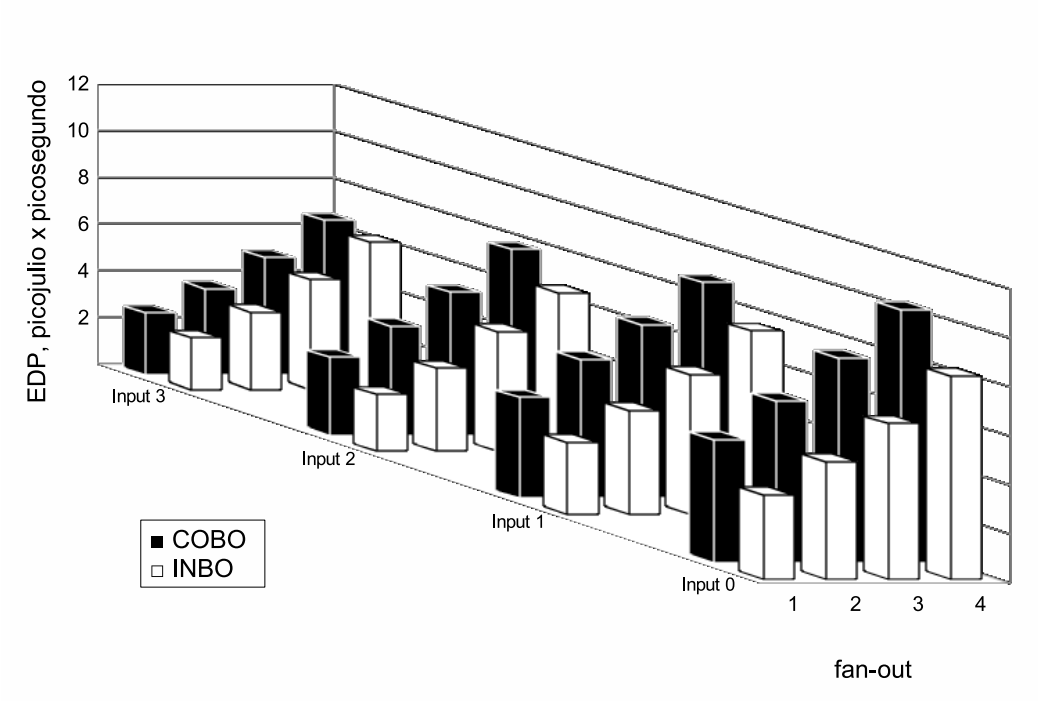



Figura 3.9: Producto energía retraso de las puertas NAND de cuatro entradas.

#### 3.2.1.5. Consumo estático

El consumo estático de las puertas NAND y NOR de cuatro entradas se ha medido por simulación eléctrica para cada posible vector de entrada. Los vectores de entrada se han numerado de forma que el índice asociado al vector de entrada $(I_3, I_2, I_1, I_o)$ es $(I_3 * 2^3 + I_2 * 2^2 + I_1 * 2^1 + I_o * 2^0)$. En el vector de entrada 5, por ejemplo, $(I_3, I_2, I_1, I_o) = (0, 1, 0, 1)$.

Cuando el árbol en el que los transistores están en serie está conduciendo, la tensión de los terminales *body* de los transistores coincide en las implementaciones, COBO e INBO, por lo que, tal y como muestra la tabla 3.2, el consumo en este caso no varía entre ambas implementaciones. Este caso se da cuando la puerta NAND recibe el vector 15 y cuando la NOR recibe el vector 0.

| | NAND | NOR |
|---|---|---|
| vector de entrada | 15 | 0 |
| consumo estático en el caso COBO | 76.1 pW | 483.5 pW |
| consumo estático en el caso INBO | 76.0 pW | 483.5 pW |

Tabla 3.2: Consumo estático de las puertas de cuatro entradas cuando el árbol serie está conduciendo.

Por otro lado, cuando el árbol serie no está conduciendo, la implementación INBO muestra menor consumo estático para casi todos los vectores de entrada tal y como se ilustra en las figuras 3.10 y 3.11 (el valor numérico del consumo estático de ambas implementaciones se encuentra en las tablas del apéndice C.1.4). En estas figuras los vectores de entrada se han agrupado según el número de transistores del árbol serie que activan. Cuanto menor sea ese número mayor será la impedancia presentada por el árbol serie y menor será la corriente subumbral, que es la mayor responsable de consumo en estática.

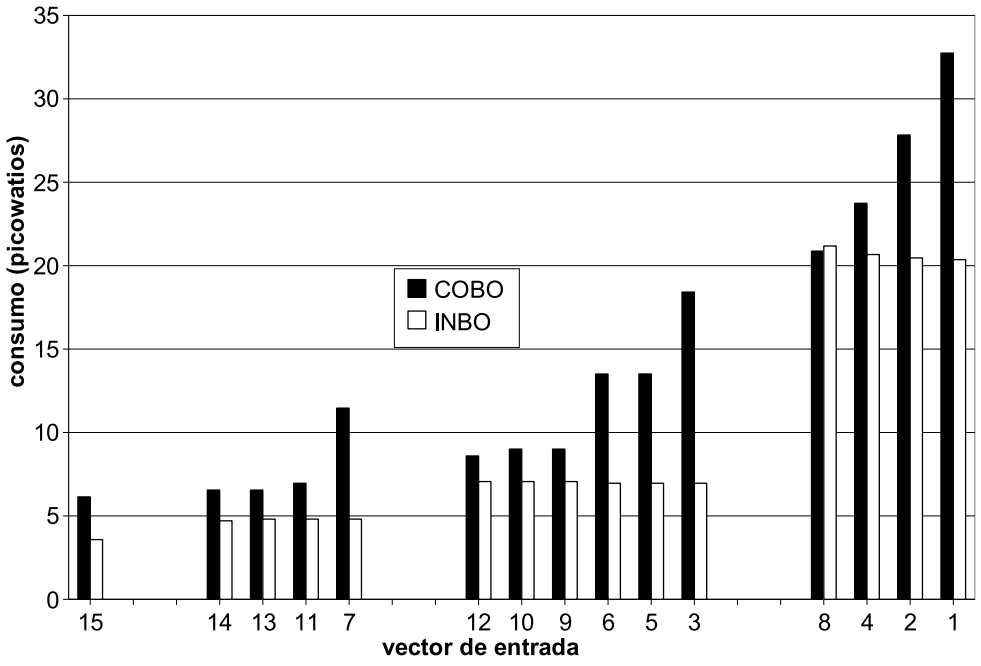



Figura 3.10: Consumo estático de las puertas NOR de cuatro entradas cuando el árbol de transistores serie no conduce.

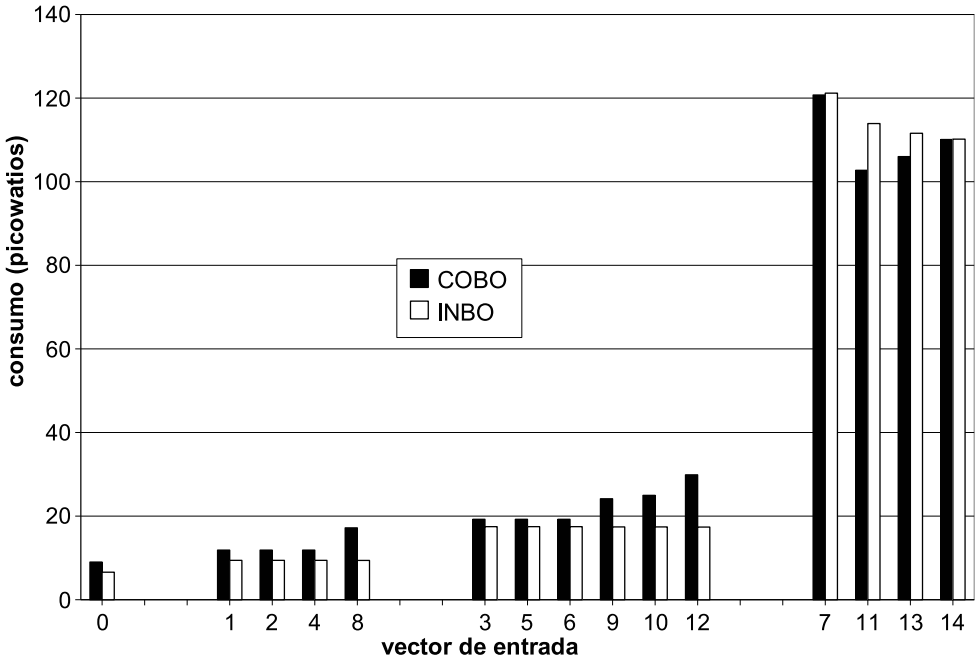



Figura 3.11: Consumo estático de las puertas NAND de cuatro entradas cuando el árbol de transistores serie no conduce.

Puede observarse que el consumo estático de las puertas INBO es casi constante dentro de cada grupo, mientras que varía de forma significativa en las puertas COBO. La razón es que, en la implementación COBO, el consumo estático dentro de cada grupo depende del número de diodos parásitos de las terminales drenador/fuente que se encuentren en polarización inversa (ver figura 2.2). A su vez esto está determinado por el número de transistores del árbol serie que están conectados al nodo de salida por un camino activo. Así, en la implementación COBO de la puerta NAND el vector de entrada $(I_3, I_2, I_1, I_o) = (1, 1, 0, 0)$ presenta más consumo en estática que el vector $(I_3, I_2, I_1, I_o) = (1, 0, 1, 0)$ dado que para el primero hay tres diodos en polarización inversa mientras que para el segundo solo hay dos. Los vectores 9 y 10, a su vez, presentarán más consumo que los vectores 3, 5 y 6 dado que en estos últimos únicamente el diodo parásito correspondiente al terminal de drenador directamente conectado al nodo de salida se encuentra en polarización inversa. Esta variación no se observa en las

implementaciones INBO: en ellas los diodos parásitos correspondientes a las uniones drenador/*body* no pueden encontrarse en polarización inversa ya que los pozos no están polarizados (excepto los de los transistores conectados a la entrada 0). En resumen, la ausencia de corrientes de polarización inversa de las implementaciones INBO hace que presenten menor consumo en estática en casi todos los casos, y es la razón de que este consumo sea casi constante dentro de cada grupo de vectores.

Como se mencionó en el apartado 2.3, los transistores de la implementación INBO presentan mayor conductividad al no experimentar el efecto sustrato. Esto puede comprobarse cuando hay un solo transistor inactivo en el árbol serie: los transistores que se encuentran entre el nodo de salida y el transistor inactivo presentan menor impedancia en la implementación INBO que en la COBO. Esta es la razón por la que la implementación INBO de la puerta NAND presenta un consumo estático ligeramente mayor para los vectores 7, 11, 13 y 14 mientras que la implementación INBO de la puerta NOR tiene un consumo estático ligeramente mayor para el vector de entrada 8.

Considerando todos los vectores de entrada con la misma probabilidad, el consumo en estática usando la implementación INBO se reduce un 10 % para las puertas NOR y un 5 % para las puertas NAND.

### 3.2.2. Puertas de ocho entradas

En este apartado veremos las prestaciones de las puertas de ocho entradas correspondientes a las figuras B.5, B.6, B.7 y B.8 de forma análoga a como se hizo con las puertas de cuatro entradas en la sección 3.2.1.

#### 3.2.2.1. Área

De nuevo, todos los transistores se han dimensionado con longitud mínima y se ha dotado a los transistores N de una anchura de 240 nm. Una vez más, para mantener similares los tiempos de subida y bajada los transistores P se han dimensionado con mayor anchura (480 nm en el caso de las puertas NAND y 2880 nm en el de las puertas NOR). El área resultante

de cada puerta se muestra en la tabla 3.3. De nuevo, la necesidad de pozos independientes en la implementación INBO implica un notable aumento del área ocupada.

| | COBO | INBO | AUMENTO |
|---|---|---|---|
| NAND | 23.4 $\mu m^2$ | 95.7 $\mu m^2$ | 309 % |
| NOR | 42.8 $\mu m^2$ | 163.1 $\mu m^2$ | 281 % |

Tabla 3.3: Área ocupada por las puertas de ocho entradas.

#### 3.2.2.2. Retraso pin a pin

Los retrasos medidos en función de la carga de salida se ilustran en las figuras 3.12, 3.13, 3.14 y 3.15 (el valor numérico se encuentra en las tablas del apéndice C.2.1). La implementación INBO muestra una mejora en velocidad aún más notable que en el caso de las puertas de cuatro entradas. Una vez más, la implementación INBO muestra retrasos más homogéneos al ser mayor la reducción del retraso en las entradas más lentas. Si comparamos el retraso de subida de la entrada más lenta (la 0) para carga de salida unitaria en ambas implementaciones vemos una mejora de un 22 % en la puerta NAND, mientras que la reducción es del 29 % en la puerta NOR.

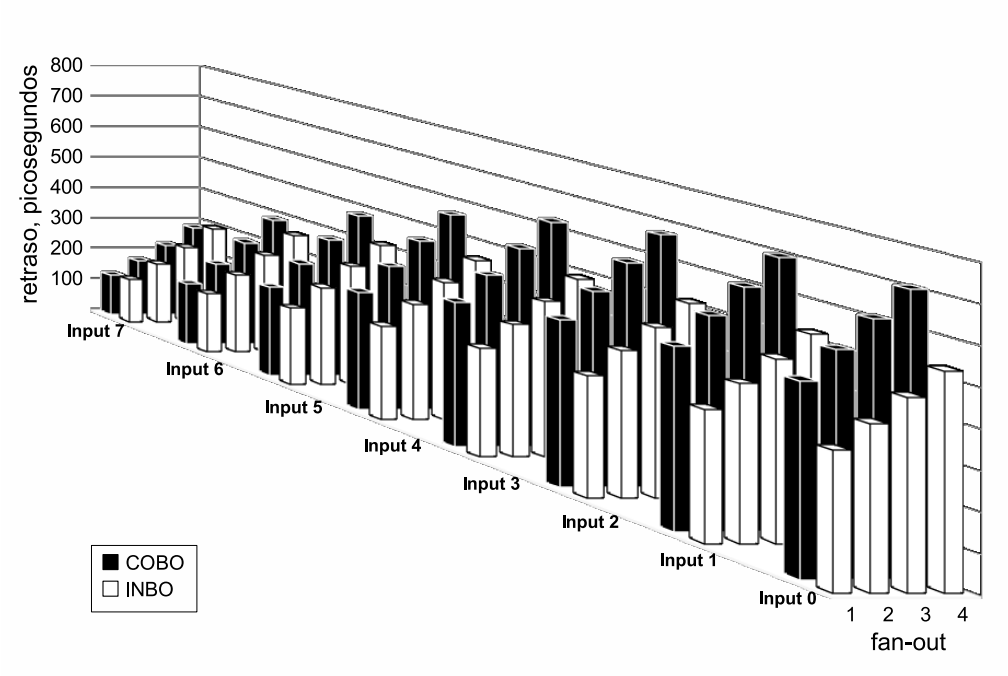



Figura 3.12: Retrasos de subida de las puertas NOR de ocho entradas.

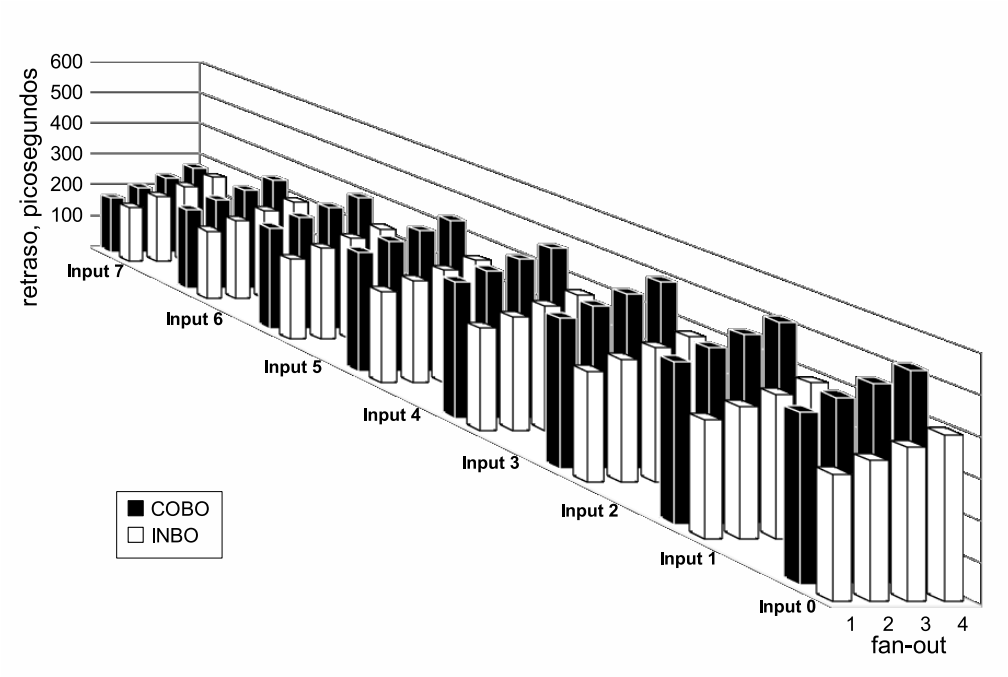



Figura 3.13: Retrasos de bajada de las puertas NOR de ocho entradas.

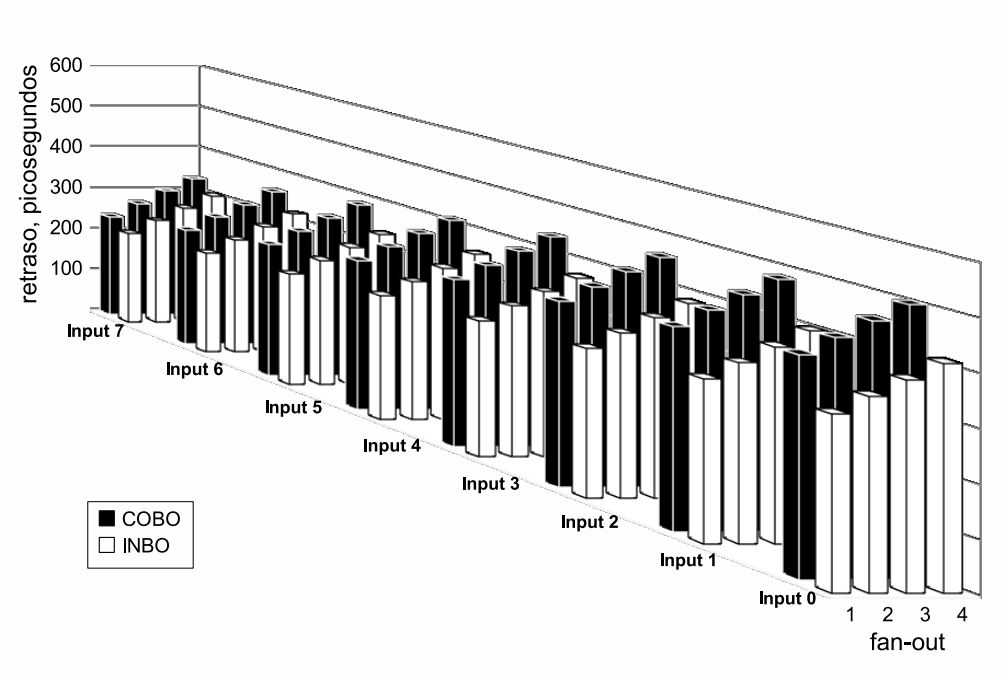



Figura 3.14: Retrasos de subida de las puertas NAND de ocho entradas.

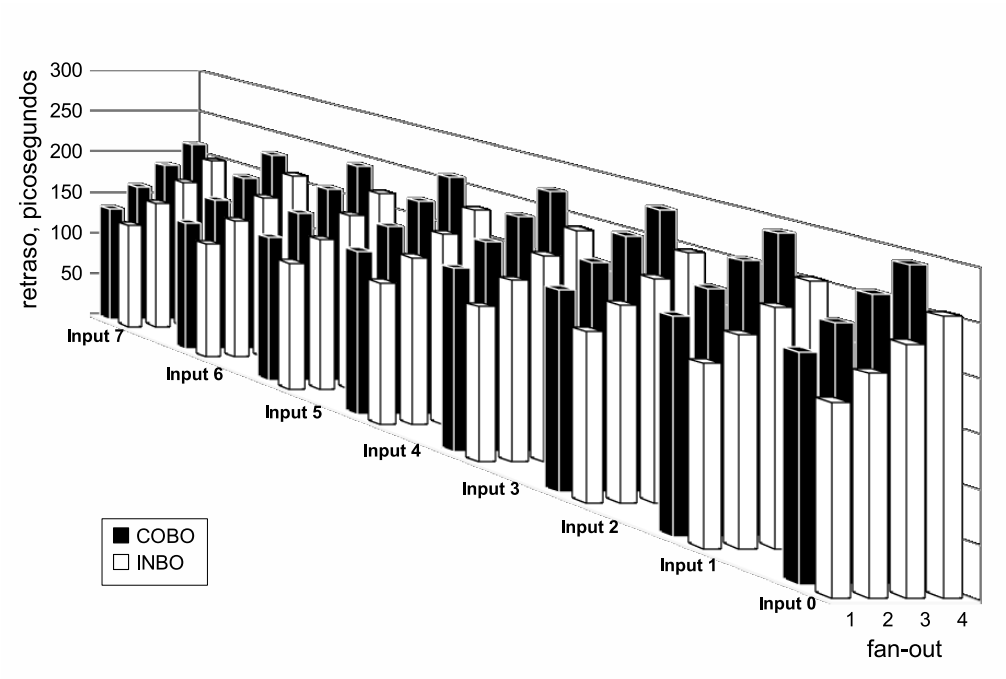



Figura 3.15: Retrasos de bajada de las puertas NAND de ocho entradas.

#### 3.2.2.3. Consumo dinámico

La energía consumida por la puerta al recibir un pulso en una de las entradas se muestra en las figuras 3.16 y 3.17 como una función de la carga de salida. El ahorro en consumo dinámico usando la implementación INBO es mayor que en el caso de las puertas de cuatro entradas: para carga unitaria, la energía consumida cuando un pulso se aplica a la entrada 0 (el caso de mayor consumo) se reduce un 24 % en las puertas NOR y un 17 % en las puertas NAND.

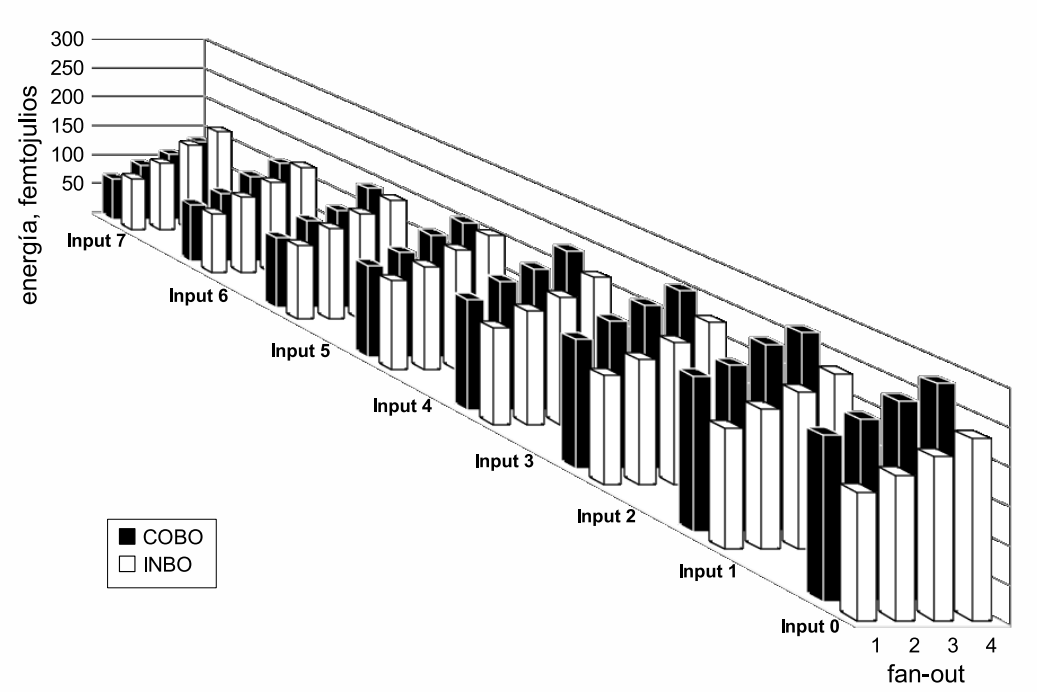



Figura 3.16: Consumo dinámico de las puertas NOR de ocho entradas.

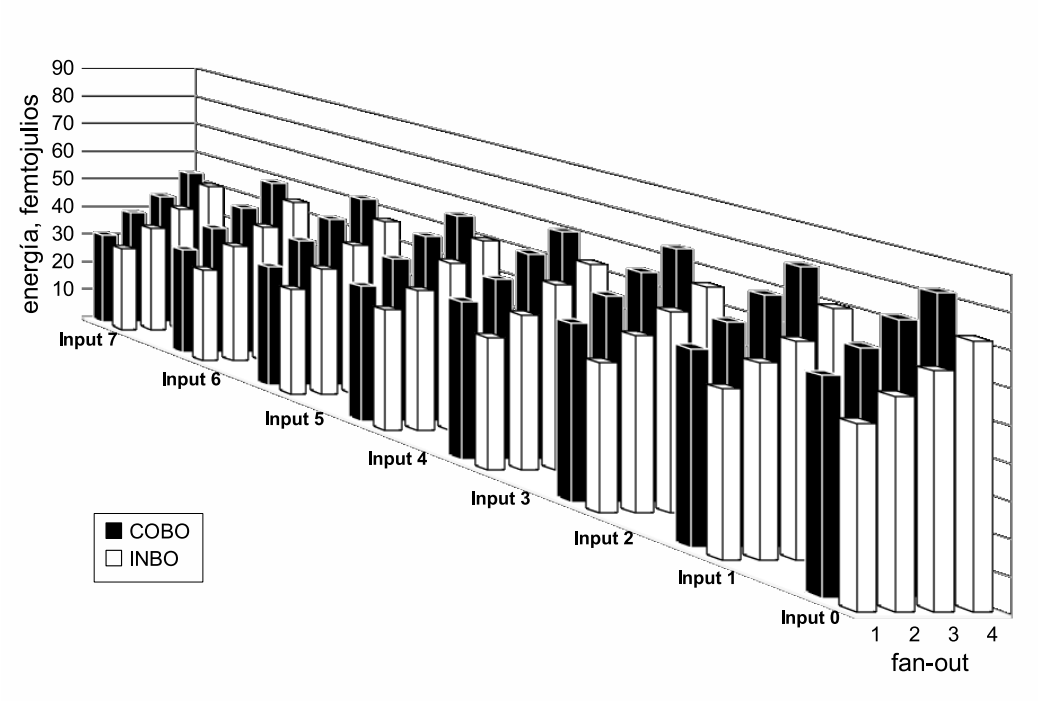



Figura 3.17: Consumo dinámico de las puertas NAND de ocho entradas.

#### 3.2.2.4. Producto energía-retraso

El producto energía-retraso (EDP) de ambas implementaciones se ilustra en las figuras 3.18 y 3.19. Al igual que con las puertas de cuatro entradas, el producto se ha calculado eligiendo el peor retraso de cada entrada (de subida o bajada) y multiplicándolo por la energía consumida cuando se aplica un pulso a dicha entrada. De nuevo, el uso de la implementación INBO reduce el EDP en casi todos los casos, siendo mayor la reducción en las entradas de índice menor (que presentan mayor EDP): la entrada 0 de la puerta NOR ve reducido su EDP entre un 59 % y un 48 % dependiendo del *fan-out* mientras que en la puerta NAND se reduce entre un 50 % y un 45 %.

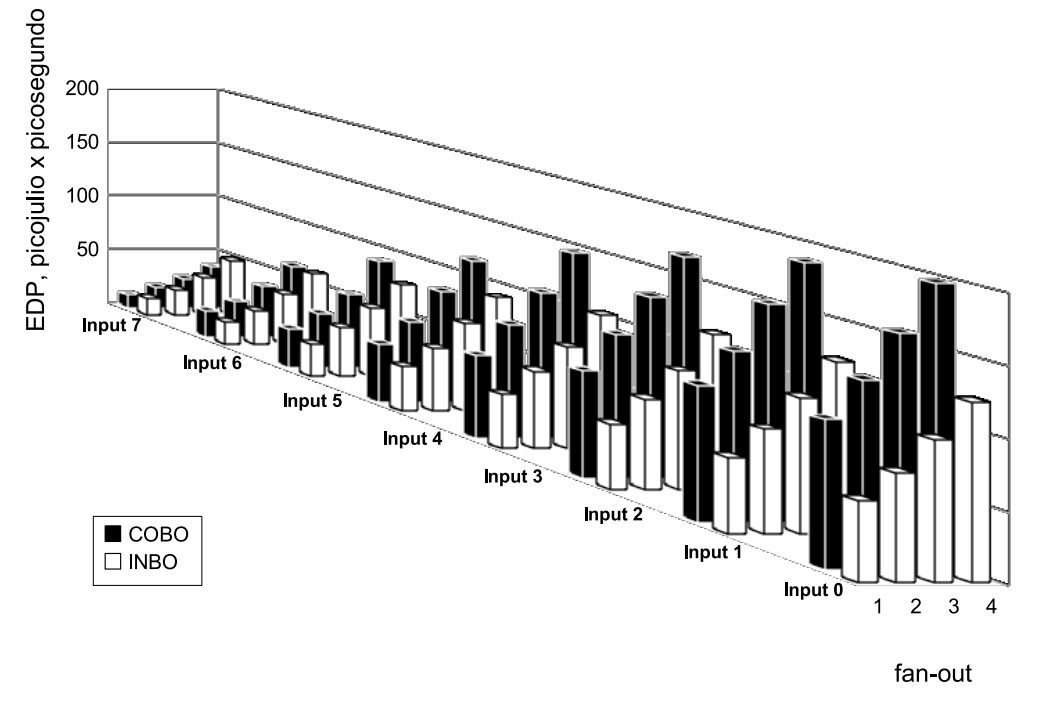



Figura 3.18: Producto energía retraso de las puertas NOR de ocho entradas.

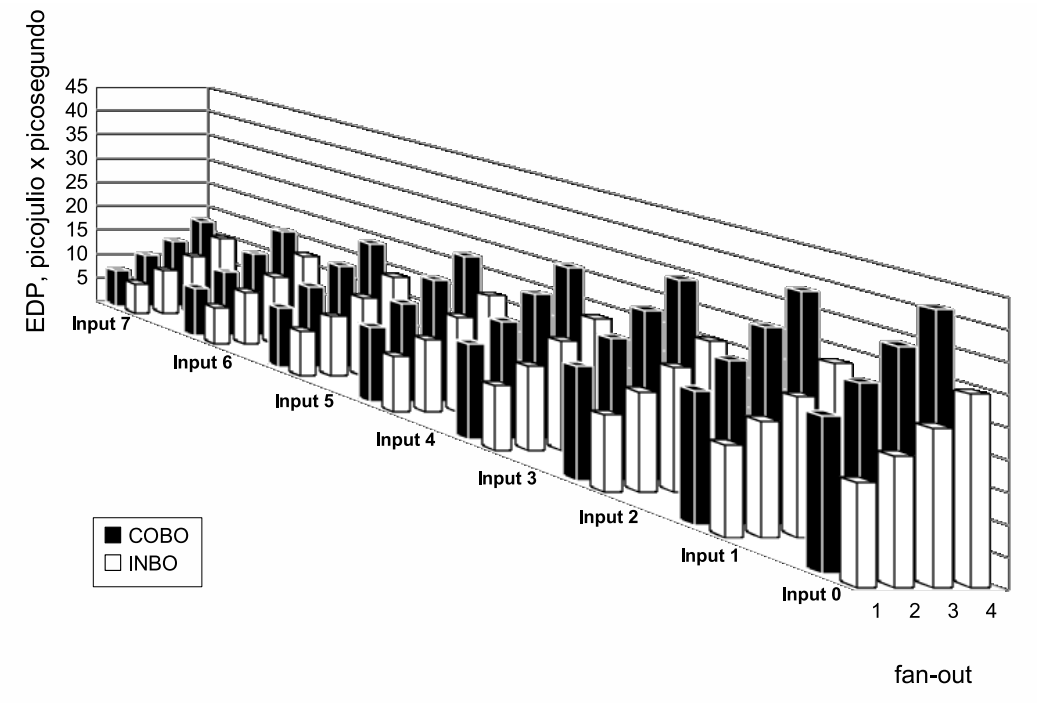



Figura 3.19: Producto energía retraso de las puertas NAND de ocho entradas.

#### 3.2.2.5. Consumo estático

Al igual que se hizo con las puertas de cuatro entradas, se ha medido el consumo estático para todos los posibles casos de entrada por simulación eléctrica. Los vectores de entrada se han numerado siguiendo el criterio empleado con las puertas de cuatro entradas, es decir, al vector de entrada

$$(I_7, I_6, I_5, I_4, I_3, I_2, I_1, I_o)$$

se le asigna el índice

$$I_7 * 2^7 + I_6 * 2^6 + I_5 * 2^5 + I_4 * 2^4 + I_3 * 2^3 + I_2 * 2^2 + I_1 * 2^1 + I_o * 2^0.$$

El número de vectores de entrada es ahora mucho mayor (256 en total), lo que ha obligado a dividir la representación de los resultados en distintas figuras para su análisis. En concreto se han agrupado en figuras los consumos estáticos correspondientes a vectores con el mismo número de ceros y unos, es decir, vectores para los hay el mismo número de transistores serie activos. Estas figuras así como las tablas con el valor numérico de los consumos pueden consultarse en el apéndice C.2.4.

Respecto al consumo estático cuando el árbol serie está conduciendo, una vez más el comportamiento en las implementaciones COBO e INBO es prácticamente el mismo al coincidir la tensión de los terminales *body* de los transistores en ambas implementaciones tal y como muestra la tabla 3.4. Esta circunstancia se da cuando la puerta NAND recibe el vector 256 y cuando la puerta NOR recibe el vector 0.

| | NAND | NOR |
|---|---|---|
| vector de entrada | 255 | 0 |
| consumo estático en el caso COBO | 159 pW | 967 pW |
| consumo estático en el caso INBO | 152 pW | 968 pW |

Tabla 3.4: Consumo estático de las puertas de ocho entradas cuando todos los transistores del árbol serie están activos.

Por otro lado, cuando el árbol serie no está activo la impedancia introducida por el elevado número de transistores serie reduce el consumo

por conducción subumbral, incrementándose la importancia relativa de las corrientes de polarización inversa. En este caso puede observarse que el consumo estático en la implementación INBO es menor que en la implementación COBO para todos los vectores de entrada. Las tablas 3.5, 3.6, 3.7 y 3.8 resumen el consumo estático medio y la varianza de este consumo para cada grupo de vectores en cada puerta. Estos consumos medios de ambas implementaciones se comparan en las figuras 3.20 y 3.21. Como en el caso de cuatro entradas, puede verse claramente que el consumo estático aumenta con el número de transistores activos en el árbol serie debido a la conducción subumbral. Las figuras 3.22 y 3.23 muestran el consumo medio de cada grupo ponderado por el número de vectores que lo integran: considerando todos los vectores de entrada posibles con la misma probabilidad, el consumo medio en estática de la puerta NOR se reduce de 17,14 a 10,74 pW, es decir, un 37 %. En el caso de la NAND el consumo se reduce de 21,58 a 7,01 pW (un 68 %).

| entradas a 1 | número de vectores | consumo medio | desviación estándar |
|---|---|---|---|
| 1 | 8 | 41,37 pW | 9,23 pW |
| 2 | 28 | 21,38 pW | 10,26 pW |
| 3 | 56 | 15,21 pW | 8,09 pW |
| 4 | 70 | 11,73 pW | 5,27 pW |
| 5 | 56 | 9,54 pW | 3,77 pW |
| 6 | 28 | 7,99 pW | 2,61pW |
| 7 | 8 | 6,86 pW | 1,59 pW |
| 8 | 1 | 6,05 pW | 0 pW |

Tabla 3.5: Consumo estático medio de cada grupo de vectores cuando el árbol serie no está activo para las puertas NOR COBO de ocho entradas.

| entradas a 1 | número de vectores | consumo medio | desviación estándar |
|---|---|---|---|
| 1 | 8 | 26,64 pW | 11,59 pW |
| 2 | 28 | 10,42 pW | 5,23 pW |
| 3 | 56 | 7,43 pW | 4,01 pW |
| 4 | 70 | 5,95 pW | 1,79 pW |
| 5 | 56 | 5,08 pW | 1,14 pW |
| 6 | 28 | 4,44 pW | 0,83 pW |
| 7 | 8 | 3,90 pW | 0,55 pW |
| 8 | 1 | 3,31 pW | 0 pW |

Tabla 3.6: Consumo estático medio de cada grupo de vectores cuando el árbol serie no está activo para las puertas NOR INBO de ocho entradas.

| entradas a 1 | número de vectores | consumo medio | desviación estándar |
|---|---|---|---|
| 0 | 1 | 12,63 pW | 0 pW |
| 1 | 8 | 13,24 pW | 1,81 pW |
| 2 | 28 | 14,26 pW | 2,89 pW |
| 3 | 56 | 15,73 pW | 4,05 pW |
| 4 | 70 | 18,08 pW | 5,57 pW |
| 5 | 56 | 23,37 pW | 10,91 pW |
| 6 | 28 | 33,10 pW | 27,68 pW |
| 7 | 8 | 58,23 pW | 14,67 pW |

Tabla 3.7: Consumo estático medio de cada grupo de vectores cuando el árbol serie no está activo para las puertas NAND COBO de ocho entradas.

| entradas a 1 | número de vectores | consumo medio | desviación estándar |
|---|---|---|---|
| 0 | 1 | 1,68 pW | 0 pW |
| 1 | 8 | 2,07 pW | 0,18 pW |
| 2 | 28 | 2,55 pW | 0,32 pW |
| 3 | 56 | 3,19 pW | 0,64 pW |
| 4 | 70 | 4,15 pW | 1,28 pW |
| 5 | 56 | 7,46 pW | 8,89 pW |
| 6 | 28 | 14,01 pW | 27,74 pW |
| 7 | 8 | 34,08 pW | 11,20 pW |

Tabla 3.8: Consumo estático medio de cada grupo de vectores cuando el árbol serie no está activo para las puertas NAND INBO de ocho entradas.

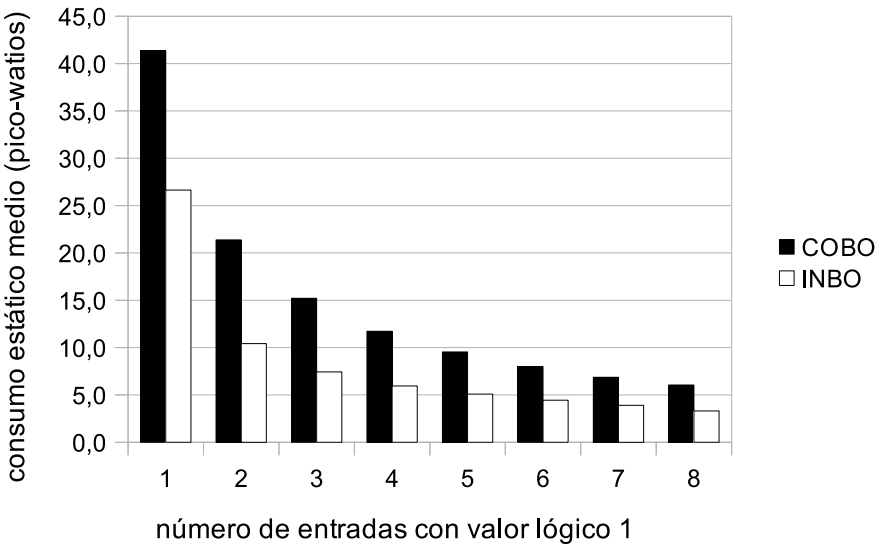



Figura 3.20: Consumo medio estático de las puertas NOR de ocho entradas cuando el árbol de transistores serie no está activo.

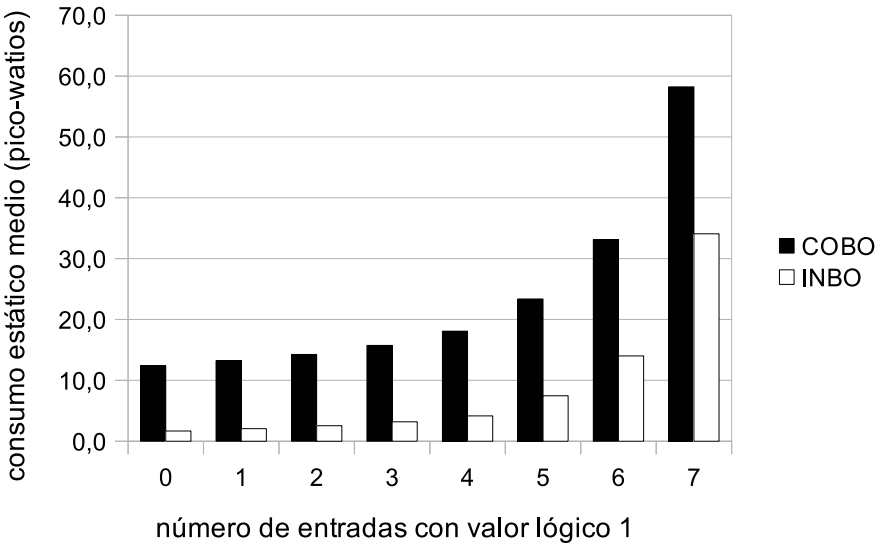



Figura 3.21: Consumo medio estático de las puertas NAND de ocho entradas cuando el árbol de transistores serie no está activo.

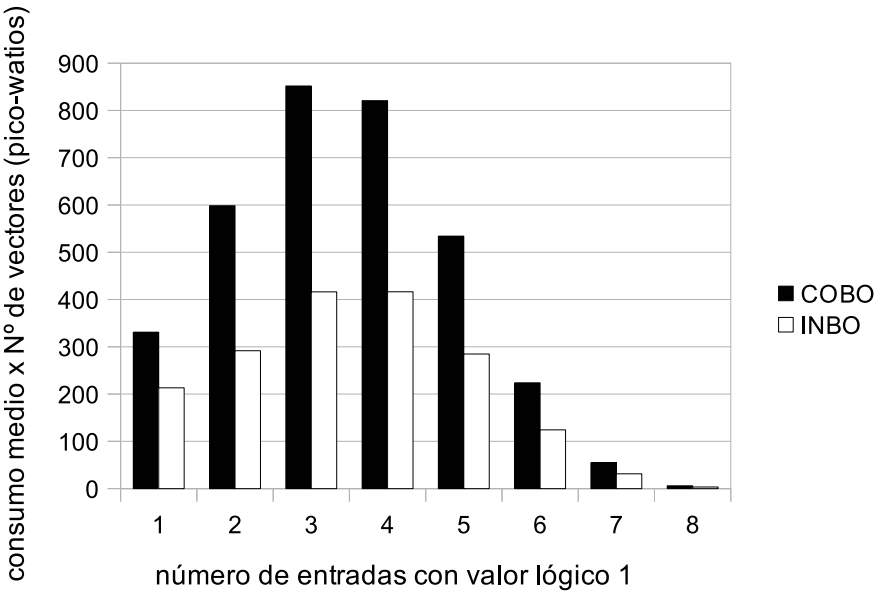



Figura 3.22: Consumo estático ponderado de las puertas NOR de ocho entradas cuando el árbol de transistores serie no está activo.

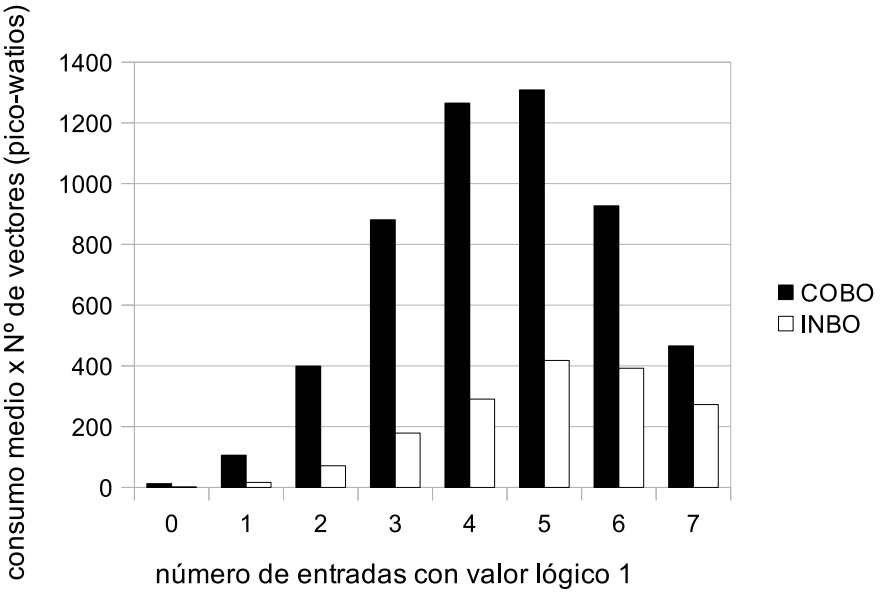



Figura 3.23: Consumo estático ponderado de las puertas NAND de ocho entradas cuando el árbol de transistores serie no está activo.

Las tablas 3.5, 3.6, 3.7 y 3.8 también muestran que la variación del consumo estático dentro de cada grupo es menor en las puertas INBO que en las COBO. La razón es la misma que en el caso de cuatro entradas: en la implementación COBO el consumo por polarización inversa dentro de cada figura está determinado por el número de diodos parásitos de los terminales drenador/fuente que se encuentren en polarización inversa, lo que a su vez está determinado por el número de transistores del árbol serie que están conectados al nodo de salida por un camino activo. En la implementación INBO, los diodos parásitos correspondientes a las uniones drenador/*body* no pueden encontrarse en polarización inversa ya que los pozos no están polarizados (excepto los de los transistores conectados a la entrada 0). Esta es la razón por la que el consumo estático de la implementación INBO es mucho menor y varía menos dentro de cada grupo de vectores.

## 3.3. Comparación de puertas con la misma área

Como se mostró en la sección 3.2, la implementación de una puerta INBO requiere un área considerablemente mayor que una puerta COBO homóloga con la misma geometría de transistores. Se podría argumentar que la anchura de los transistores de una puerta COBO podría aumentarse hasta que el área de la puerta alcanzase el de su equivalente INBO para mejorar su conductancia. No obstante esto aumentaría la conducción subumbral así como las corrientes de polarización inversa y, en consecuencia, el consumo estático. Además, al sobredimensionar los transistores aumentaría la capacidad parásita asociada a las entradas de la puerta de modo que las puertas que alimentasen dichas entradas serían más lentas y consumirían más energía. Esto no ocurre si se opta por la implementación INBO: la capacidad de entrada de una puerta INBO no aumenta con respecto a su equivalente COBO con transistores de la misma geometría, de modo que las puertas que generan sus entradas no se ven afectadas de forma negativa. Para ilustrar esto se ha simulado el funcionamiento de una puerta NOR COBO de cuatro entradas (figura B.9) del mismo tamaño de la puerta NOR INBO anteriormente presentada (figura B.4) y se han comparado sus prestaciones.

La puerta se ha implementado usando transistores de longitud mínima y anchura 1440 nm para los transistores N y 6140 nm para los transistores P de forma que los retrasos de subida y bajada sean similares y el área total ocupada sea prácticamente idéntica a la alternativa INBO descrita en el apartado 3.2.1 y que usaba transistores N de 240 nm de anchura y transistores P de 1440 nm de anchura. En los apartados siguientes se hará un análisis comparativo del retraso y el consumo de las puertas mencionadas.

### 3.3.1. Retraso pin a pin

Los transistores sobredimensionados de la implementación COBO aceleran la carga y descarga de la capacidad de salida, de modo que hay un notable aumento de velocidad respecto a la puerta NOR COBO original. No obstante, aumentar el tamaño de los transistores implica incrementar las capacidades parásitas. Además también se aumenta la capacidad de salida ya que en el circuito de pruebas la puerta bajo test genera la entrada de una puerta similar (figura 3.1). Por ello la puerta COBO sobredimensionada sigue siendo más lenta que la puerta INBO del mismo tamaño. Las tablas 3.9 y 3.10 recogen los retrasos pin a pin de la puerta sobredimensionada en función de la carga de salida. Los retrasos de la puerta sobredimensionada se comparan con su homóloga INBO en las figuras 3.24 y 3.25.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 173 ps | 161 ps | 132 ps | 88 ps |
| 2 | 210 ps | 198 ps | 169 ps | 122 ps |
| 3 | 249 ps | 236 ps | 207 ps | 158 ps |
| 4 | 288 ps | 276 ps | 246 ps | 198 ps |

Tabla 3.9: Retraso de subida de la puerta NOR COBO de cuatro entradas sobredimensionada.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 133 ps | 127 ps | 112 ps | 84 ps |
| 2 | 149 ps | 143 ps | 128 ps | 101 ps |
| 3 | 165 ps | 158 ps | 143 ps | 117 ps |
| 4 | 180 ps | 173 ps | 157 ps | 132 ps |

Tabla 3.10: Retraso de bajada de la puerta NOR COBO de cuatro entradas sobredimensionada.

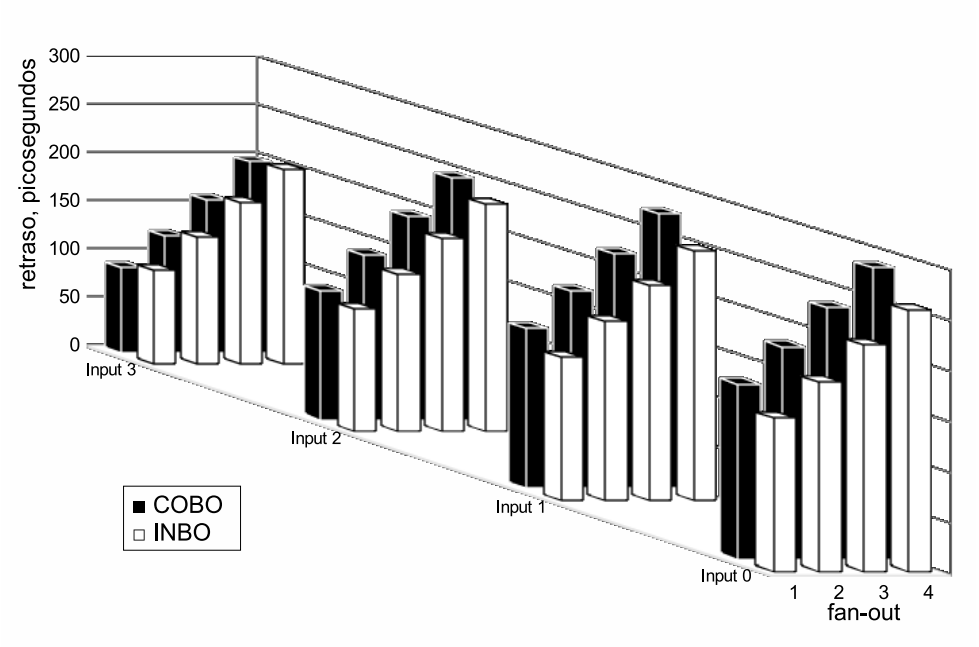



Figura 3.24: Retrasos de subida de la puerta NOR INBO y de la NOR COBO sobredimensionada.

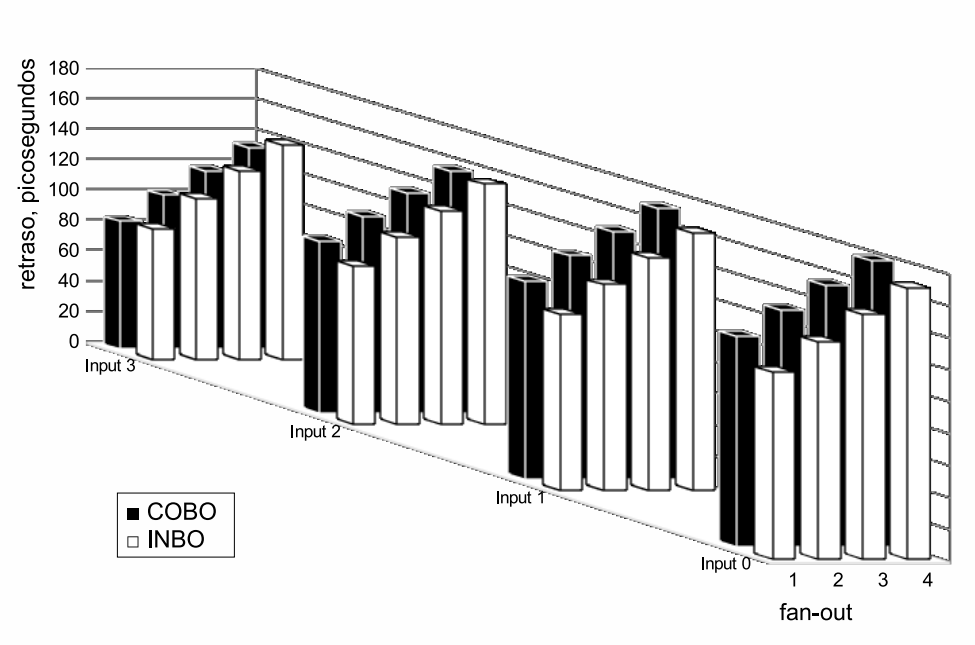



Figura 3.25: Retrasos de bajada de la puerta NOR INBO y de la NOR COBO sobredimensionada.

Puede observarse que el retraso de la puerta COBO es mayor para casi cualquier entrada: bajo *fan-out* unitario el retraso de bajada aumenta de 117 a 133 picosegundos (13.6 %) al usar la puerta COBO, mientras que el retraso de subida aumenta de 151 a 174 picosegundos (14.5 %). Solo en el caso de la entrada de menor retraso, I3, la puerta COBO es ligeramente más rápida.

### 3.3.2. Consumo dinámico

La tabla 3.11 muestra la energía consumida por la puerta NOR COBO sobredimensionada cuando se aplica un pulso a cada una de sus entradas. En la figura 3.26 se compara este consumo con el de la puerta INBO del mismo tamaño. Como se mencionó, la mayor capacidad parásita de los transistores sobredimensionados implica un mayor consumo que en el caso INBO. El consumo de la puerta COBO es notablemente mayor para cualquier entrada y bajo cualquier *fan-out*. Por ejemplo, para la entrada de mayor consumo la versión COBO consume 276 fJ bajo *fan-out* unitario frente a los 56 fJ de la versión INBO, esto es, el consumo dinámico del la puerta COBO es un 393 %

mayor que el de la INBO cuando ambas puertas se diseñan para ocupar el mismo área.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 276 fJ | 245 fJ | 198 fJ | 156 fJ |
| 2 | 333 fJ | 302 fJ | 262 fJ | 199 fJ |
| 3 | 383 fJ | 355 fJ | 311 fJ | 244 fJ |
| 4 | 443 fJ | 416 fJ | 374 fJ | 300 fJ |

Tabla 3.11: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NOR COBO de cuatro entradas sobredimensionada.

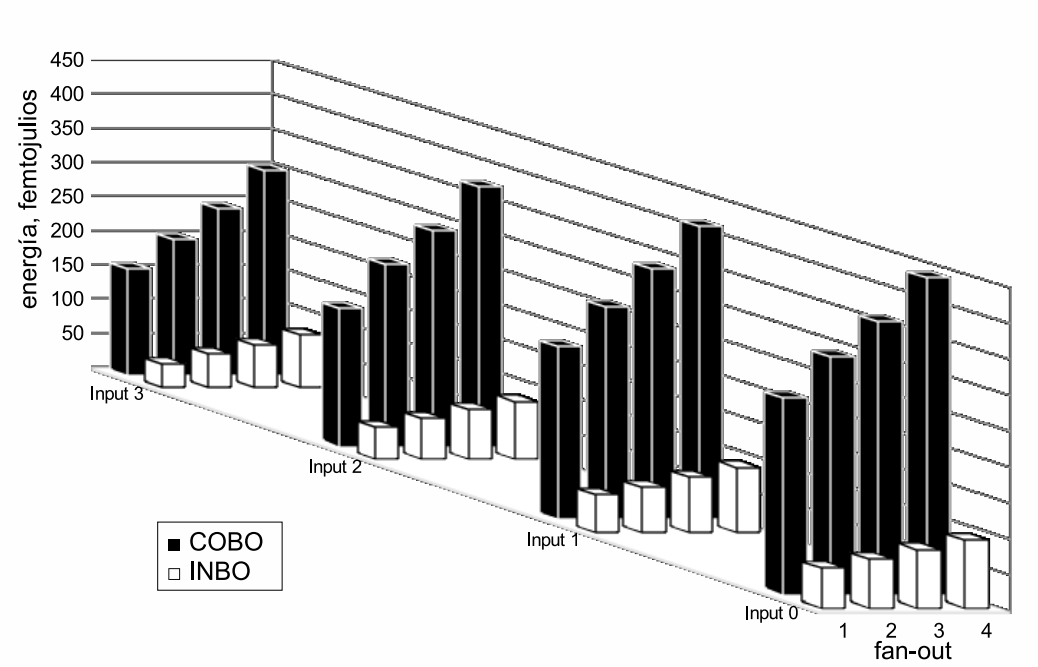



Figura 3.26: Consumo dinámico de la puerta NOR INBO y de la NOR COBO sobredimensionada.

### 3.3.3. Producto energía-retraso

De nuevo, el producto energía retraso se ha calculado eligiendo el peor retraso de cada entrada (el de subida o el de bajada) y multiplicándolo por la energía consumida cuando un pulso se aplica a dicha entrada. Estos

productos se muestran en la tabla 3.12 y se comparan con los de la puerta INBO en la figura 3.27. El producto es notablemente mayor en la puerta sobredimensionada para cualquier entrada y bajo cualquier *fan-out*. Bajo *fan-out* unitario el mayor de los productos pasa de 8.49 pJ×pS a 47.89 pJ×pS al usar la puerta sobredimensionada, lo que supone un aumento del 464 % respecto a la puerta INBO.

| | E.D.P. (pJ×pS) | | | |
|---|---|---|---|---|
| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
| 1 | 47,89 | 39,44 | 26,12 | 13,79 |
| 2 | 70,01 | 59,73 | 44,15 | 24,22 |
| 3 | 95,17 | 83,69 | 64,27 | 38,59 |
| 4 | 127,69 | 114,57 | 92,04 | 59,23 |

Tabla 3.12: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NOR COBO de cuatro entradas sobredimensionada.

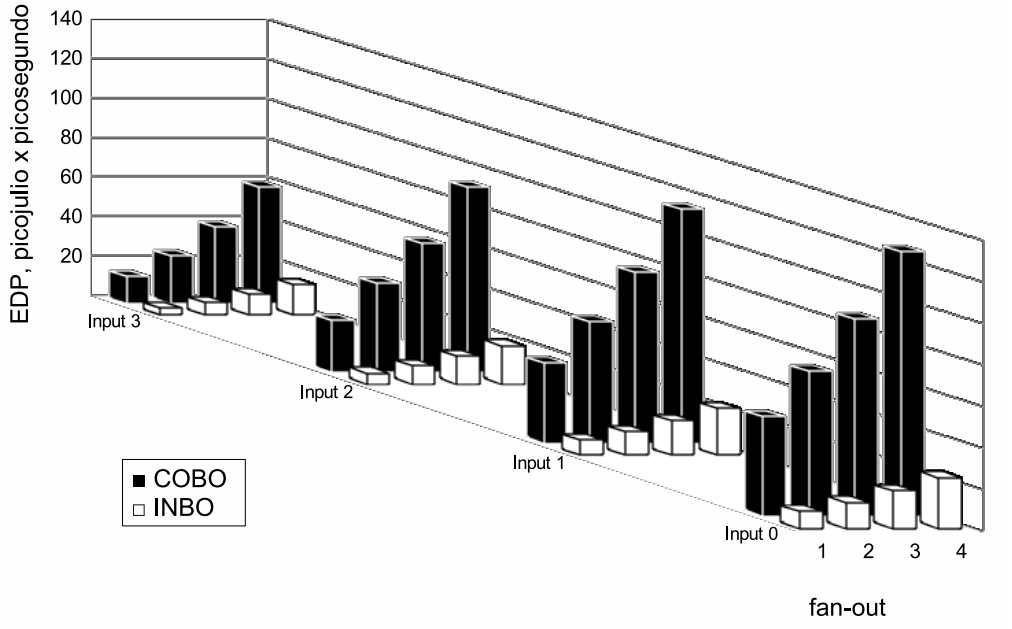



Figura 3.27: Producto energía retraso de las puertas NOR INBO y de la NOR COBO sobredimensionada.

### 3.3.4. Consumo estático

El consumo estático de la puerta NOR COBO sobredimensionada para cada vector de entrada se muestra en la tabla 3.13. Aumentar la anchura de los transistores implica aumentar la conducción subumbral, la cual es una fuente importante de consumo en estática. De forma análoga a como se hizo en la sección 3.2.1.5, el consumo estático de la puerta NOR COBO sobredimensionada cuando el árbol serie está cortado se compara con el de su homólogo INBO en la figura 3.28. Como puede observarse, el consumo estático de la puerta COBO sobredimensionada puede aumentar hasta un 300 % respecto a la puerta INBO para ciertos vectores de entrada.

| vector de entrada | consumo COBO |
|---|---|
| 0 | 108,0 pW |
| 1 | 85,1 pW |
| 2 | 78,6 pW |
| 3 | 75,3 pW |
| 4 | 78,6 pW |
| 5 | 36,0 pW |
| 6 | 29,5 pW |
| 7 | 29,5 pW |
| 8 | 26,2 pW |
| 9 | 26,2 pW |
| 10 | 22,9 pW |
| 11 | 19,6 pW |
| 12 | 19,6 pW |
| 13 | 19,6 pW |
| 14 | 19,6 pW |
| 15 | 13,1 pW |

Tabla 3.13: Consumo estático de las puertas NOR de cuatro entradas.

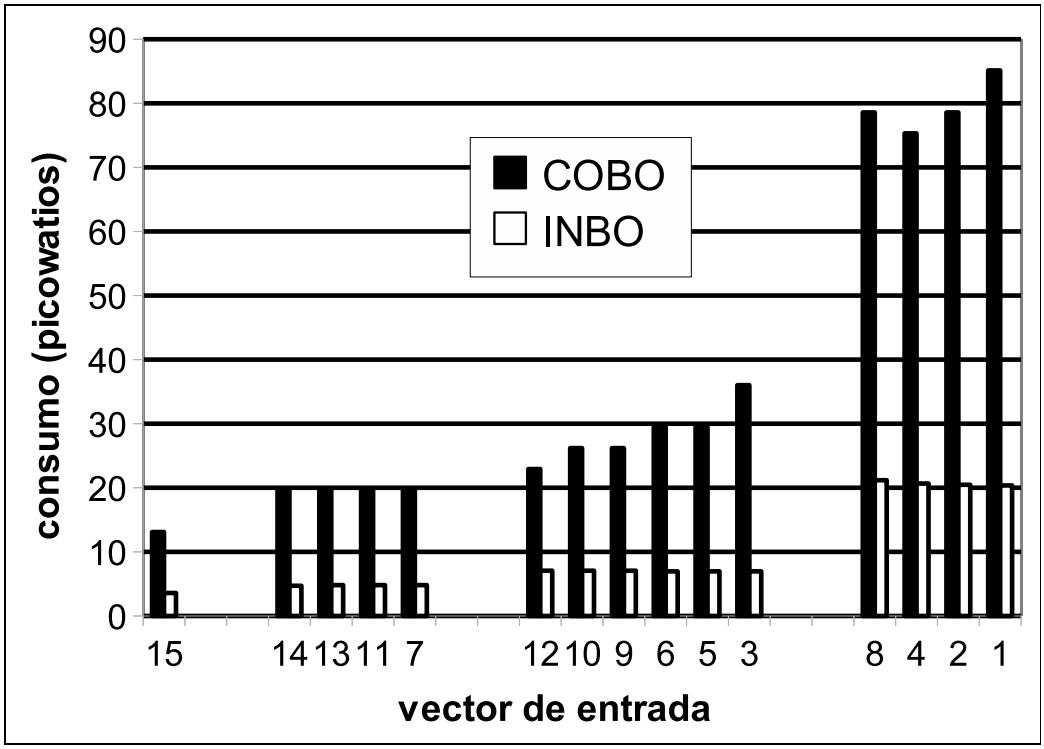



Figura 3.28: Consumo estático de la puerta NOR COBO de cuatro entradas sobredimensionada y de su homóloga INBO cuando el árbol de transistores serie está cortado.

## 3.4. Resumen

Se ha analizado la variante al diseño de puertas *bulk*-CMOS complementarias consistente en emplear transistores con los terminales de fuente y *body* conectados. Las tecnologías convencionales de fabricación CMOS actuales lo hacen posible al permitir el uso de múltiples pozos. Se han realizado simulaciones eléctricas de puertas NAND y NOR tanto de cuatro entradas como de ocho entradas diseñadas de acuerdo al estilo tradicional (COBO) y al propuesto (INBO) para obtener datos de retraso, consumo estático y consumo dinámico.

Claramente, la alternativa INBO requiere un área muy superior a la COBO para transistores de la misma geometría. No obstante, las puertas INBO obtienen mejores resultados en todos los aspectos analizados,

incluyendo un menor consumo estático, un menor consumo estático, un menor y más homogéneo retraso pin-a-pin y, en consecuencia, un menor producto energía-retraso (EDP), con reducciones de entre el 40 % y el 50 %.

Se ha mostrado que un diseño COBO tradicional sobredimensionado hasta igualar el área de la alternativa INBO mejora la conductividad de sus transistores respecto a los de la puerta COBO original. No obstante, sus prestaciones siguen siendo sensiblemente inferiores a las de la puerta INBO de la misma área cuando se aplica globalmente ya que el sobredimensionado implica un aumento en las corrientes parásitas así como en las capacidades de entrada, lo que afecta negativamente a las prestaciones. En el caso ilustrado, el producto energía-retraso de la versión COBO puede ser hasta cinco veces mayor que el de la versión INBO y su consumo estático puede ser hasta cuatro veces mayor.

Los resultados obtenidos indican que la alternativa INBO tiene aplicación en escenarios donde el área no sea importante frente a otras restricciones como las enumeradas a continuación:

- Necesidad de reducir el retraso sin aumentar las capacidades de los nodos como, por ejemplo, en los caminos críticos de los bloques combinacionales.
- Necesidad de tener un retraso más uniforme entre las distintas entradas de una puerta.
- Necesidad de implementar puertas de un gran número de entradas usando pocos niveles lógicos.
- Necesidad de reducir el consumo estático y/o dinámico en partes críticas del circuito, por ejemplo en aquellas con alta frecuencia de conmutación.

# Capítulo 4

# Esquemas de temporización en circuitos síncronos

Como se mencionó en el apartado 1.1, los circuitos secuenciales son un tipo de circuito digital en los que se computan iterativamente funciones booleanas realimentando la entrada con salidas previamente computadas. El funcionamiento del circuito debe planificarse de forma que el cómputo de un valor de salida no se inicie hasta que la entrada correspondiente no esté disponible. Los circuitos secuenciales síncronos, que son los más ampliamente utilizados, llevan a cabo este control temporal mediante señales de reloj periódicas. Tal y como se mostró en la figura 1.1, los CSS constan de elementos combinacionales que computan funciones booleanas, biestables que codifican el estado del sistema y líneas conductoras que propagan las señales de control y la información entre los distintos componentes. La importancia relativa del retraso de estas líneas crece conforme mejoran las tecnologías de fabricación de transistores. Esto es debido a que disminuye el retraso de las puertas lógicas, y al mismo tiempo, aumenta la densidad de integración, con lo que la longitud de las líneas crece respecto al tamaño de los componentes que interconectan y son, en comparación, más lentas. Sin embargo, lo realmente problemático es que las diferencias entre los retrasos de las líneas alcanzan magnitudes comparables a las de los retrasos de los elementos que interconectan. En particular, las diferencias entre los

retrasos de las líneas que distribuyen las señales de reloj pueden producir problemas de funcionamiento en el sistema si no son debidamente tomadas en cuenta durante su diseño. En este capítulo estudiaremos estos problemas y se plantearan distintas alternativas para solucionarlos.

## 4.1. Problemas de sincronización

El aumento de la densidad de integración y del tamaño de la superficie activa útil de los *chips* hace que la distribución de las señales de reloj se haya convertido en un problema crítico. En esta situación, cualquier diseñador moderno tiene que conocer la problemática asociada al incumplimiento de las condiciones temporales de buena operación de los circuitos provocados por problemas de sincronización. Con el término *problemas de sincronización* se alude a los fallos y errores causados por la mala temporización de las señales en los sistemas digitales con memoria, suponiendo que tanto el diseño lógico como la operación de los distintos componentes del sistema son correctos. En último término, la raíz de estos problemas está en la ocurrencia de cambios en señales que llegan a uno o más elementos de memoria en un momento inadecuado, causando una operación errónea de éstos.

La generación y distribución de las señales de reloj implica afrontar problemas complejos [28, 29]. Uno de ellos es el *cross-talk*: la red de distribución de reloj actúa como una línea de transmisión, de modo que puede ser interferida por ella misma o por otras señales del sistema. Además no es posible controlar completamente el instante en el que se aplican los flancos de las señales de reloj a los distintos elementos de memoria. Esta falta de precisión viene determinada principalmente por dos factores [30]: La desviación de reloj causada por la distribución de la señal de reloj por el sistema, conocida habitualmente como *clock skew*, término que usaremos a partir de ahora para referirnos a este fenómeno, y la variación o imprecisión causada por el propio circuito de generación de reloj, conocida como *clock jitter*. La incertidumbre en la llegada del flanco de reloj es la suma del tiempo de *clock skew* y el de *clock jitter*.

Aunque depende significativamente de la frecuencia de operación del

sistema, la imprecisión del reloj está generalmente en torno al 10 % del periodo de reloj [30]. En cualquier caso, debe remarcarse que los efectos de la imprecisión en el reloj son altamente dependientes del estilo de diseño del circuito, especialmente de sus elementos de memoria, así como del esquema de reloj empleado. En diseño con *flip-flops*, por ejemplo, si el flanco activo llega demasiado pronto, los biestables pueden capturar valores que aún no son correctos, produciéndose un fallo de funcionamiento. En circuitos dinámicos, si el flanco de reloj que marca el comienzo de la fase de evaluación llega antes de que termine la precarga de alguna puerta, la salida de dicha puerta puede dar un valor de cero de forma incorrecta. Estos dos ejemplos de mal funcionamiento son fácilmente detectables y a veces pueden evitarse reduciendo la frecuencia de reloj, aunque esto disminuya el rendimiento del sistema. Otros efectos perniciosos de la falta de precisión de la señal de reloj no pueden detectarse ni corregirse tan fácilmente.

Como hemos mencionado, la imprecisión del reloj tiene dos componentes: El *jitter* proviene del generador de la señal de reloj, mientras que el *skew* es propio de la ruta de distribución de reloj del sistema. Comparado con el *clock skew*, el *clock jitter* es relativamente inocuo: si la imprecisión en la frecuencia del reloj se subestima y causa mal funcionamiento, siempre es posible disminuir la frecuencia nominal del generador de reloj para garantizar una cota superior en la frecuencia real y hacer que el sistema funcione correctamente sin tener que rediseñarlo. Por el contrario, la aparición de un problema debido a una mala estimación del *clock skew* puede exigir rediseñar e implementar nuevamente el sistema. El presente trabajo está enfocado a los aspectos de diseño del sistema, por lo que nos vamos a centrar en el fenómeno del *clock skew*. En el siguiente apartado vamos a presentar más detalladamente en que consiste este fenómeno y vamos a realizar un análisis físico del mismo.

## 4.2. El fenómeno del desajuste de reloj o *clock skew*

El *clock skew* se produce porque los retrasos de los caminos que transportan la señal de reloj no son iguales entre sí. Cada segmento de uno de esos caminos tiene asociado una capacidad y una resistencia que determinan su contribución al retraso. Desgraciadamente, estas características dependen no solo de la longitud del segmento, sino también de los elementos conectados y de las características del propio entorno, por lo que no son predecibles. A continuación vamos a presentar un análisis físico sencillo que explica la existencia de este fenómeno.

### 4.2.1. Análisis físico del *clock skew*

El desajuste de reloj aparece cuando los retrasos introducidos por las líneas que transportan la señal son distintos para diferentes puntos del sistema. El problema aparece cuando dichos retrasos son lo suficientemente significativos como para desincronizar el sistema y causar, por lo tanto, un mal funcionamiento del mismo. Para explicar su origen vamos a considerar la figura 4.1. En la figura se presenta cómo se distribuye la señal de reloj $CK_{in}$ para controlar las distintas partes de un sistema. Para ello mostramos un modelo RC sencillo de las líneas. Según el modelo, las líneas que transportan el reloj están constituidas por un conjunto de resistencias y condensadores distribuidos. De esta forma las señales $CK_A$ y $CK_C$ van a sufrir retrasos distintos. Estos retrasos no son completamente predecibles ya que las variaciones en el proceso de fabricación impiden que puedan controlarse completamente las características físicas de las líneas. Además los retrasos no sólo dependen de las características físicas de las líneas, sino también de los elementos conectados y de las características propias del entorno que no son controlables (temperatura, fluctuaciones en la alimentación, etc.). Por lo tanto, observamos que aunque al plantear el diseño del sistema se haya considerado que las señales $CK_A$ y $CK_C$ tengan un mismo comportamiento, en la práctica dichas señales pueden estar desincronizadas entre sí.

Para establecer un estudio analítico del desajuste de reloj podemos considerar como una primera aproximación un modelo de las líneas de conexión como una red de resistencias y condensadores concentrados. Esto nos permite extraer expresiones de retrasos en las líneas y establecer los límites superior e inferior para el desajuste de reloj. Aunque este modelo suministra resultados excesivamente pesimistas para un cálculo preciso, es suficiente para nuestro ejemplo. Un estudio de estas características se hace en [31].

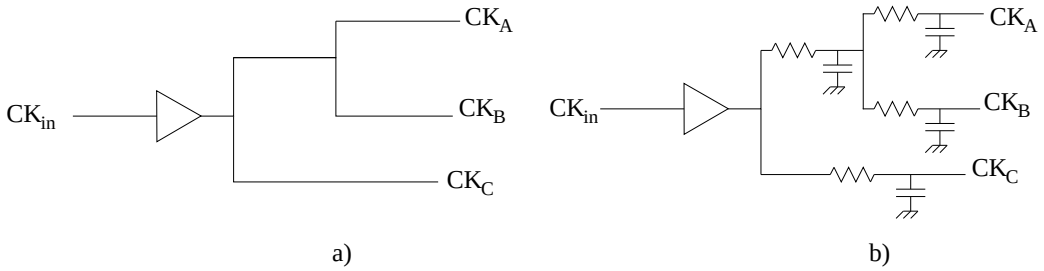



Figura 4.1: a) Ejemplo de distribución de reloj. b) Modelo RC de las líneas.

El modelo RC concentrado de una línea de conexión se muestra en la figura 4.2.a). El valor de la resistencia se calcula considerando la longitud y la anchura de la línea:

$$R = \frac{\rho}{h}\frac{L}{W} = R_s\frac{L}{W} \tag{4.2.1}$$

donde $\rho$ es la resistividad del material, $h$ es un parámetro tecnológico que corresponde al grosor de la línea y $L$ y $W$ son parámetros de diseño que se corresponden con la longitud y anchura de la línea. Los parámetros tecnológicos pueden expresarse en función de la constante $R_s$ , que representa la resistencia del material.

Para calcular el valor del condensador asociado a la línea de conexión podemos usar un modelo de dos placas paralelas. Una de las placas es la que soporta la señal (polisilicio o metal), mientras que la otra placa corresponde al sustrato. Basándose en este modelo, la expresión de la capacidad viene dada por:

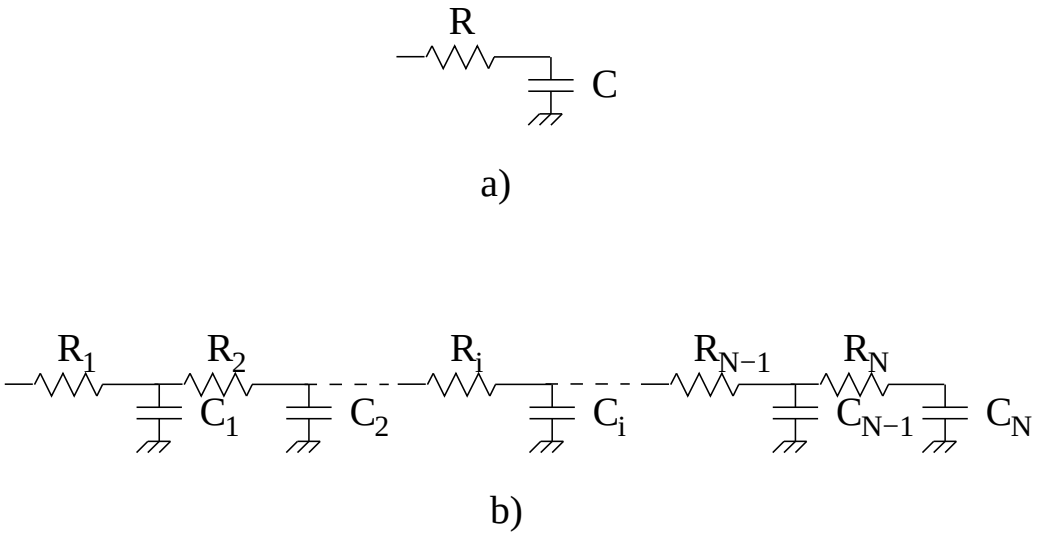



Figura 4.2: Modelado de una línea de conexión. a) Modelo RC concentrado. b) Modelo RC distribuido.

$$C = \frac{\varepsilon}{t_{ox}} LW \tag{4.2.2}$$

donde $\varepsilon$ es la permisividad del dieléctrico ($SiO_2$), $t_{ox}$ es el grosor del óxido y $L$ y $W$ son la longitud y anchura de línea respectivamente. Este modelo puede considerarse de primer orden ya que desprecia otros efectos capacitivos tales como los condensadores que se forman entre las distintas capas. Estos efectos de segundo orden pueden llegar a duplicar el valor obtenido en la expresión anterior.

Aplicando los valores anteriores al modelo $RC$ concentrado de la línea de conexión, la constante de tiempo $\tau$ que determina el retraso que introduce la línea viene dado por la siguiente expresión:

$$\tau = rcL^2 \tag{4.2.3}$$

donde $r$ y $c$ son la resistencia y capacidad por unidad de longitud.

Se observa que el retraso crece de forma cuadrática con la distancia. Este hecho hay que tenerlo en cuenta al establecer consideraciones de diseño en sistemas de alta velocidad.

Otro efecto que es necesario añadir en la capacidad de la línea es el valor de la capacidad de la carga conectada a la misma. Dicha carga es proporcional a la capacidad de puerta de los transistores MOS que actúan de carga, por lo que su valor será proporcional a $C_{ox}L_gW_g$, siendo $C_{ox}$ la capacidad de óxido fino y $L_gW_g$ representa el área de la puerta.

Un modelo más exacto de la línea es el modelo RC distribuido. En este caso la longitud de la línea se divide en $N$ segmentos. Cada segmento tiene su modelo RC como se muestra en la figura 4.2.b. El cálculo de la corriente de carga del condensador $i$ viene dado por la siguiente expresión:

$$I_i = C_i\frac{\partial V_i}{\partial t} = c\Delta L\frac{\partial V_i}{\partial t} = \frac{(V_{i+1} - V_i) - (V_i - V_{i-1})}{r\Delta L} \tag{4.2.4}$$

donde $\Delta L$ corresponde a la longitud del segmento. Si hacemos tender cada segmento a un tamaño infinitesimal se obtiene la ecuación de difusión:

$$rc\frac{\partial V}{\partial t} = \frac{\partial^2 V}{\partial x^2} \tag{4.2.5}$$

donde $V$ es la tensión en un punto de la línea y $x$ es la distancia entre ese punto y la fuente de señal. No existe ninguna solución cerrada de esta expresión, si bien se suelen utilizar expresiones aproximadas [31, 32]. Nuestro interés se centra en el cálculo del retraso de propagación. En este caso la solución viene dada por la expresión:

$$\tau = kx^2 \tag{4.2.6}$$

donde k es una constante. Una expresión alternativa fue dada por Elmore [33] para el circuito mostrado en la figura aplicando un modelo discreto:

$$\tau_N = \sum_{i=1}^{N} R_i \sum_{j=1}^{N} C_j = \sum_{j=1}^{N} C_j \sum_{i=1}^{N} R_i = rc(\Delta L)^2\left[\frac{N(N+1)}{2}\right] \tag{4.2.7}$$

Haciendo tender cada segmento a un tamaño infinitesimal se obtiene que:

$$\tau = \frac{rcL^2}{2} \tag{4.2.8}$$

De nuevo, como era previsible, se ha obtenido una función cuadrática del retraso en función de la longitud de la línea. Comparando la expresión obtenida con la del modelo RC concentrado (ecuación 4.2.3), se observa que los datos en el caso concentrado son más pesimistas que los del modelo distribuido (en un factor de 2).

Aunque estos análisis consideran una red de distribución lineal, puede hacerse un análisis similar sobre una red ramificada como la mostrada en 4.3 llegándose a conclusiones similares [34]: El retraso asociado a las líneas crece de forma cuadrática con su longitud. Esto hace que cualquier avance en densidad de integración tenga un gran impacto sobre la importancia relativa de estos retrasos.

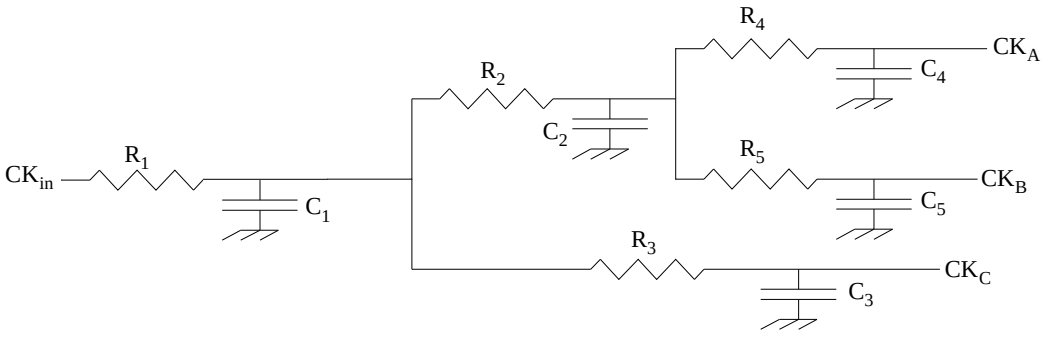



Figura 4.3: Ejemplo de un modelo de distribución del reloj.

## 4.3. Análisis de esquemas de sincronización existentes

A la hora de estudiar los sistemas controlados por reloj, es vital la elección de los elementos de memoria así como la del número de señales o fases de reloj. En este apartado vamos a analizar distintas técnicas de control con reloj (sistemas con una o dos fases de reloj). Nuestro interés se centrará en estudiar las restricciones que impone al diseñador el uso de una o varias fases de reloj en función del elemento de almacenamiento que se utilice.

### 4.3.1. Esquema de temporización de una fase de reloj basado en *flip-flops*

Dada su simplicidad, este esquema de reloj se ha usado y se sigue usando en multitud de diseños. Sin embargo, un *clock skew* severo puede hacerlo inadecuado en sistemas de alta velocidad [31, 35, 36, 37]. Esto se ilustra en la figura 4.4 . Del análisis de esta figura se puede obtener el periodo mínimo

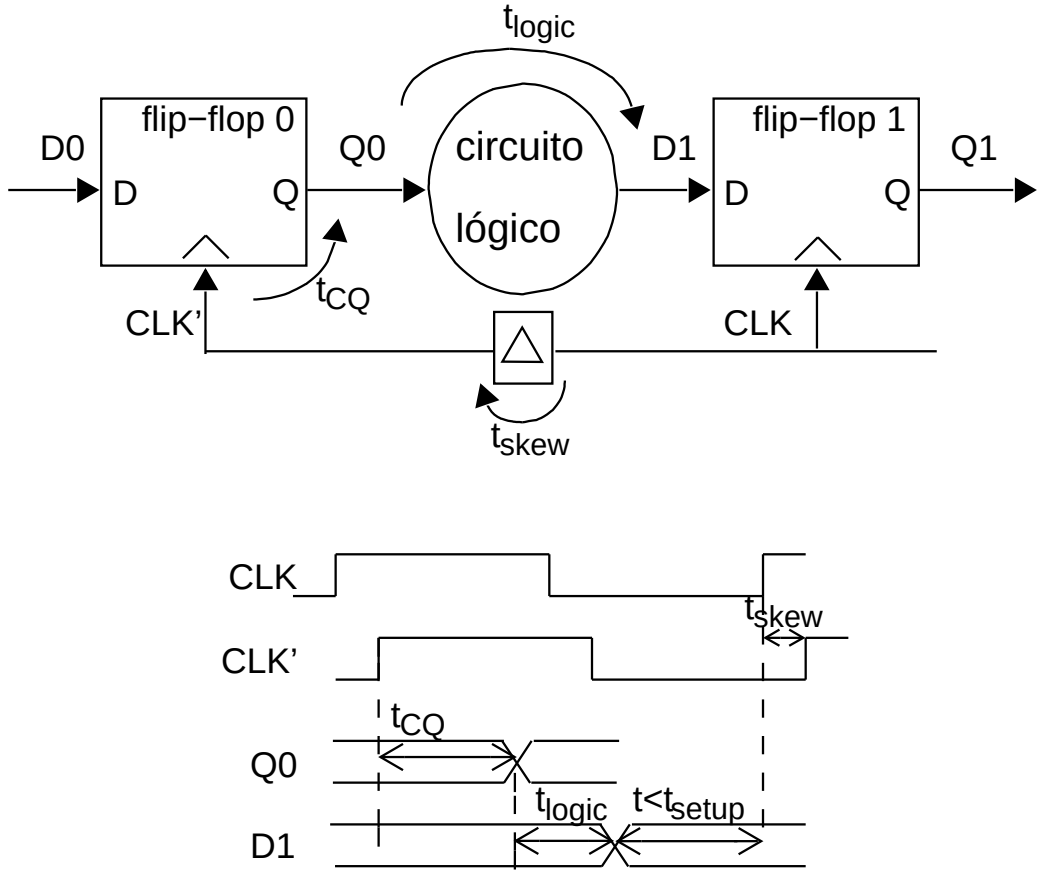



Figura 4.4: Violación de restricciones para el camino largo en un sistema de una sola fase con *flip-flops*.

(frecuencia máxima) para que el sistema opere correctamente (ecuación 4.3.1)

$$t_{clk} > t_{CQMax} + t_{logicMax} + t_{setup} + t_{skew} \quad (4.3.1)$$

En la ecuación 4.3.1, $t_{clk}$ es el periodo de reloj, $t_{flip-flopMax}$ es el retraso máximo del *flip-flop*, es decir, el tiempo máximo que transcurre entre el flanco activo de la señal de reloj hasta que aparece el dato correcto en la salida, $t_{logicMax}$ es el retraso máximo de la lógica combinacional, $t_{setup}$ es el tiempo que las entradas de los *flip-flops* deben estar correctas y estables antes del flanco activo para garantizar que el dato se captura de forma correcta y $t_{skew}$ es el valor máximo del *clock skew*. Es evidente que cuanto mayor es el *skew* mayor es la degradación de la velocidad de operación del sistema.

Por otra parte, para que el sistema opere correctamente se debe asegurar que las entradas del *flip-flop* permanezcan estables un tiempo igual o superior a $t_{hold}$ (ecuación 4.3.2).

$$t_{hold} < t_{CQMin} + t_{logicMin} - t_{skew} \quad (4.3.2)$$

Si esta condición no se cumple, puede ocurrir que la salida de un *flip-flop* haga cambiar de forma prematura la entrada de otro tal y como se ilustra en la figura 4.5. Una manera de evitar esto podría ser insertando *buffers* que añadan retraso a la lógica combinacional. Otra posible solución es hacer que la ruta de la señal de reloj vaya en sentido inverso a la ruta de datos. Esta última técnica (denominada *clock reversing* por algunos fabricantes [38]) solo es aplicable si no existe realimentación en la ruta de datos.

Incluso si se ha previsto y precalculado el *clock skew* en la fase de diseño para intentar garantizar el cumplimiento de las desigualdades 4.3.1 y 4.3.2, debido a variaciones del propio proceso de fabricación, así como de las condiciones de funcionamiento, podría ocurrir que el *clock skew* fuese mayor de lo previsto y que se incumpliera la desigualdad 4.3.2. En este caso el sistema operaría de forma incorrecta para cualquier frecuencia de reloj por baja que fuese.

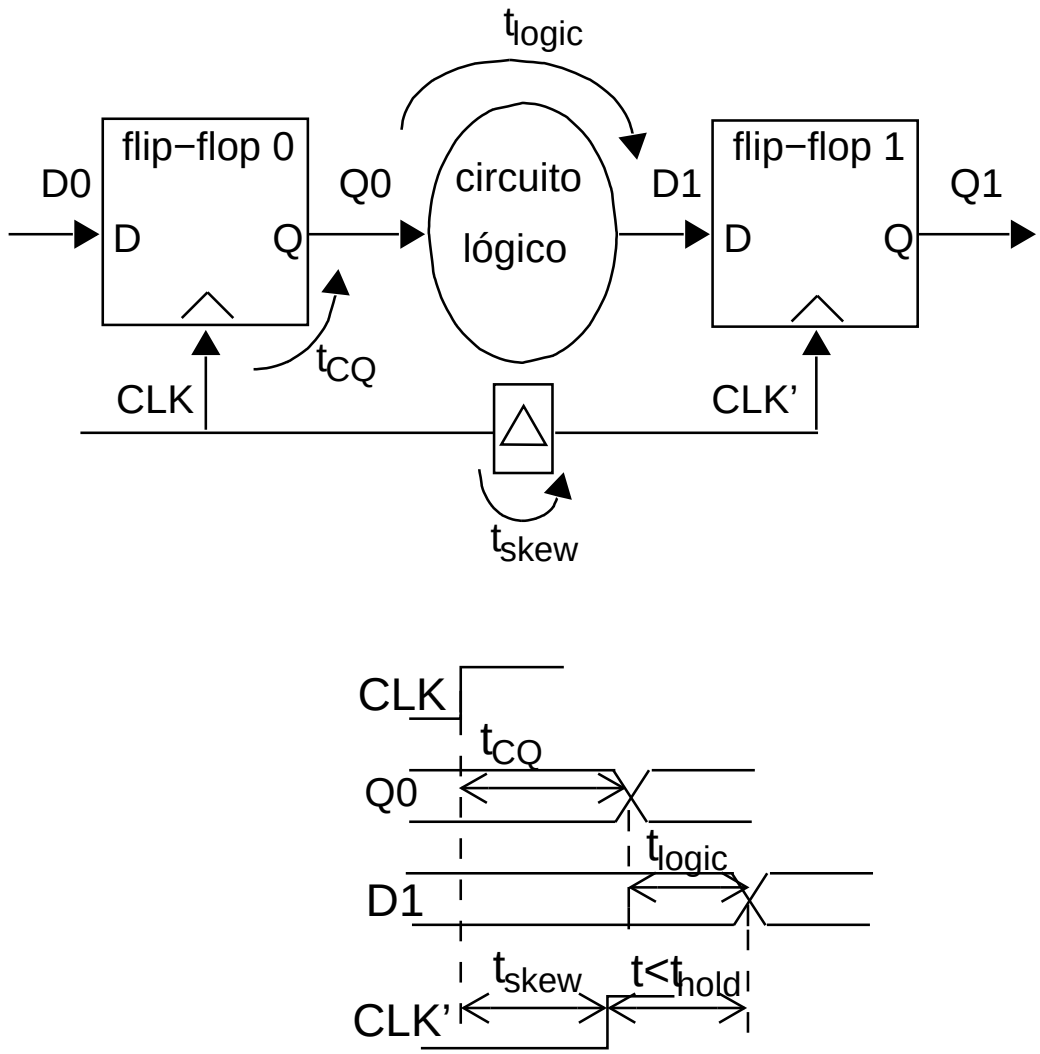



Figura 4.5: Problema de carrera por el camino rápido en un sistema de una sola fase con *flip-flops*.

### 4.3.2. Esquemas de múltiples fases de reloj

Para evitar que el *clock skew* cause mal funcionamiento pueden usarse esquemas de múltiples fases de reloj. Estos esquemas usan múltiples señales de reloj generadas a partir del reloj primario en su última fase de amplificación. Estos sistemas tienen el inconveniente de que necesitan más líneas de reloj, pero permiten evitar los problemas producidos por

el *clock skew*. Son utilizados esquemas de dos fases de reloj no solapadas en *mainframes* de IBM [39]. Un ejemplo de este tipo de esquemas es el esquema de temporización *máster-slave* (figura 4.6.a). En este esquema los datos son primero capturados por los *latches* etiquetados como *máster* y, posteriormente, estos datos se pasan a los *latches* etiquetados como *slave*. El tiempo de no solapamiento entre los niveles sensibles de las señales de reloj evita que ambos *latches* estén simultáneamente en estado transparente. Vamos a realizar un análisis de los requisitos de buen funcionamiento, incluyendo cómo evitar las carreras que puedan ser provocadas por el *clock skew*. Los requisitos temporales, ilustrados en la figura4.6.b, son los siguientes:

- Los pulsos activos de las señales de reloj deben tener un ancho mínimo.
- El dato capturado por los *latches master* debe ser válido durante un tiempo $t_b$ mayor que el tiempo de *setup* ($t_{setup}$) antes de dicha captura.
- El dato capturado por los *latches master* debe mantenerse válido durante un tiempo $t_a$ mayor que el tiempo de *hold* ($t_{hold}$) tras el flanco de captura.
- Como consecuencia de lo anterior, la suma de los retrasos mínimos de los *latches slave*, la lógica combinacional ($t_{logicMin}$) y el tiempo de no solapamiento entre las fases activas de ambos relojes ($t_{noSol}$) debe ser mayor que la suma del tiempo de *hold* y el *skew* ($t_{skew}$).
- El periodo de reloj ($t_{clk}$) debe ser mayor que la suma del tiempo de *set-up,* el tiempo de no solapamiento y los retrasos máximos de los *latches slave* y la lógica combinacional ($t_{slaveMax}$, $t_{logicMax}$).

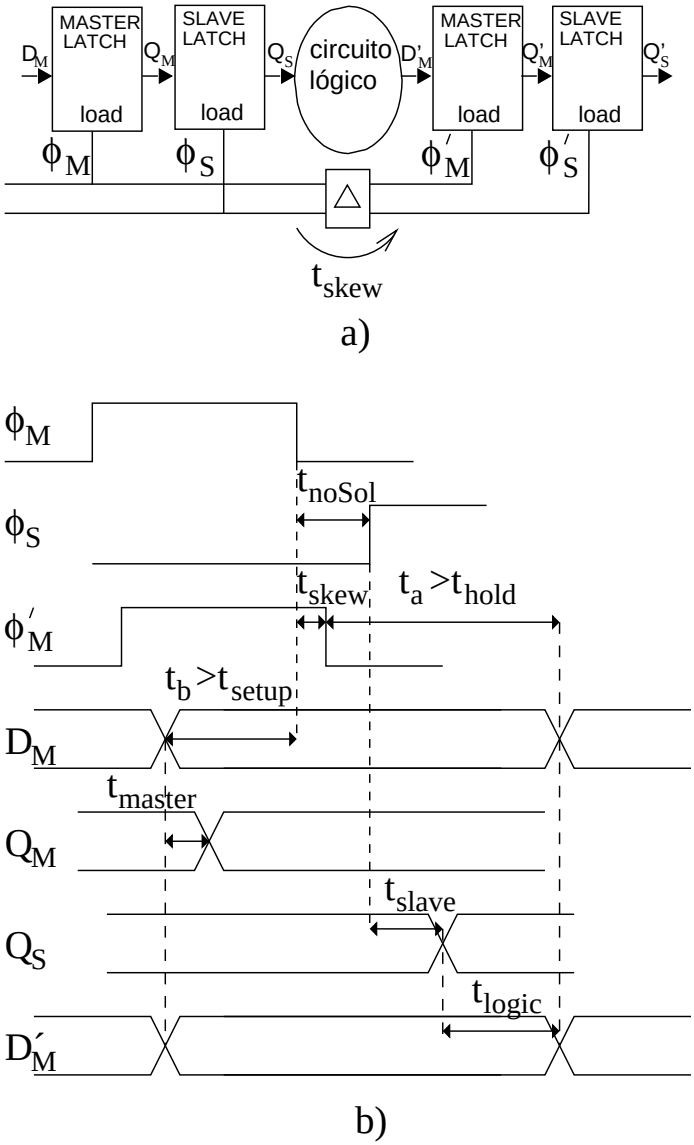



Figura 4.6: Esquema de temporización *máster-slave*.

Las dos últimas condiciones se formulan en las desigualdades 4.3.3 y 4.3.4. Nótese que son similares a las ecuaciones 4.3.1 y 4.3.2 con la diferencia de que aparece el tiempo de no solapamiento entre fases. Este parámetro permite evitar las carreras incluso si, al implementar el sistema, el *skew* es superior al previsto, ya que en ese caso se puede incrementar el tiempo de no solapamiento para que se satisfagan las condiciones de buen funcionamiento.

Los esquemas de dos fases tienen otras ventajas como permitir el testado de circuitos tal y como se detalla en [31].

$$t_{clk} > t_{setup} + t_{noSol} + t_{slaveMax} + t_{logicMax} \quad (4.3.3)$$

$$t_{hold} < t_{noSol} + t_{slaveMin} + t_{logicMin} - t_{skew} \quad (4.3.4)$$

## 4.4. Esquemas de reloj de *latches* alternantes

En este apartado vamos a presentar dos nuevos esquemas de sincronización multifase; uno de dos fases de reloj y otro de cuatro fases de reloj. El objetivo que perseguimos es mejorar las prestaciones que se obtienen con el esquema *máster-slave* manteniendo la principal característica que supone el uso de más de una fase de reloj, es decir, la inmunidad al *clock skew*. Las prestaciones las mediremos en términos de velocidad de operación y consumo de potencia. La comparación entre esquemas de sincronización la realizaremos en el siguiente capítulo.

Centrándonos en los esquemas multifase que vamos a analizar en este apartado, ambos están basados en estructuras ya conocidas para esquemas de una fase de reloj que emplean *flip-flops* de doble flanco activo [40, 41]. En la figura 4.7 se muestra un diseño básico a nivel lógico de un *flip-flop* con estas características. Sobre el esquema de doble flanco activo vamos a proponer y analizar esquemas de varias fases de reloj que proporcionan, como demostraremos, inmunidad al *clock skew* y ventajas en consumo de energía/potencia.

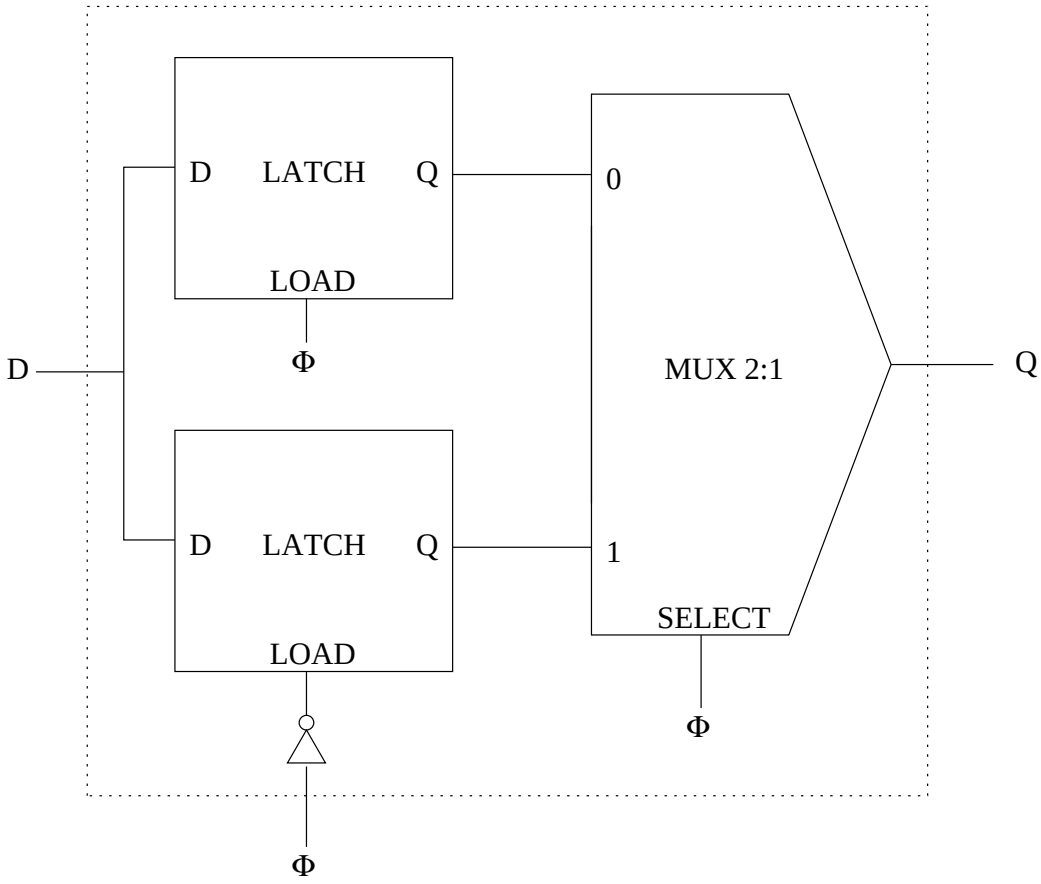



Figura 4.7: *flip-flop* disparado por flanco de subida y de bajada.

### 4.4.1. Esquema de *latches* alternantes de dos fases de reloj

El primero de los esquemas de múltiples fases que vamos a proponer lo denominamos PALACS (*Parallel Alternating Latches Clocking Scheme*, esquema de reloj de latches paralelos alternantes). Su esquema a nivel lógico se muestra en la figura 4.8.a. En la figura 4.8.b se muestran las formas de ondas que describen su comportamiento. Para implementar el bloque de registros, el esquema usa estructuras PRAWN (*Parallel Read And Write Node*, nodo de lectura y escritura paralela). Una estructura PRAWN está formada por dos *latches* conectados en paralelo que comparten la entrada y

una llave de paso a la salida de cada *latch* cuyas salidas están conectadas. La carga de ambos *latches* está controlada por señales de reloj distintas, y cada llave de paso está a su vez controlada por una fase de reloj opuesta a la que controla la carga del *latch* correspondiente.

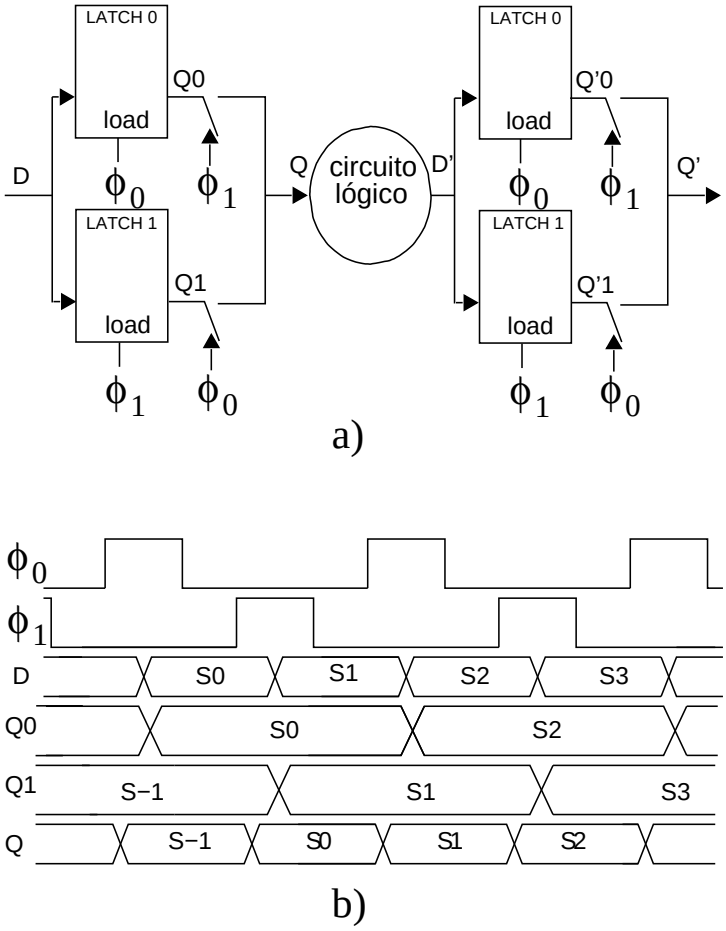



Figura 4.8: Esquema PALACS de dos fases de reloj.

Este esquema, al contrario que el esquema *máster-slave*, permite leer y escribir simultáneamente el bloque de registros durante el nivel activo de cada fase de reloj. En efecto, mientras $\phi_0$ está activa, el *latch* 0 carga la entrada actual mientras el *latch* 1 mantiene la entrada previa. El dato del *latch* 1 se lee durante la fase activa de $\phi_0$, dado que su llave de paso está controlada por $\phi_0$. Cuando $\phi_0$ se desactiva el *latch* 0 deja de ser transparente y ambas fases permanecen inactivas un intervalo de tiempo lo suficientemente largo

como para evitar problemas relacionados con el *clock skew*. Durante este intervalo, ambas llaves están en estado de alta impedancia, pero el dato previo permanece cargado en la salida de las llaves de paso debido a la capacidad parásita asociada al nodo. Cuando $\phi_1$ se activa, el mecanismo de lectura-escritura se pone de nuevo en marcha, pero ambos *latches* alternan sus funciones, es decir, el *latch* 1 carga un nuevo valor mientras se lee el *latch* 0. Podríamos decir que este esquema de temporización es el equivalente de dos fases de reloj del esquema de una fase basado en *flip-flops* disparados por doble flanco de reloj [36].

Una ventaja importante del esquema PALACS respecto al *máster-slave* es que la frecuencia de reloj se reduce en un 50 % para una misma tasa de datos. Esto trae consigo importantes beneficios, principalmente en la reducción de la energía consumida por la red de distribución del reloj. En efecto, en el esquema PALACS, el número de transiciones de reloj es de dos por cada ciclo de cómputo, mientras que en el *máster-slave* el número es cuatro por ciclo de cómputo. Esto significa que la potencia disipada por la red de distribución podría, en principio, reducirse hasta un 50 %. Este ahorro puede llegar a ser muy significativo en el consumo total de potencia del sistema: Según [42] y [43],la potencia disipada por la red de distribución del reloj supone entre un 15 % y un 50 % del total del consumo de un microprocesador. En microprocesadores modernos de altas prestaciones, la red de distribución es responsable de hasta un 70 % del consumo dinámico [44]. Además los generadores de reloj en el esquema PALACS son más fáciles de diseñar y presentan una menor incertidumbre al requerir señales de reloj de la mitad de la frecuencia que en el caso de un único flanco activo para la misma velocidad de operación. Otra ventaja interesante es que el retraso de propagación de la estructura PRAWN es menor que el retraso de propagación de la estructura *máster-slave*, dado que en el esquema *máster-slave* la señal de entrada al bloque de registros debe propagarse a través de dos *latches* antes de llegar al circuito combinacional, mientras que en el esquema PALACS solo tiene que propagarse a través de un *latch* y una llave de paso cuyo retraso es típicamente menor que el retraso de un *latch*. Esto produce una mejora en la velocidad de operación del sistema.

A continuación vamos a determinar las condiciones de buen funcionamiento para este esquema de forma similar a como ya se hizo para los esquemas tradicionales analizados en el capítulo anterior. Los requisitos temporales, ilustrados en la figura 4.9, son los siguientes:

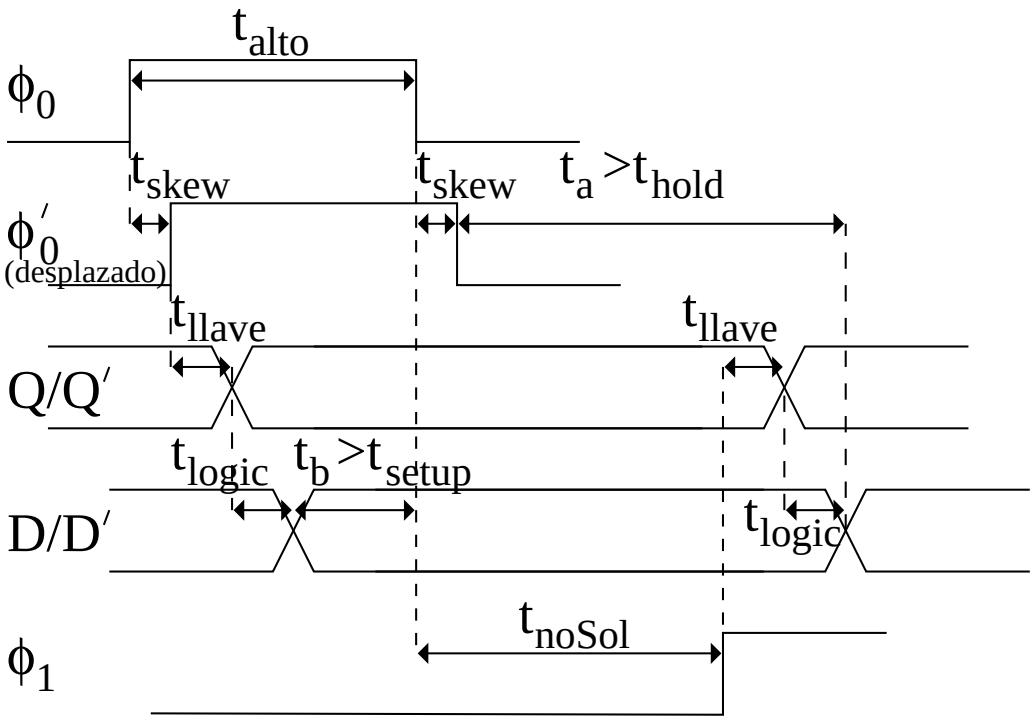



Figura 4.9: Restricciones temporales para el esquema PALACS de dos fases de reloj.

- La duración de los pulsos activos de las señales de reloj ($t_{alto}$) debe tener un valor mínimo.
- El tiempo de no solapamiento entre las fases activas de ambos relojes ($t_{noSol}$) debe ser mayor que cero (las fases activas no pueden solaparse).
- El dato capturado por los *latches* debe ser válido durante un tiempo $t_b$ mayor que el tiempo de *setup* ($t_{setup}$) antes de dicha captura.
- El dato capturado por los *latches* debe mantenerse válido durante un tiempo $t_a$ mayor que el tiempo de *hold* ($t_{hold}$) tras el flanco de captura.

- Como consecuencia de lo anterior, la suma de los retrasos mínimos de las llaves de paso ($t_{llaveMin}$), la lógica combinacional ($t_{logicMin}$) y el tiempo de no solapamiento entre las fases activas de ambos relojes ($t_{noSol}$) debe ser mayor que la suma del tiempo de *hold* y el *skew* ($t_{skew}$).

- La duración de los pulsos activos debe ser mayor que la suma del tiempo de *set-up,* el *skew* y los retrasos máximos de los componentes combinacionales ($t_{llaveMax}$, $t_{logicMax}$).

Las dos últimas condiciones se formulan en las desigualdades 4.4.1 y 4.4.2. Teniendo en cuenta que en este esquema de temporización hay dos intervalos $t_{alto}$ y $t_{noSol}$ en cada ciclo de reloj, podemos emplear ambas desigualdades para obtener la expresión del periodo mínimo de las señales de reloj 4.4.3.

$$t_{alto} > t_{skew} + t_{llaveMax} + t_{logicMax} + t_{setup} \quad (4.4.1)$$

$$t_{noSol} > maximo\{0, t_{hold} - t_{llaveMin} - t_{logicMin} + t_{skew}\} \quad (4.4.2)$$

$$t_{clk} = 2(t_{alto} + t_{noSol}) > 2(t_{skew} + t_{llaveMax} + t_{logicMax} + t_{setpup} + máximo\{0, t_{hold} - t_{llaveMin} - t_{logicMin} + t_{skew}\}) \quad (4.4.3)$$

### 4.4.2. Esquema PALACS de cuatro fases de reloj

Para comprender ciertos inconvenientes del esquema PALACS de dos fases de reloj debemos exponer una técnica de implementación de CSS denominada *segmentación* o *pipelining* [13]. En particular explicaremos como funciona un sistema secuencial síncrono segmentado. A modo de ilustración, supongamos una función lógica $F$ que solo es factible implementar en al menos $n$ niveles de lógica combinacional debido a su complejidad. Un esquema de dicha implementación junto con un cronograma que muestra su funcionamiento se ilustra en la figura 4.10. En el cronograma el circuito está siendo usado para computar $F$ para una secuencia de datos de entrada

$\{ d_1, d_2, d_3, \cdots \}$. En este escenario lo más relevante no es lo que tarda en computarse $F$ para una entrada particular sino lo que tarda en computarse para la secuencia de entrada completa. En este contexto se define el *rendimiento* [13] como el número de veces que el sistema computa la función por unidad de tiempo. Para una secuencia de entrada larga, el rendimiento es la frecuencia con la que el circuito va proporcionando los valores computados $\{ F(d_1), F(d_2), F(d_3), \cdots \}$. Naturalmente, esta coincide con la frecuencia con la que recibe la secuencia de entrada. La inversa de esta frecuencia es el tiempo que transcurre desde que se proporciona un dato hasta que se proporciona el siguiente y lo denominaremos *periodo de cómputo*. Conviene reducir el periodo de cómputo para aumentar el rendimiento, sin embargo este periodo debe tener un valor mínimo para garantizar el buen funcionamiento del sistema. Para encontrar una cota inferior del periodo de cómputo que garantice el buen funcionamiento necesitamos conocer los siguientes parámetros:

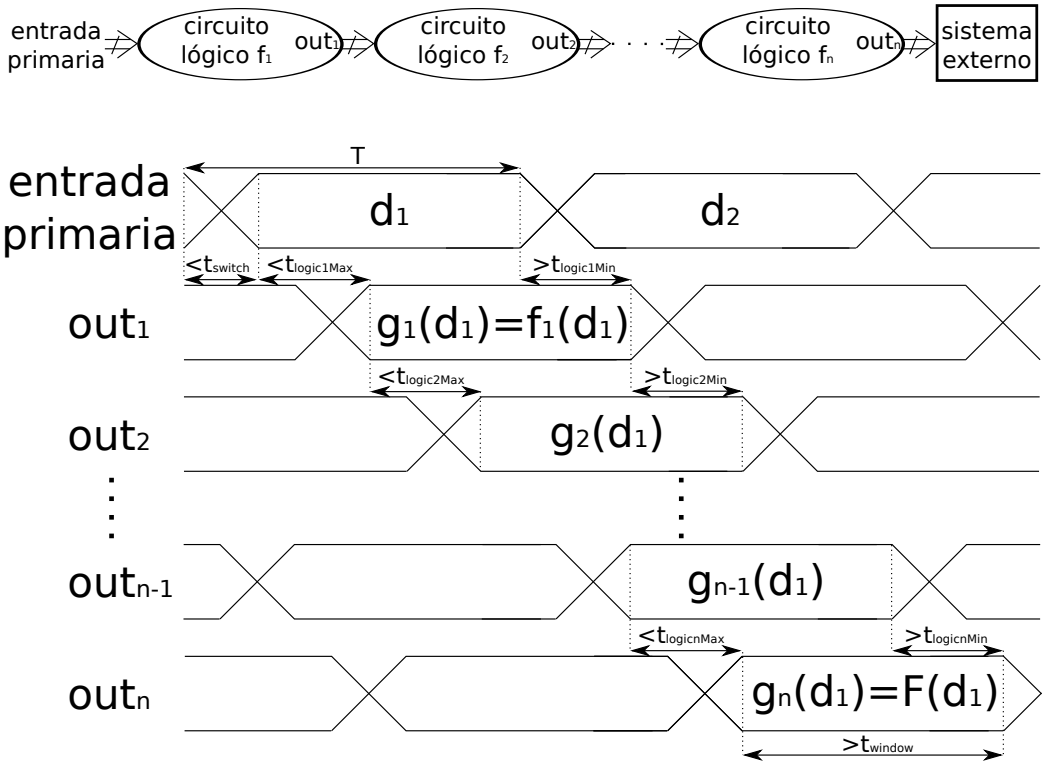



Figura 4.10: Implementación combinacional de la función F

- Una cota superior de la amplitud de la ventana de metaestabilidad del sistema externo que leerá los valores computados. Recordemos que la ventana de metaestabilidad es el periodo durante el cual la salida debe mantener el valor a ser leído para garantizar su correcta captura.

- Una cota superior del tiempo que tarda en conmutar completamente la entrada primaria del circuito. Esto dependerá, entre otras cosas, del sistema que genere dicha entrada.

- Una cota inferior del retraso de contaminación del circuito, es decir, del tiempo que seguirá mostrando la salida correspondiente a un patrón de entrada después de que dicho patrón deje de estar presente en la entrada.

- Una cota superior del retraso del circuito.

En el cronograma de la figura 4.10, $f_i$ denota la función computada por cada nivel de lógica $i$ mientras que $g_i$ denota la función computada por la concatenación de niveles de lógica del 1 al $i$. Así, $g_1 = f_1$, $g_{i+1} = g_i \circ f_{i+1}$ y $g_n = F$. El periodo de cómputo se ha denotado como $T$, la cota de la amplitud de la ventana de metaestabilidad del sistema que captura la salida como $t_{window}$ y la del tiempo de conmutación del sistema que genera la entrada como $t_{switch}$. Además, para cada nivel de lógica $K$ se ha denominado $t_{logicKMax}$ a una cota superior de su retraso y $t_{logicKMin}$ a una cota inferior de su retraso de contaminación. Es evidente que el retraso total del circuito puede acotarse superiormente por $\sum_{K=1}^{n} t_{locigKMax}$ mientras que el retraso de contaminación del circuito puede acotarse inferiormente por $\sum_{K=1}^{n} t_{locigKMin}$, de modo que para garantizar el buen funcionamiento del circuito podría tomarse un $T$ que cumpliese lo siguiente:

$$T > t_{switch} + \sum_{K=1}^{n} t_{locigKMax} + t_{window} - \sum_{K=1}^{n} t_{locigKMin} \tag{4.4.4}$$

La inecuación 4.4.4 puede reescribirse de la siguiente forma:

$$T > t_{externo} + \sum_{K=1}^{n} t_{faseK} \tag{4.4.5}$$

donde $t_{externo} = t_{switch} + t_{window}$ y $t_{faseK} = t_{locigKMax} - t_{locigKMin}$. Dado que $t_{locigKMax} > t_{locigKMin}$ tenemos que $t_{nivelK} > 0$ para cualquier nivel de lógica $K$. Para mostrar con más claridad como depende $T$ del número de niveles de lógica tomemos $t_{faseInf} = \text{mín}_{k=1}^{n}\, t_{faseK}$. Una condición necesaria pero no suficiente para garantizar el buen funcionamiento del sistemas será la siguiente:

$$T > t_{externo} + n * t_{faseInf} \tag{4.4.6}$$

El buen funcionamiento del circuito de la figura 4.10 no puede garantizarse para periodos de cómputo inferiores al mostrado en la ecuación 4.4.6. Esta restricción puede ser muy fuerte si el número de niveles de lógica es grande. Si se requiere un rendimiento más elevado puede aplicarse la técnica de segmentación secuencial síncrona mencionada al principio de esta sección, lo que convertirá al circuito combinacional de la figura 4.10 en un CSS que realizará la misma función. Para hacerlo se introducirá entre cada par de niveles de lógica adyacentes un bloque de registros que almacenarán los valores $g_i$ intermedios. Si, por ejemplo, los registros introducidos son biestables tipo D disparados por el flanco de subida de una señal de reloj común, el circuito quedaría como el ilustrado en la figura 4.11. En dicha figura $t_{CQ}$ denota el retraso de los registros tipo *flip-flop* introducidos.

A primera vista esta modificación puede no parecer ventajosa ya que el tiempo empleado para computar un valor individual aumenta debido al retraso de los registros introducidos. Sin embargo puede mejorarse notablemente el rendimiento, es decir, el tiempo necesario para computar una secuencia larga de valores. Para ver como puede ser esto posible analicemos el mínimo periodo de cómputo que garantice el buen funcionamiento del sistema segmentado haciendo las siguientes simplificaciones:

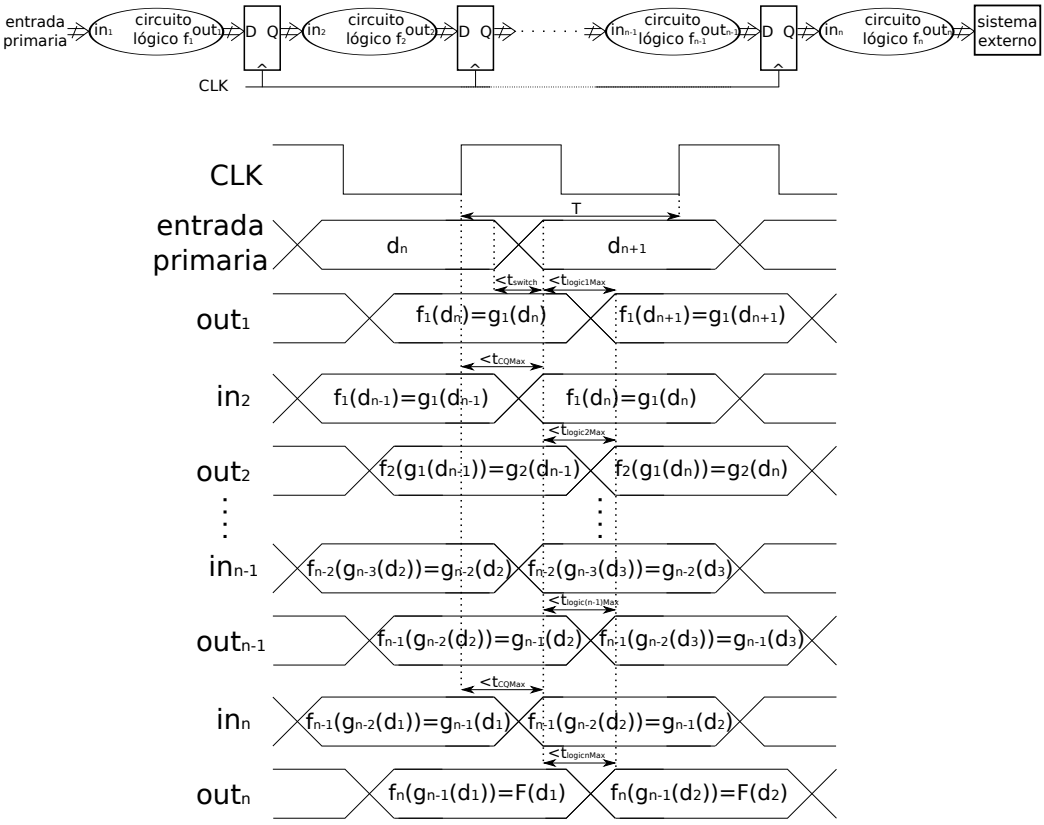



Figura 4.11: Implementación secuencial síncrona segmentada de la función F

- Los sistemas externos conectados a la entrada y la salida del sistema segmentado se encuentran sincronizados con el reloj de este.
- La amplitud de la ventana de metaestabilidad del sistema que captura la salida no es mayor que la de los biestables del sistema segmentado.
- El tiempo que tarda en conmutar la entrada primaria no es mayor que el retraso de los biestables del sistema segmentado.
- Los retrasos de contaminación son lo suficientemente grandes como para garantizar que no se viola el tiempo de *hold* de los *flip-flops*. Recuérdese que de no ser así no podría garantizarse el buen funcionamiento para ningún periodo de reloj tal y como se detalló en la sección 4.3.1.
- No hay desplazamiento de reloj.

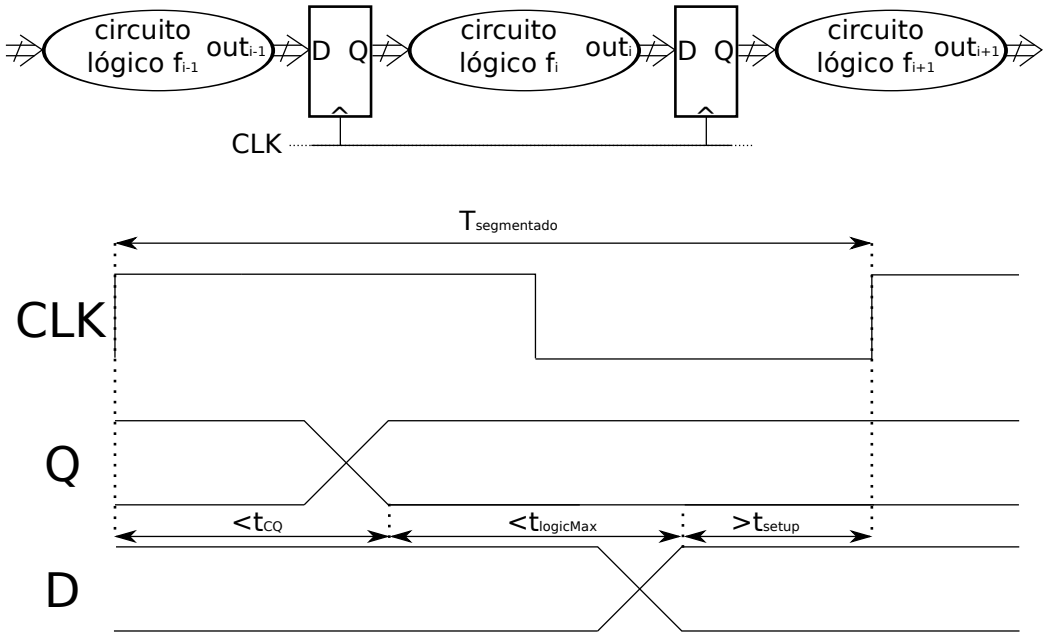



Figura 4.12: Periodo de cómputo del sistema segmentado

Sea $t_{slowest}$ el mayor retraso de todos los niveles combinacionales, es decir$t_{slowest} = \text{máx}_{K=1}^{n} t_{locigKMax}$, para garantizar el buen funcionamiento basta con asegurarse de que se cumple la restricción del tiempo de *setup* como se ilustra en la figura 4.12. De esta forma el periodo de cómputo del sistema segmentado quedaría acotado de la forma siguiente:

$$T_{segmentado} > t_{CQ} + t_{slowest} \tag{4.4.7}$$

Nótese que, al contrario que en la implementación no segmentada, el rendimiento no depende del número de niveles de lógica combinacional sino únicamente del retraso del nivel de lógica más lento. Puede por tanto añadirse niveles al sistema sin perjudicar el rendimiento siempre y cuando ninguno de los niveles introducidos tenga un retraso superior al del nivel más lento. Si el número de niveles es elevado y el retraso de estos es similar, el rendimiento de la implementación segmentada puede ser hasta $n$ veces mayor [13].

Como se ha hecho notar, el tiempo de ciclo del sistema segmentado construido con *flip-flops* está condicionado por el retraso del segmento más lento, por lo que mejoras en la velocidad de los otros segmentos resultan

inútiles. Esto es debido a que, independientemente del momento en que llegue un dato a un registro, dicho dato no podrá seguir propagándose por el circuito hasta que el registro reciba un flanco activo de la señal de reloj. Por eso se dice que en el esquema de temporización basado en *flip-flops* los flancos activos son *flancos duros* [3]. De forma análoga, en el esquema PALACS de dos fases de reloj ilustrado en la figura 4.8 los flancos de subida de las señales que controlan la carga de los *latches* son flancos duros ya que cuando un dato llega a la salida de un *latch* dicho dato no puede seguir propagándose por el circuito hasta que la señal de carga del *latch* homólogo reciba el próximo flanco de subida. En sistemas sin flancos duros es posible usar técnicas llamadas de *substracción de tiempo* (*time borrowing*) [3, 45] para permitir segmentos de retraso mayor que el tiempo de ciclo. En esta técnica el tiempo excedido del ciclo por los segmentos lentos se compensa con el ahorrado en otros segmentos más rápidos. La posibilidad de emplear substracción de tiempo da mayor flexibilidad al diseñador, por lo que sería deseable la eliminación de flancos duros. Con ese propósito se introduce el esquema PALACS de cuatro fases de reloj que se muestra en la figura 4.13.

En este esquema se independizan las señales que controlan la carga de los *latches* de las que controlan las llaves de paso. Este esquema es prácticamente idéntico al de dos fases salvo que, al separar las señales de carga de las que activan las llaves de paso, el dato a la salida de un *latch* puede seguir propagándose por el circuito antes de que el *latch* homólogo comience a cargar. Nótese que, debido a la multitud de señales de control que tiene este esquema, hay que contemplar una enorme casuística como se ilustra en el cronograma de la figura 4.13: la carga de los *latches* puede comenzar antes o después de que su entrada sea válida, el retraso a considerar para los latches puede ser el de la entrada de datos o la de control de carga, etc. Esto no hace viable dar una fórmula cerrada que muestre como está restringido el periodo tal y como se hizo en los esquemas anteriores. Para solucionarlo el periodo y la forma de onda requerida por las señales de reloj puede calcularse de forma automatizada empleando el algoritmo que se presentará en el capítulo 5. De momento podemos adelantar que, dado que el esquema PALACS de dos fases es un caso particular del de esquema PALACS de cuatro fases en el que las

señales de control de carga y las de control de las llaves de paso coinciden, el rendimiento computacional de este último será como mínimo tan bueno como el del primero. Naturalmente, tiene el inconveniente de que requiere el doble de señales de reloj, y cada una de ellas tiene la misma frecuencia que la del PALACS de dos fases para el mismo rendimiento, por lo que la energía consumida por su red de distribución de reloj será mayor.

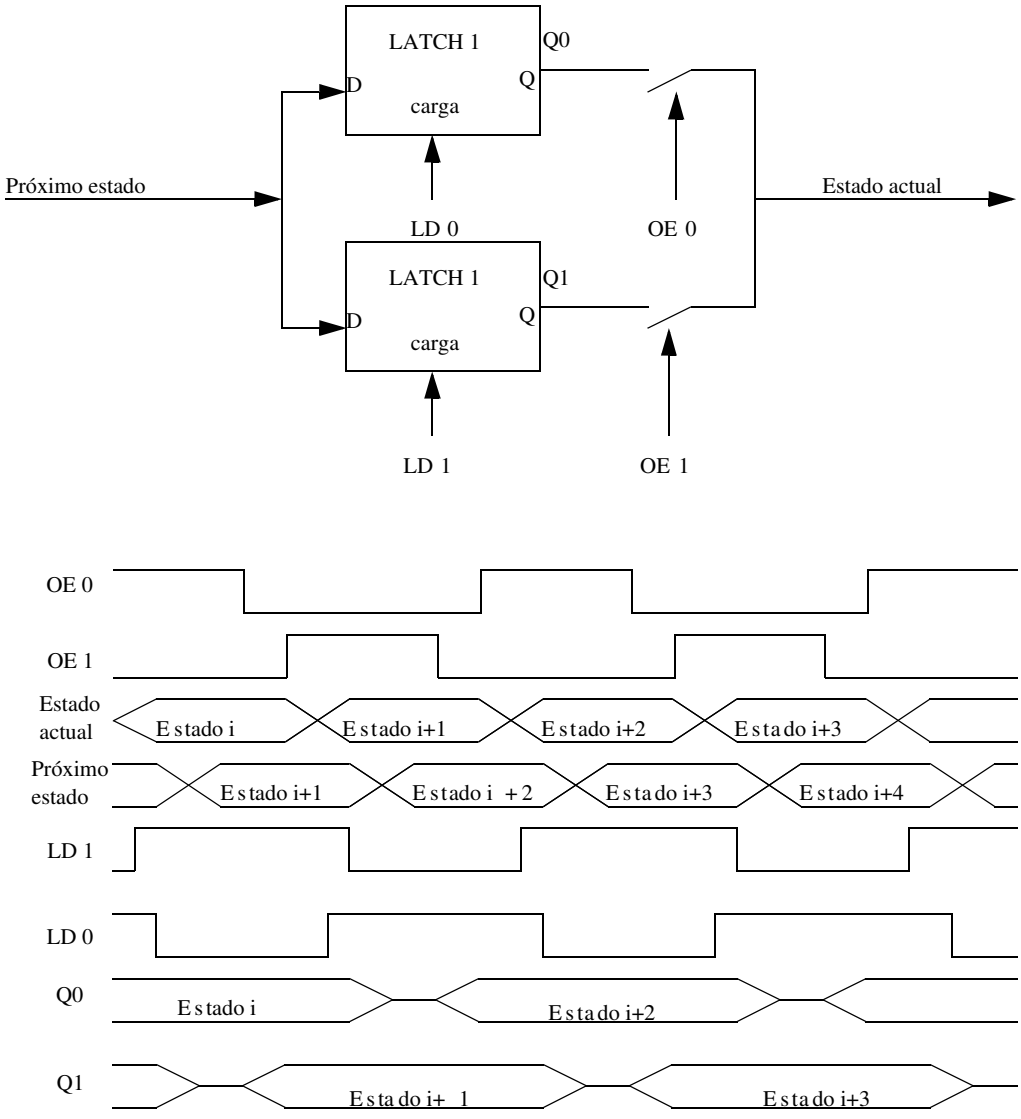



Figura 4.13: Esquema de temporización PALACS de cuatro fases de reloj.

### 4.4.3. Codificación VHDL para esquemas PALACS

En este apartado se describe como utilizar el lenguaje de descripción de hardware VHDL [46] para diseñar sistemas síncronos que usen los esquemas PALACS. Se asume que el lector es capaz de comprender descripciones de sistemas secuenciales escritas en VHDL. No obstante, haremos una breve introducción al lenguaje:

VHDL es un lenguaje de descripción de hardware digital. Fue iniciado por el departamento de defensa de Estados Unidos en los años ochenta para que los componentes que proporcionaban sus distintos suministradores fuesen documentados de manera uniforme. Actualmente, VHDL es un estándar del IEEE muy difundido y utilizado en la industria, existiendo multitud de herramientas que lo usan para modelar, sintetizar y simular sistemas digitales. VHDL permite dos estilos de descripción: estructural y de comportamiento. En la descripción estructural el sistema se describe indicando cuales son sus componentes y como están interconectados. La descripción de comportamiento es de más alto nivel, ya que en ella se indica qué hace el sistema sin tener que especificar detalles de implementación. Esto no significa que una descripción estructural se corresponda de forma exacta con la implementación final: una herramienta de síntesis puede partir de la descripción estructural para generar un circuito optimizado con una funcionalidad equivalente. Para describir esquemas PALACS usaremos ambos estilos. Veremos como describir la estructura del PALACS de cuatro fases, ya que el PALACS de dos fases es un caso particular de este.

Los diseños síncronos basados en lógica estática usan registros controlados por un conjunto de señales de reloj para sincronizar los cambios de estado. En el caso de PALACS, los registros que codifican el estado actual se encuentran estructurados como se vio en las figuras 4.8 y 4.13. El primer paso consiste en describir el *latch* con salida en alta impedancia. Para ello puede usarse el código de la figura 4.14. Se ha incluido una señal de *clear* para la inicialización. Obsérvese que tanto la señal de *clear* como la de habilitación de la salida son activas en nivel bajo. Tras esto el componente declarado se usa para describir la estructura PALACS como se muestra en la figura 4.15.

```
entity latchOEclr is
port (d,noe, ld, nclr : in std_logic;q : out std_logic);
end latchOEclr;
architecture comportamiento of latchOEclr is
signal qi: std_logic;
begin
asignacion: process(ld, d, nclr)
begin
    if ld='1' then
        qi<=d;
    end if;
    if nclr='0' then
        qi<='0';
    end if;
end process;
salida: process(qi,noe)
begin
    q<='Z';
    if noe='0' then
        q<=qi;
    end if;
end process;
end comportamiento;
```

Figura 4.14: Descripción funcional de los componentes que integran la estructura PALACS.

```
use work.latchOEclr;
entity palacs4 is
port (d, noe0, noe1, clk0, clk1, nclr : in std_logic;
            q : out std_logic);
end palacs4;
architecture estructural of palacs4 is
component latchOEclr
port (d, noe, ld, nclr : in std_logic; q : out std_logic);
end component;
begin
    latch0 : latchOEclr port map (d => d,
                                  noe => noe0,
                                  ld => clk0,
                                  nclr => nclr,
                                  q => q);
latch1 : latchOEclr port map (d => d,
                                  noe => noe1,
                                  ld => clk1,
                                  nclr => nclr,
                                  q => q);
end estructural;
```

Figura 4.15: Descripción estructural de la estructura PALACS.

La descripción de la parte combinacional del sistema no requiere ningún tratamiento especial. Igual que en cualquier diseño puede usarse el estilo estructural o bien expresiones que calculen el próximo estado y la salida de la máquina de estado en función del estado actual. Después se describen las conexiones entra la parte combinacional y la secuencial, es decir, las estructuras PALACS, de forma estructural. A modo de ejemplo, la figura 4.16 muestra como describir un contador ascendente de cuatro bits.

```
use work.palacs4;
entity contadorMod16 is
port (noe0, noe1, clk0, clk1, nclr : in std_logic;
            salida : out std_logic_vector (3 downto 0));
end contadorMod4;
architecture estructural of contadorMod4 is
signal estado_cuenta,proximo_estado: unsigned (3 downto 0);
component palacs4
port (d, noe0, noe1, clk0, clk1, nclr : in std_logic; --
            q : out std_logic);
end component;
begin
    logica: process(estado_cuenta)
    begin
        salida<=std_logic_vector(estado_cuenta);
        proximo_estado<=estado_cuenta+1;
    end process;
    genera_registros: for i in 3 downto 0 generate
    digito : palacs4 port map (q => estado_cuenta(i),
                                    noe0 => noe0,
                                    noe1 => noe1,
                                    clk0 => clk0,
                                    clk1 => clk1,
                                    nclr => nclr,
                                    d => proximo_estado(i));
    end generate genera_registros;
end estructural;
```

Figura 4.16: Descripción VHDL de un contador de cuatro bits usando PALACS.

# Capítulo 5

# Prestaciones de los esquemas de reloj

El objetivo específico de este capítulo es realizar un análisis comparativo de las prestaciones de los esquemas de reloj. Nos vamos a centrar en los esquemas insensibles al *clock skew*, esto es, los esquemas multifase. En concreto, el esquema de dos fases de reloj *máster-slave* (M-S), el esquema de dos fases de *latches* alternantes (PALACS) y el esquema de cuatro fases de *latches* alternantes (PALACS-4) presentado en el capítulo anterior.

El análisis va a estar centrado en dos aspectos básicos: la velocidad de operación y el consumo de potencia. Así, en este capítulo, para poder llevar a cabo el análisis sobre ejemplos concretos, vamos a emplear los componentes propios de una librería de celdas estándar donde existen elementos de memoria adecuados para cada uno de los esquemas de reloj que vamos a analizar. Aunque los resultados obtenidos dependen de la tecnología empleada, estos resultados son extrapolables a otras tecnologías y la metodología de análisis tiene un carácter general.

En principio podría pensarse que las ecuaciones 4.3.3 y 4.4.3 presentadas en el capítulo 4 podrían emplearse para hacer una comparación. Sin embargo, no es fácil hacer comparaciones de periodo mínimo (frecuencia máxima) teniendo que determinar el tiempo de no solapamiento adecuado. Por ello, para llevar a cabo un comparación rigurosa en cuanto a la frecuencia de

operación máxima se han desarrollado algoritmos específicos para cada esquema de reloj que nos permiten obtener las formas de onda de las señales de reloj que garantizan el correcto funcionamiento del esquema en función del *clock skew* y de los parámetros temporales de los componentes del circuito.

Este capítulo se organiza del siguiente modo: El siguiente apartado está dedicado a presentar los algoritmos que se van a emplear para obtener la forma de onda de las señales de reloj para cada uno de los esquemas. Posteriormente, y sobre dos circuitos relativamente simples, vamos a realizar el estudio comparativo aplicando a cada uno de ellos los diferentes esquemas de reloj.

## 5.1. Algoritmos para obtener la forma de onda de las señales de reloj

Con objeto de comparar las prestaciones de una implementación concreta de estos esquemas de temporización surgió la necesidad de automatizar el proceso de obtención de la forma de onda requerida para las señales de reloj en función de los parámetros temporales del circuito y del *clock skew*. Para ello se han desarrollado una serie de algoritmos que han sido posteriormente implementados en lenguaje C. Estos algoritmos realizan su cometido generando un cronograma en el que se asegura que se cumplan las restricciones temporales que garantizan que funciona correctamente un circuito secuencial síncrono genérico como el mostrado en la figura 1.1. El algoritmo comienza estableciendo una condición de inicio de estabilidad a partir de la cual comienzan a cambiar las señales con los tiempos de retraso que corresponden a cada componente. El algoritmo va estableciendo los instantes de tiempo de cambio de las señales con el objetivo de minimizar el periodo de reloj, pero asegurando asimismo un correcto funcionamiento. Este proceso se realiza de forma reiterada hasta que se obtiene un cronograma periódico. A partir de este cronograma se pueden determinar parámetros como el tiempo de no solapamiento y la frecuencia de reloj. De esta forma, es posible realizar un análisis de la frecuencia de operación en función del *clock*

*skew*.

En este punto es necesario dar una definición formal del valor del *skew* que será usada en los algoritmos. Sea $tr$ una transición producida en una señal de reloj conectada a un conjunto de entradas de control $C$:

- Definimos el instante nominal de llegada de la transición $tr$ como el primero de aquellos en el que dicha transición alcanza una entrada de control, es decir, $min\{llegada(tr, i) \ : \ i \in C\}$, donde $llegada(tr, i)$ denota el instante de llegada de la transición $tr$ a la entrada de control $i$.

- Definimos el *skew* de la transición $tr$ como la diferencia máxima entre el instante de llegada de dicha transición a una entrada de control y su instante nominal de llegada, es decir, $max\{llegada(tr, i) \ : \ i \in C\} - min\{llegada(tr, i) \ : \ i \in C\}$.

Nótese que, según esta definición, el *skew* solo puede tomar valores positivos.

### 5.1.1. Algoritmo para encontrar la forma de onda de las señales de reloj en el esquema de temporización *máster-slave*

El algoritmo que genera un cronograma del funcionamiento de un circuito *máster-slave* se muestra en la figura 5.1. El significado de sus variables y parámetros es el siguiente:

- $t_{skewSu}$: Máximo desplazamiento que pueden sufrir las transiciones de subida de la señal de reloj de los *latches slave*

- $t_{skewSd}$: Máximo desplazamiento que pueden sufrir las transiciones de bajada de la señal de reloj de los *latches slave*

- $t_{skewMu}$: Máximo desplazamiento que pueden sufrir las transiciones de subida de la señal de reloj de los *latches máster*

- $t_{skewMd}$: Máximo desplazamiento que pueden sufrir las transiciones de bajada de la señal de reloj de los *latches máster*

$CLK_{Mu}[0] \leftarrow 0$
$CLK_{Md}[0] \leftarrow pw_{minmaster} + t_{skewMu}$
$QM[0] \leftarrow Lmaster_{CQmax} + t_{skewMu}$
$CLK_{Su}[0] \leftarrow CLK_{Mu}[0] + t_{skewMu} + t_{holdmaster} - L_{slaveCQmin} - LC_{min}$
$S[0] \leftarrow max\{CLK_{Su}[0] + t_{skewSu} + L_{slaveCQmax}, QM[0] + L_{DQmax}\}$
$CLK_{Sd}[0] \leftarrow max\{QM[0] + t_{setupslave}, CLK_{Su}[0] + t_{skewSu} + pw_{minslave}\}$
$NS[0] \leftarrow S[0] + LC_{max}$
$i \leftarrow 0$

$HACER$

$i \leftarrow i + 1$
$CLK_{Mu}[i] \leftarrow CLK_{Sd}[i-1] + t_{skewSd} + t_{holdslave} - L_{masterCQmin}$
$QM[i] \leftarrow max\{CLK_{Mu}[i] + t_{skewMu} + L_{masterCQmax}, NS[i-1] + L_{masterDQmax}\}$
$CLK_{Md}[i] \leftarrow max\{NS[i-1] + t_{setupmaster}, CLK_{Mu}[i] + t_{skewMu} + pw_{minmaster}\}$
$CLK_{Su}[i] \leftarrow CLK_{Md}[i] + t_{skewMd} + t_{holdmaster} - L_{slaveCQmin} - LC_{min}$
$CLK_{Sd}[i] \leftarrow max\{QM[i] + t_{setupslave}, CLK_{Su}[i] + t_{skewSu} + pw_{minslave}\}$
$S[i] \leftarrow max\{CLK_{Su}[i] + t_{skewSu} + L_{slaveCQmax}, QM[i] + L_{slaveDQmax}\}$
$NS[i] \leftarrow S[i] + LC_{max}$

$HASTA\,QUE\,CLK_{Mu}[i] - CLK_{Mu}[i-1] = CLK_{Md}[i] - CLK_{Md}[i-1] = QM[i] - QM[i-1] = CLK_{Su}[i] - CLK_{Su}[i-1] = CLK_{Sd}[i] - CLK_{Sd}[i-1] = S[i] - S[i-1] = NS[i] - NS[i-1]$

$W_M \leftarrow CLK_{Md}[i] - CLK_{Mu}[i]$
$W_S \leftarrow CLK_{Sd}[i] - CLK_{Su}[i]$
$desfase \leftarrow CLK_{Mu}[i] - CLK_{Su}[i]$
$T \leftarrow CLK_{Mu}[i] - CLK_{Mu}[i-1]$

Figura 5.1: Algoritmo para el cálculo de la forma de onda de las señales de reloj en el esquema de temporización *máster-slave*.

- $LC_{max}$: Máximo retraso del circuito lógico
- $LC_{min}$: Retraso de contaminación del circuito lógico, es decir, amplitud del intervalo de tiempo mínimo en el que su salida permanecerá estable después de que haya cambiado su entrada
- $Lmaster_{DQmax}$: Máximo retraso de un *latch máster* cuando cambia su entrada encontrándose su carga activada
- $Lmaster_{CQmax}$: Máximo retraso de un *latch máster* cuando se activa su carga teniendo su entrada un valor válido
- $Lmaster_{CQmin}$: Retraso de contaminación de un *latch máster* cuando se activa su carga, es decir, amplitud mínima del intervalo de tiempo en el que su salida permanecerá estable después de que se haya activado su carga
- $Lslave_{DQmax}$: Máximo retraso de un *latch slave* cuando cambia su entrada encontrándose su carga activada
- $Lslave_{CQmax}$: Máximo retraso de un *latch slave* cuando se activa su carga teniendo su entrada un valor válido
- $Lslave_{CQmin}$: Retraso de contaminación de un *latch slave* cuando se activa su carga, es decir, amplitud mínima del intervalo de tiempo en el que su salida permanecerá estable después de que se haya activado su carga
- $t_{setupmaster}$: Tiempo de *setup* de un *latch máster*
- $t_{setupslave}$: Tiempo de *setup* de un *latch slave*
- $t_{holdmaster}$: Tiempo de *hold* de un *latch máster*
- $t_{holdslave}$: Tiempo de *hold* de un *latch slave*
- $pw_{minmaster}$: Mínima anchura de un pulso alto en la señal de carga de un *latch máster* que garantiza la correcta carga de un dato

- $pw_{minslave}$: Mínima anchura de un pulso alto en la señal de carga de un *latch slave* que garantiza la correcta carga de un dato

- $S[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que las señales de estado (s) han adquirido su nuevo valor

- $NS[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que las señales de próximo estado (NS, *next state*) han adquirido su nuevo valor

- $QM[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que la salida de los *latches* etiquetados como *máster* ha cambiado a su nuevo valor

- $CLK_{Su}[i]$: Instante del ciclo de computación i-ésimo en el que sube la señal de reloj de los *latches slave*

- $CLK_{Sd}[i]$: Instante del ciclo de computación i-ésimo en el que baja la señal de reloj de los *latches slave*

- $CLK_{Mu}[i]$: Instante del ciclo de computación i-ésimo en el que sube la señal de reloj de los *latches máster*

- $CLK_{Md}[i]$: Instante del ciclo de computación i-ésimo en el que baja la señal de reloj de los *latches máster*

- $W_S$: Anchura que debe tener el pulso alto de la señal de reloj de los *latches slave*

- $W_M$: Anchura que debe tener el pulso alto de la señal de reloj de los *latches máster*

- $desfase$: Retraso que debe tener la señal de reloj *slave* respecto a la señal de reloj *máster*

- $T$: Periodo que deben tener las señales de reloj

Como se indicó, estos algoritmos parten de una condición inicial estable. En este caso se supondrá que inicialmente los *latches* de la figura 4.6 etiquetados como *slave* han tenido almacenado el estado inicial durante un tiempo suficiente como para que aparezca el siguiente estado a la entrada de los *latches* etiquetados como *máster* y que el primer pulso positivo corresponde al reloj de los *latches máster*. A partir de esa condición inicial el algoritmo va estableciendo los instantes de cambio de las señales de reloj tan pronto como sea posible sin violar las condiciones que garanticen el buen funcionamiento, en este caso las condiciones de *setup*, *hold* y anchura mínima de pulso de las señales de carga de los *latches máster* y *slave*. La figura 5.2 muestra el cronograma generado por el algoritmo con dichas restricciones resaltadas. La condición de salida del bucle es necesaria para que el algoritmo termine cuando el cronograma se vuelva periódico.

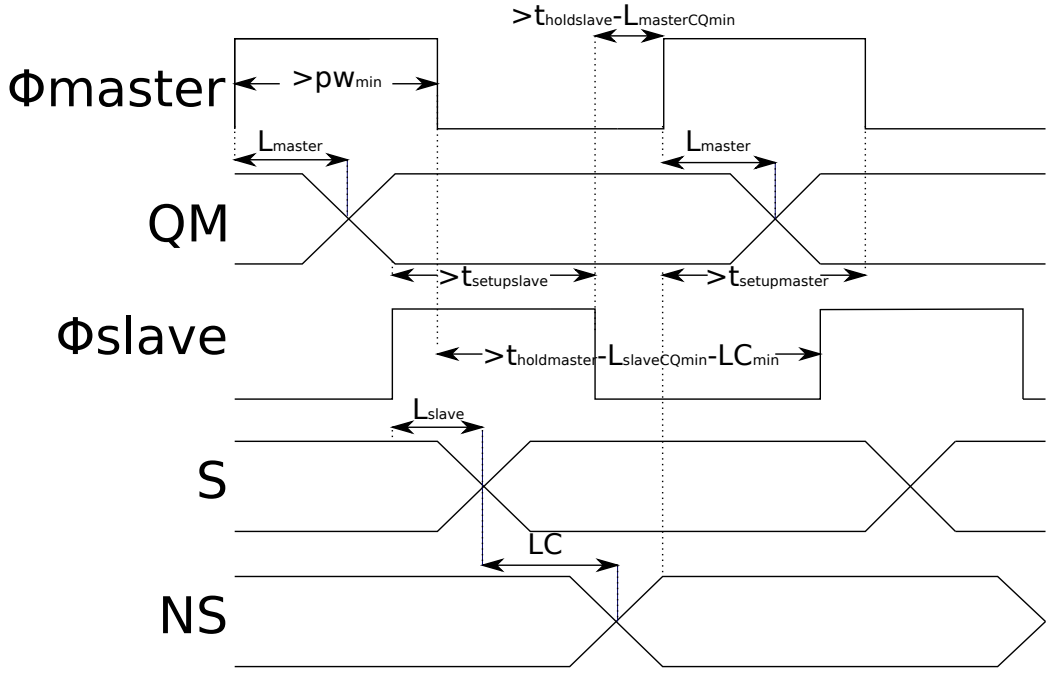



Figura 5.2: Cronograma generado por el algoritmo para el esquema *máster-slave.*

### 5.1.2. Algoritmo para encontrar las formas de onda de las señales de reloj en el esquema de temporización PALACS de dos fases

El algoritmo que genera el cronograma de un circuito PALACS de dos fases se muestra en la figura 5.3. El significado de sus variables y parámetros es el siguiente:

$CLK_u[0] \leftarrow 0$
$S[0] \leftarrow t_{skewu} + K_{cmax}$
$NS[0] \leftarrow S[0] + LC_{max}$
$CLK_d[0] \leftarrow max\{NS[0] + t_{setup}, pw_{min} + t_{skewu}\}$
$Q[0] \leftarrow max\{CLK_u[0] + t_{skewu} + L_{CQmax}, NS[0] + L_{DQmax}\}$
$i \leftarrow 0$

$HACER$

$i \leftarrow i + 1$
$CLK_u[i] \leftarrow CLK_d[i-1] + t_{skewd} + max\{0, t_{hold} - K_{cmin} - LC_{min}\}$
$S[i] \leftarrow max\{CLK_u[i] + t_{skewu} + K_{cmax}, Q[i-1] + K_{imax}\}$
$NS[i] \leftarrow S[i] + LC_{max}$
$CLK_d[i] \leftarrow max\{NS[i] + t_{setup}, CLK_a[i] + t_{skewu} + pw_{min}\}$
$Q[i] \leftarrow max\{CLK_u[i] + t_{skewu} + L_{CQmax}, NS[i] + L_{DQmax}\}$

$HASTA\,QUE\,CLK_u[i] - CLK_u[i-1] = S[i] - S[i-1] = NS[i] - NS[i-1] = CLK_d[i] - CLK_d[i-1] = Q[i] - Q[i-1]$

$W \leftarrow CLK_d[i] - CLK_u[i]$
$T \leftarrow 2 * (CLK_u[i] - CLK_u[i-1])$

Figura 5.3: Algoritmo para el cálculo de la forma de onda de las señales de reloj en el esquema de temporización PALACS de dos fases.

- $t_{skewu}$: Máximo desplazamiento que pueden sufrir las transiciones de subida de la señal de reloj
- $t_{skewd}$: Máximo desplazamiento que pueden sufrir las transiciones de bajada de la señal de reloj

- $LC_{max}$: Máximo retraso del circuito lógico
- $LC_{min}$: Retraso de contaminación del circuito lógico, es decir, amplitud del intervalo de tiempo mínimo en el que su salida permanecerá estable después de que haya cambiado su entrada
- $K_{cmax}$: Máximo retraso de una llave de paso cuando se activa siendo su entrada válida
- $K_{cmin}$: Retraso de contaminación de una llave de paso cuando se activa, es decir, amplitud mínima del intervalo en el que su salida permanecerá estable después de ser activada
- $K_{imax}$: Máximo retraso de una llave de paso cuando cambia su entrada encontrándose activada
- $L_{DQmax}$: Máximo retraso de un *latch* cuando cambia su entrada encontrándose su carga activada
- $L_{CQmax}$: Máximo retraso de un *latch* cuando se activa su carga teniendo su entrada un valor válido
- $t_{setup}$: Tiempo de *setup* de un *latch*
- $t_{hold}$: Tiempo de *hold* de un *latch*
- $pw_{min}$: Mínima anchura de un pulso alto en la señal de carga de un *latch* que garantiza la correcta carga de un dato
- $S[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que las señales de estado (s) han adquirido su nuevo valor
- $NS[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que las señales de próximo estado (ns) han adquirido su nuevo valor
- $Q[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que la salida de los *latches* etiquetados como (i mod 2) ha cambiado a su nuevo valor

- $CLK_u[i]$: Instante del ciclo de computación i-ésimo en el que sube la señal de reloj (i mod 2)
- $CLK_d[i]$: Instante del ciclo de computación i-ésimo en el que baja la señal de reloj (i mod 2)
- $W$: Anchura que debe tener el pulso alto de las señales de reloj
- $T$: Periodo que deben tener las señales de reloj

De nuevo, el algoritmo debe partir de una condición inicial estable. En este caso se supondrá que inicialmente los *latches* de la figura 4.8 etiquetados como 1 han tenido almacenados un estado inicial válido durante un tiempo lo suficientemente grande como para que aparezca a su salida y que el primer pulso positivo corresponde al reloj 0. Igual que antes, el algoritmo cambia cuanto antes las señales de reloj pero garantizando el buen funcionamiento, es decir, velando para que se cumplan las condiciones de *setup*, *hold* y anchura mínima de pulso de las señales de carga de los *latches*. Para ello se postula que si la entrada de un *latch* cambia en un instante $t_i$ mientras que se activa su carga en un instante $t_c$, entonces el nuevo valor de la entrada aparece en su salida en un instante no posterior a $max\{t_i + K_{imax}, t_c + K_{cmax}\}$ tal y como se describe en [47]. La figura 5.4 muestra el cronograma resultante con dichas condiciones resaltadas. Como en el caso *máster-slave*, este algoritmo comprueba en cada iteración si el cronograma se ha vuelto periódico, en cuyo caso termina.

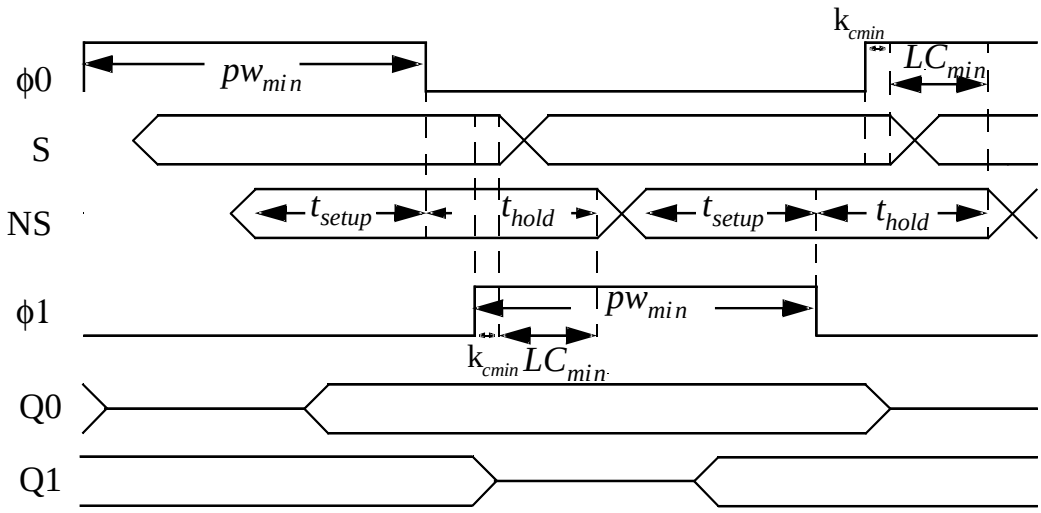



Figura 5.4: Cronograma generado por el algoritmo para el esquema PALACS de dos fases.

### 5.1.3. Algoritmo para encontrar las formas de onda de las señales de reloj en el esquema de temporización PALACS de cuatro fases

El algoritmo para obtener el cronograma del esquema PALACS de cuatro fases se muestra en la figura 5.5. El significado de sus variables y parámetros es el siguiente:

- $t_{skewuCLK}$: Máximo desplazamiento que pueden sufrir las transiciones de subida de las señales de reloj de carga
- $t_{skewdCLK}$: Máximo desplazamiento que pueden sufrir las transiciones de bajada de la señales de reloj de carga
- $t_{skewuOE}$: Máximo desplazamiento que pueden sufrir las transiciones de subida de las señales de reloj de habilitación de salida
- $t_{skewdOE}$: Máximo desplazamiento que pueden sufrir las transiciones de bajada de la señales de reloj de habilitación de salida

$CLK_{uOE}[0] \leftarrow 0$
$CLK_{uLD}[0] \leftarrow 0$
$S[0] \leftarrow t_{skewuOE} + K_{cmax}$
$NS[0] \leftarrow S[0] + LC_{max}$
$CLK_{dLD}[0] \leftarrow max\{NS[0] + t_{setup}, pw_{min} + t_{skewuLD}\}$
$Q[0] \leftarrow max\{CLK_{uLD}[0] + t_{skewuLD} + L_{CQmax}, NS[0] + L_{DQmax}\}$
$i \leftarrow 0$

$HACER$

$i \leftarrow i + 1$
$CLK_{uOE}[i] \leftarrow CLK_{dLD}[i-1] + t_{skewdLD} + t_{hold} - K_{cmin} - LC_{min}$
$CLK_{dOE}[i-1] \leftarrow CLK_{uOE}[i] - t_{skewdOE}$
$CLK_{uLD}[i] \leftarrow CLK_{dLD}[i-1] + t_{skewdLD} + t_{hold} - L_{CQmin} - K_{imin} - LC_{min}$
$S[i] \leftarrow max\{CLK_{uOE}[i] + t_{skewuOE} + K_{cmax}, Q[i-1] + K_{imax}\}$
$NS[i] \leftarrow S[i] + LC_{max}$
$CLK_{dLD}[i] \leftarrow max\{NS[i] + t_{setup}, CLK_{uLD}[i] + t_{skewuLD} + pw_{min}\}$
$Q[i] \leftarrow max\{CLK_{uLD}[i] + t_{skewuLD} + L_{CQmax}, NS[i] + L_{DQmax}\}$

$HASTA\,QUE\,CLK_{uOE}[i] - CLK_{uOE}[i-1] = CLK_{uLD}[i] - CLK_{uLD}[i-1] = S[i] - S[i-1] = NS[i] - NS[i-1] = Q[i] - Q[i-1]$

$T \leftarrow 2 * (CLK_{uLD}[i] - CLK_{uLD}[i-1])$
$CLK_{dOE}[i] \leftarrow CLK_{dOE}[i-1] + T/2$
$W_{LD} \leftarrow CLK_{dLD}[i] - CLK_{uLD}[i]$
$W_{OE} \leftarrow CLK_{dOE}[i] - CLK_{uOE}[i]$
$desfase \leftarrow CLK_{uLD}[i] - CLK_{uOE}[i]$

Figura 5.5: Algoritmo para el cálculo de la forma de onda de las señales de reloj en el esquema de temporización PALACS de cuatro fases.

- $LC_{max}$: Máximo retraso del circuito lógico
- $LC_{min}$: Retraso de contaminación del circuito lógico, es decir, amplitud del intervalo de tiempo mínimo en el que su salida permanecerá estable después de que haya cambiado su entrada
- $K_{cmax}$: Máximo retraso de una llave de paso cuando se activa siendo su entrada válida
- $K_{cmin}$: Retraso de contaminación de una llave de paso cuando se activa, es decir, amplitud mínima del intervalo en el que su salida permanecerá estable después de ser activada
- $K_{imax}$: Máximo retraso de una llave de paso cuando cambia su entrada encontrándose activada
- $K_{imin}$: Retraso de contaminación de una llave de paso cuando cambia su entrada encontrándose activada, es decir, amplitud mínima del intervalo en el que su salida permanecerá estable después de que cambie su entrada
- $L_{DQmax}$: Máximo retraso de un *latch* cuando cambia su entrada encontrándose su carga activada
- $L_{CQmax}$: Máximo retraso de un *latch* cuando se activa su carga teniendo su entrada un valor válido
- $L_{CQmin}$: Retraso de contaminación de un *latch* cuando se activa su carga teniendo su entrada un valor válido, es decir, amplitud mínima del intervalo en el que su salida permanecerá estable después de que se active su carga
- $t_{setup}$: Tiempo de *setup* de un *latch*
- $t_{hold}$: Tiempo de *hold* de un *latch*
- $pw_{min}$: Mínima anchura de un pulso alto en la señal de carga de un *latch* que garantiza la correcta carga de un dato

- $S[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que las señales de estado (s) han adquirido su nuevo valor
- $NS[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que las señales de próximo estado (ns) han adquirido su nuevo valor
- $Q[i]$: Instante del ciclo de computación i-ésimo en el que se sabe que la salida de los *latches* etiquetados como (i+1 mod 2) ha cambiado a su nuevo valor
- $CLK_{uLD}[i]$: Instante del ciclo de computación i-ésimo en el que sube la señal de reloj de carga (i+1 mod 2)
- $CLK_{dLD}[i]$: Instante del ciclo de computación i-ésimo en el que baja la señal de reloj de carga (i+1 mod 2)
- $CLK_{uOE}[i]$: Instante del ciclo de computación i-ésimo en el que sube la señal de reloj de habilitación de salida (i mod 2)
- $CLK_{dOE}[i]$: Instante del ciclo de computación i-ésimo en el que baja la señal de reloj de habilitación de salida (i mod 2)
- $W_{LD}$: Anchura que debe tener el pulso alto de las señales de reloj de carga
- $W_{OE}$: Anchura que debe tener el pulso alto de las señales de reloj de habilitación de salida
- $T$: Periodo que deben tener las señales de reloj
- $desfase$: Tiempo transcurrido desde que se activa la señal de habilitación de salida de un *latch* hasta que se activa la señal de carga del *latch* homólogo.

Como en los algoritmos anteriores, se parte de una condición inicial estable. En este caso se supondrá que inicialmente los *latches* de la figura 4.13 etiquetados con 0 han tenido almacenado un estado inicial válido durante un tiempo suficiente como para que dicho estado aparezca en su salida y

que el primer pulso corresponderá a la señal de reloj $OE_0$. De nuevo, el algoritmo va estableciendo los instantes de cambio de las señales de reloj procurando que sean tan pronto como sea posible siempre que se garantice el buen funcionamiento, es decir, que se cumplen las condiciones de *setup*, *hold* y anchura mínima de pulso de las señales de carga de los *latches*. Para ello, y como en el algoritmo para el PALACS de dos fases, se postula que si la entrada de un *latch* cambia en un instante $t_i$ mientras que se activa su carga en un instante $t_c$, entonces el nuevo valor de la entrada aparece en su salida en un instante no posterior a $max\{t_i + K_{imax}, t_c + K_{cmax}\}$ [47]. La figura 5.6 muestra el cronograma resultante con dichas condiciones resaltadas. Como en los algoritmos anteriores, en cada iteración se comprueba si el cronograma se ha vuelto periódico, en cuyo caso termina el proceso.

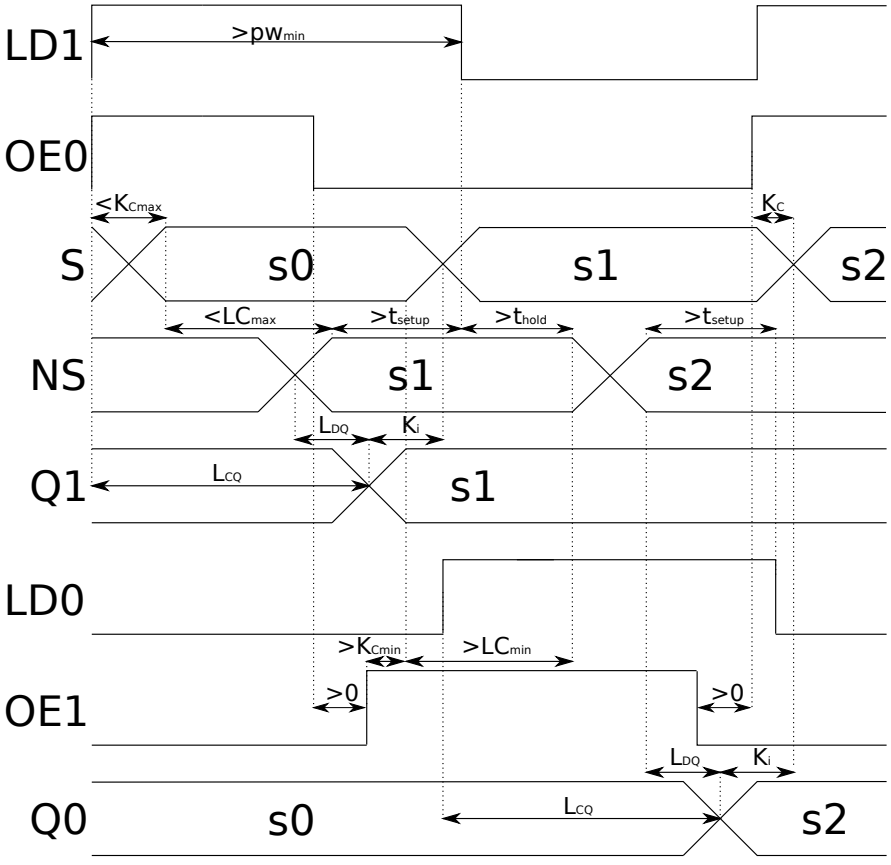



Figura 5.6: Cronograma generado por el algoritmo para el esquema PALACS de cuatro fases.

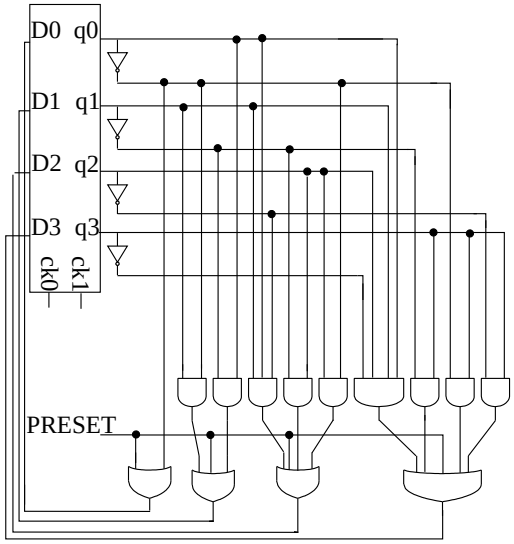
## 5.2. Análisis de las prestaciones de los esquemas de reloj

En este apartado pretendemos analizar las prestaciones en cuanto a velocidad de operación y consumo de energía de los diferentes esquemas de reloj multifase. Para llevar a cabo este análisis vamos a estudiar el comportamiento de dos circuitos simples pero que consideramos adecuados para nuestros propósitos. Los circuitos se muestran en la figura 5.7. En concreto, en la figura 5.7.a se muestra un contador de cuatro bits y en la figura 5.7.b un contador de un bit. Este último se empleará para analizar la variación del consumo de energía debido a la señal de reloj en los diferentes esquemas, por lo que hemos elegido un circuito donde la parte combinacional sea despreciable. El análisis de la velocidad de operación, en cambio, lo realizaremos sobre el circuito contador de cuatro bits.

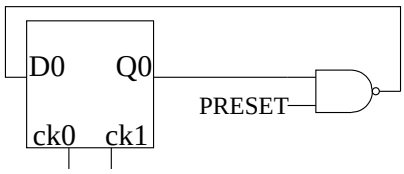
Hemos implementado ambos circuitos usando celdas estándar de una tecnología CMOS de 0.35 $\mu$m de AMS. En concreto, los *latches* empleados son transparentes en nivel bajo. Para comprobar el correcto funcionamiento de las implementaciones se han llevado a cabo simulaciones eléctricas con SPECTRE [48] dentro del entorno Design Framework II de Cadence. Para obtener las formas de onda de las señales de reloj que requieren estos circuitos se ha usado la implementación de los algoritmos presentados en la sección 5.1.

### 5.2.1. Tolerancia al *clock skew* de los esquemas PALACS

La tolerancia al *clock skew* del esquema de temporización *máster-slave* ha sido ampliamente tratada en la literatura y es bien conocido que, para cualquier valor máximo del desplazamiento, existe una frecuencia de reloj para la que este esquema garantiza un buen funcionamiento [3]. En este apartado comprobaremos que los esquemas de reloj PALACS de dos y cuatro fases comparten con el *máster-slave* esta tolerancia al *clock skew* . Para ello seguiremos el siguiente procedimiento:




a)




b)

Figura 5.7: a) Contador de 4 bits b) Contador de 1 bit.

- En primer lugar es necesario realizar un análisis estático de tiempos en el circuito para determinar el valor de los parámetros temporales que permitan obtener formas de onda de las señales de reloj adecuadas en ausencia de *clock skew*.

- En segundo lugar, por simulación eléctrica con SPECTRE, vamos a verificar el correcto funcionamiento en ausencia de *clock skew*.

- En tercer lugar, empleando las mismas formas de onda vamos a incluir un *clock skew* de manera que se observe un mal funcionamiento.

- En cuarto lugar, vamos a medir el *clock skew* y, empleando los algoritmos de cada esquema, vamos a recalcular las formas de onda de las señales de reloj.

- Por último, vamos a simular el circuito con estas nuevas formas de onda para verificar que, efectivamente, vuelve a funcionar correctamente.

El primer paso, análisis temporal de los circuitos, es común a los dos esquemas PALACS. Este análisis temporal lo hemos llevado a cabo dentro del entorno Design Framework II, obteniendo el fichero de retrasos SDF para el circuito. A partir de él hemos obtenido el valor de los siguientes parámetros temporales:

- El retraso máximo del circuito lógico es de 887 ps.

- El retraso de contaminación del circuito lógico es de 372 ps.

- El retraso máximo de una llave de paso cuando se activa siendo su entrada válida es de 98 ps.

- El retraso de contaminación de una llave de paso cuando se activa siendo su entrada válida es de 8 ps.

- El retraso máximo de una llave de paso cuando cambia su entrada encontrándose activada es de 98 ps.

- El retraso máximo de un *latch* cuando cambia su entrada encontrándose su carga activada es de 525 ps.

- El retraso máximo de un *latch* cuando se activa su carga teniendo su entrada un valor válido es de 679 ps.
- El retraso de contaminación del *latch* cuando se activa su carga es de 415 ps.
- El tiempo de *setup* de un *latch* es de 270 ps.
- El tiempo de *hold* de un *latch* es de 0.
- La mínima anchura de pulso alto en una señal de carga de un *latch* que garantiza la correcta carga del dato es de 510 ps.

Para estos esquemas se han usado *latches* que tenían integrada la llave de paso como señal de *output enable*.

Vamos a continuar aplicando el resto de pasos del procedimiento para observar la tolerancia del *clock skew* con el esquema de reloj PALACS de dos fases. El siguiente paso consiste en obtener la forma de onda de las fases de reloj empleando los parámetros temporales y para un *clock skew* nulo. Estas formas de onda se obtienen usando la implementación del algoritmo de la figura 5.4. Se ha simulado el funcionamiento del circuito con estas formas de onda obteniéndose el resultado mostrado en la figura 5.8. Puede observarse que el circuito funciona correctamente.

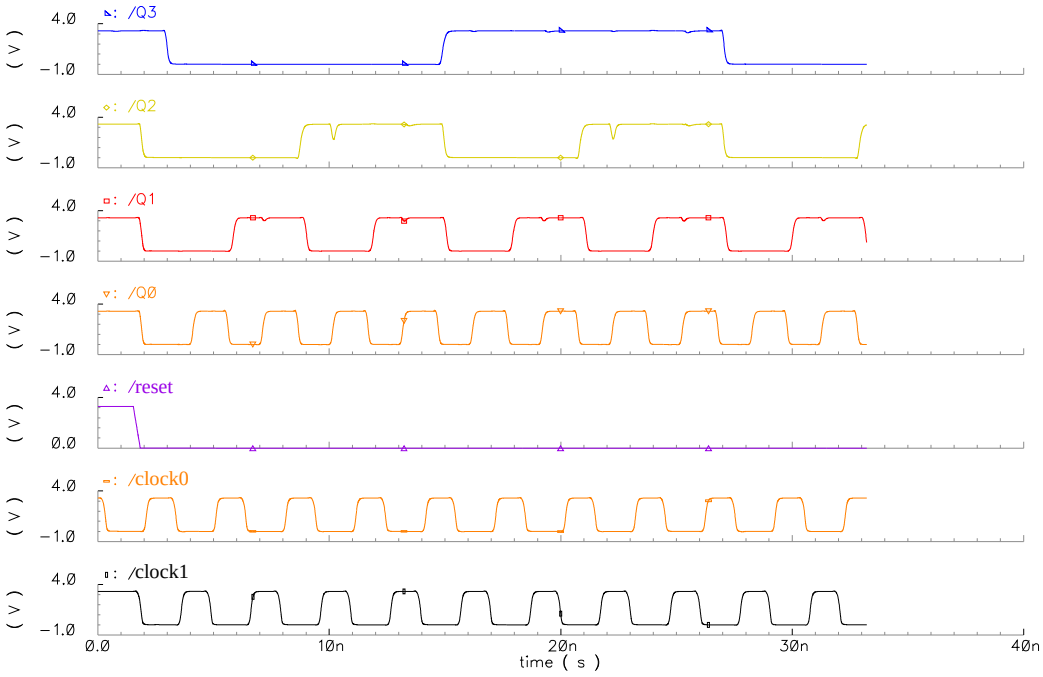



Figura 5.8: Simulación eléctrica del funcionamiento del contador de cuatro bits implementado usando el esquema PALACS de dos fases en ausencia de *clock skew*.

Sin variar la forma de onda de las señales de reloj nominales, introducimos desplazamiento en las señales que controlan los *latches* de los dos bits más significativos haciéndolas pasar por cadenas de *buffers*. Al introducir cuatro *buffers* en la ruta de los relojes, la simulación eléctrica muestra que el circuito deja de funcionar correctamente. Esto puede observarse en la figura 5.9.

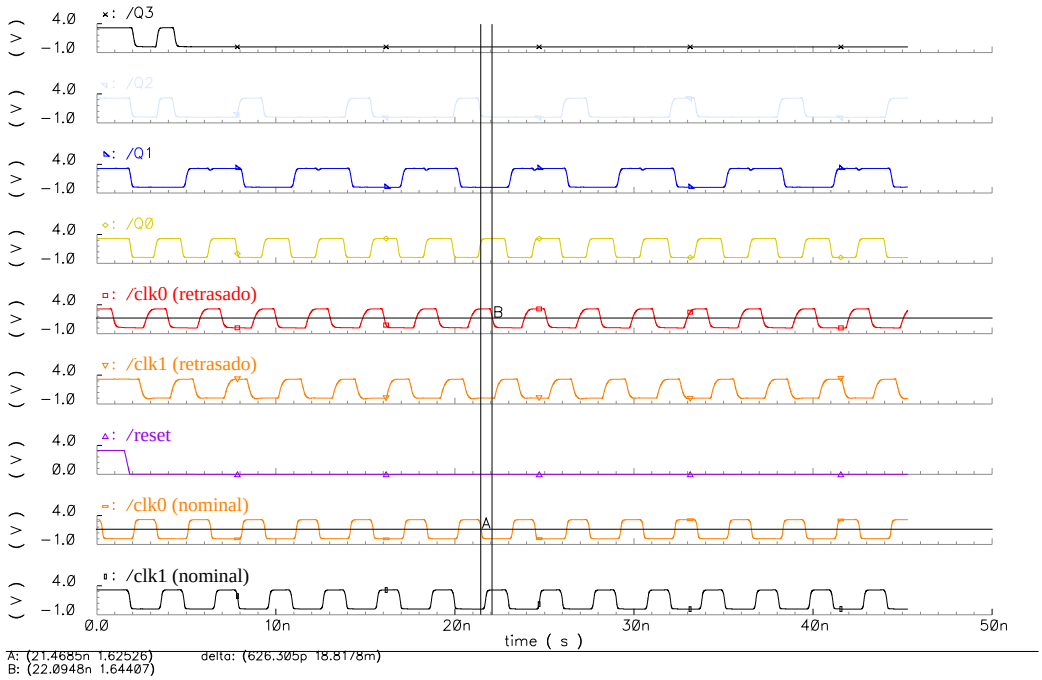



Figura 5.9: Simulación eléctrica del funcionamiento del contador de cuatro bits implementado usando el esquema PALACS de dos fases bajo un *clock skew* que causa mal funcionamiento.

Medimos el desplazamiento introducido y obtuvimos un máximo de 647.69 ps para los flancos de subida y de 633.29 ps para los flancos de bajada. Empleamos de nuevo el algoritmo para encontrar formas de onda para las señales de reloj que toleraran esos desplazamientos y volvimos a simular obteniendo el resultado mostrado en la figura 5.10. A pesar de que hay *glitches*, el circuito funciona correctamente ya que al final de los pulsos activos de cada señal de reloj (nominal o desplazada) las señales tienen el valor correcto.

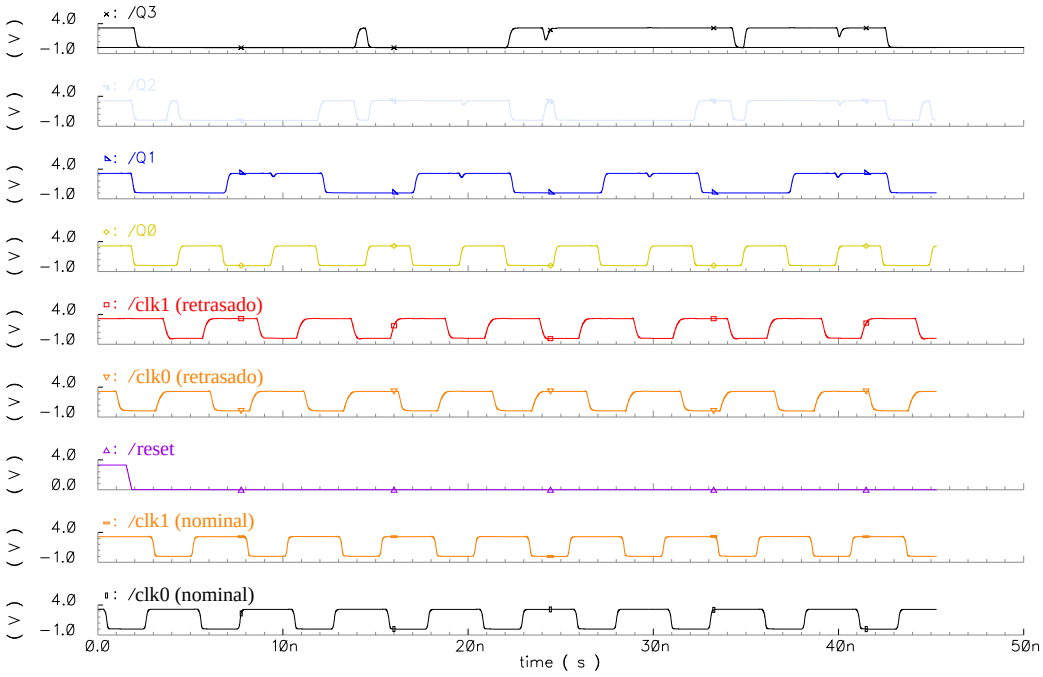



Figura 5.10: Simulación eléctrica del funcionamiento del contador de cuatro bits implementado usando el esquema PALACS de dos fases tolerante a un *clock skew* de cuatro *buffers*.

Comprobemos ahora la tolerancia al *clock skew* del esquema de cuatro fases de reloj. De nuevo se ha aplicado el algoritmo para obtener las formas de onda de las señales de reloj y simulado el funcionamiento del circuito en ausencia de *clock skew* obteniéndose el resultado mostrado en la figura 5.11. Puede observarse que el circuito funciona correctamente.

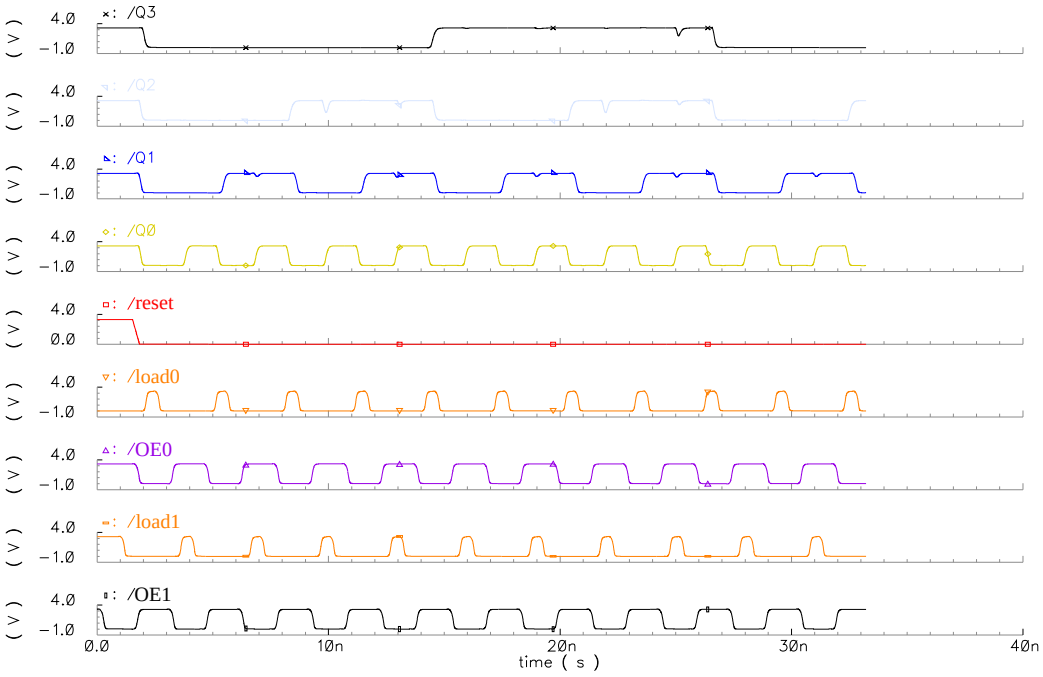



Figura 5.11: Simulación eléctrica del funcionamiento del contador de cuatro bits implementado usando el esquema PALACS de cuatro fases en ausencia de *clock skew*.

De nuevo introducimos desplazamiento en las señales de reloj que controlan los *latches* de los dos bits más significativos haciéndolas pasar por cadenas de *buffers* sin variar la forma de onda de las señales de reloj nominales. Al introducir cuatro *buffers* en la ruta de los relojes, la simulación eléctrica mostró que el circuito dejaba de funcionar correctamente. Esto puede observarse en la figura 5.12. En efecto, el final de un pulso activo en la señal nominal de carga 1 coincide con un pulso *glitch* alto en la señal del dígito 2. En el instante nominal del final de un pulso activo de una señal de carga, las señales de entrada de todos los *latches* deberían tener el valor correcto para que no se viole el tiempo de *setup*.

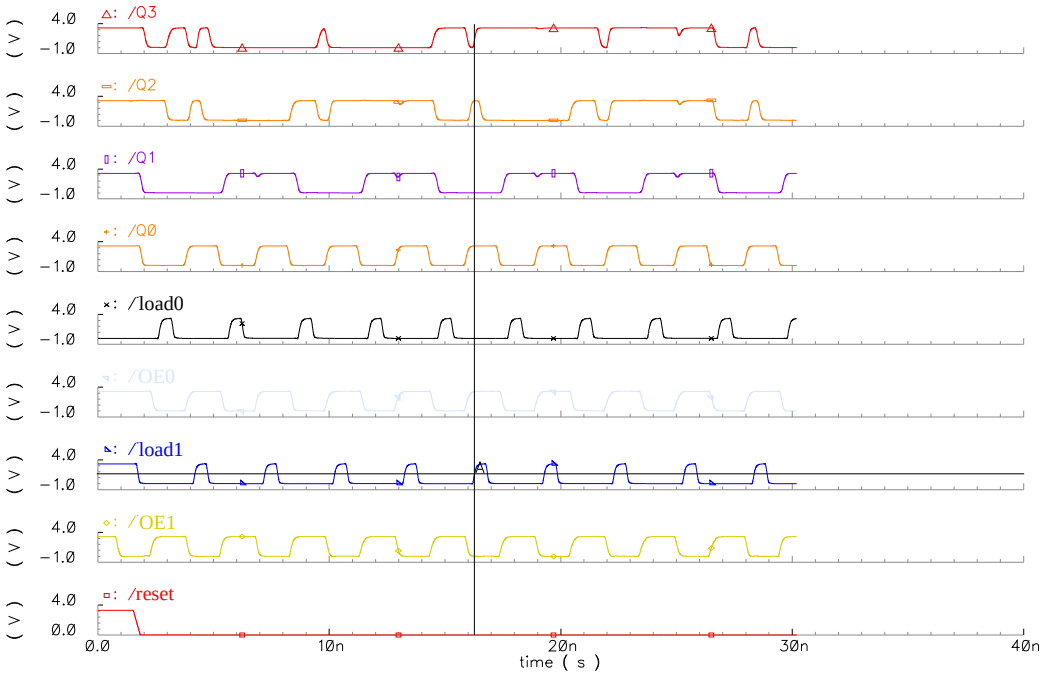



Figura 5.12: Simulación eléctrica del funcionamiento del contador de cuatro bits implementado usando el esquema PALACS de cuatro fases bajo un *clock skew* que causa mal funcionamiento (las señales de reloj mostradas son las nominales).

Para solucionarlo hay que encontrar formas de onda para las señales de reloj que toleren los *clock skew* introducidos aplicando el algoritmo. Se observa que los relojes de habilitación de salida tienen un desplazamiento máximo de 608.9 ps para los flancos de subida y de 599 ps para los flancos de bajada, mientras que los relojes de activación de carga tienen un desplazamiento máximo de 568.5 ps para los flancos de subida y de 570.6 ps para los flancos de bajada. Al simular el circuito con las nuevas señales de reloj obtuvimos el resultado mostrado en la figura 5.13. Las salidas transitorias señaladas son irrelevantes ya que no ocurren cerca del final de un pulso activo de ninguna señal de carga sea cual sea su desplazamiento, de modo que el circuito funciona correctamente.

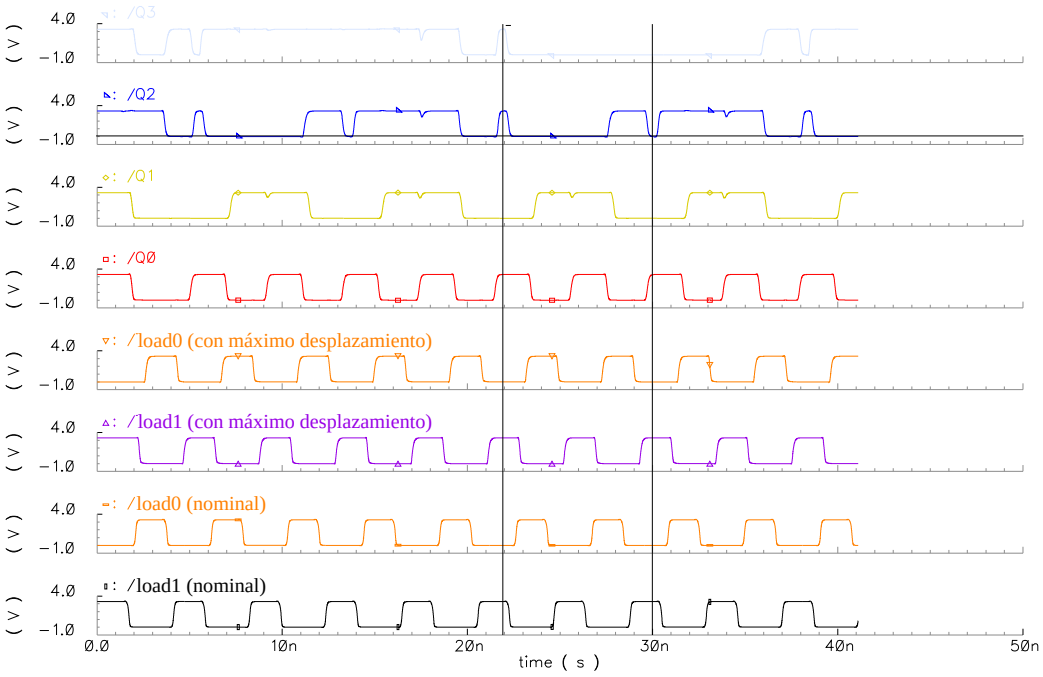



Figura 5.13: Simulación eléctrica del funcionamiento del contador de cuatro bits implementado usando el esquema PALACS de cuatro fases tolerante a un *clock skew* de cuatro *buffers*.

### 5.2.2. Análisis de la velocidad de operación

En este apartado vamos a realizar un análisis comparativo de la máxima frecuencia de operación (mínimo periodo de reloj) del contador de cuatro bits usando los tres esquemas de reloj multifase (*máster-slave*, PALACS y PALACS-4). El procedimiento para obtener el periodo mínimo consiste en aplicar el algoritmo correspondiente a cada esquema de reloj. En este análisis, la implementación del contador para los esquemas PALACS no ha variado respecto al apartado anterior. Para el esquema *máster-slave* se usaron *latches* que carecían de señal de *output enable* pero que disponían de salida en doble raíl. Esto permitió prescindir de los inversores y, por tanto, disminuir el retraso del circuito lógico. Los parámetros del circuito para el esquema *máster-slave* son los siguientes:

- El retraso máximo del circuito lógico es de 776 ps.

- El retraso de contaminación del circuito lógico es de 239 ps.
- El retraso máximo de un *latch máster* cuando cambia su entrada encontrándose su carga activada es de 500 ps.
- El retraso máximo de un *latch máster* cuando se activa su carga teniendo su entrada un valor válido es de 689 ps.
- El retraso de contaminación del *latch máster* cuando se activa su carga es de 415 ps.
- El retraso máximo de un *latch slave* cuando cambia su entrada encontrándose su carga activada es de 594 ps.
- El retraso máximo de un *latch slave* cuando se activa su carga teniendo su entrada un valor válido es de 769 ps.
- El retraso de contaminación del *latch slave* cuando se activa su carga es de 444 ps.
- El tiempo de *setup* de un *latch* es de 190 ps.
- El tiempo de *hold* de un *latch* es de 0.
- La mínima anchura de pulso alto en una señal de carga de un *latch* que garantiza la correcta carga del dato es de 370 ps.

Como ya hemos comprobado en el apartado anterior, el periodo mínimo depende del *clock skew*. Así, para un $t_{skew} = 0$ el contador de cuatro bits puede alcanzar una frecuencia de cómputo de 534 MHz con el esquema *máster-slave*, mientras que con los esquemas PALACS puede alcanzar un frecuencia de 662 MHz. Esto supone una mejora de un 24 % respecto al esquema *máster-slave*. Para ver como afecta el *clock skew* a la velocidad de operación del contador se han obtenido las frecuencias de cómputo que pueden conseguirse con cada esquema con *clock skew* entre 0 y $T_0$, siendo $T_0$ la inversa de la frecuencia de cómputo máxima alcanzable con los esquemas PALACS. Esto se ha hecho aplicando de forma reiterada los algoritmos suponiendo que el desplazamiento máximo de todas las señales de reloj es el

mismo y que el desplazamiento máximo en transiciones de subida y bajada es idéntico. El resultado se muestra en la figura 5.14.

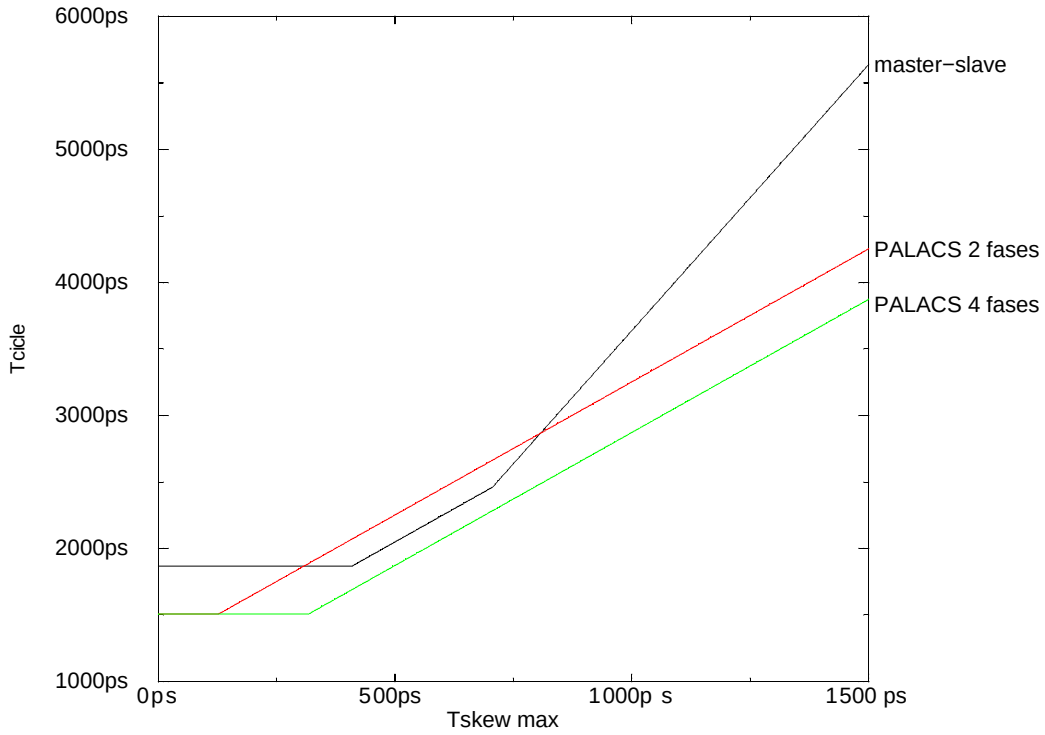



Figura 5.14: Periodo de cómputo frente a *clock skew* para distintos esquemas de reloj aplicados al contador de cuatro bits.

Como ya hemos visto, el tiempo mínimo de ciclo de cómputo para los esquemas PALACS es de 1510 ps. Esto es lógico dado que ese tiempo es precisamente la suma del retraso máximo de un *latch* con su llave de paso y el retraso del circuito combinacional. Por otro lado el tiempo de ciclo mínimo alcanzable con el esquema *máster-slave* es de 1870 ps, que es precisamente la suma de los retrasos máximos de un *latch máster*, un *latch slave* y el circuito combinacional.

Pueden observarse cambios en la pendiente de la curva de cada esquema de temporización. En el esquema PALACS de dos fases el cambio de pendiente se produce cuando $CLK_a[i] + t_{skewa} + K_{cmax}$ comienza a ser mayor que $Q[i-1] + K_{imax}$ mientras que en el de cuatro fases se produce cuando $OE_a[i] + t_{skewaOE} + K_{cmax}$ comienza a ser mayor que $Q[i-1] + K_{imax}$. En

el esquema *máster-slave* se producen dos cambios de pendiente: uno cuando $CLK_{0a}[i] + t_{ske0a} + pw_{minslave}$ comienza a ser mayor que $QM[i] + t_{setupslave}$ y otro cuando $CLK_{1a}[i] + t_{skew1a} + L_{masterCQmax}$ comienza a ser mayor que $NS[i-1] + L_{masterDQmax}$.

> Como puede verse, el esquema PALACS de cuatro fases se alcanza siempre la mayor velocidad, y solo en un pequeño rango el esquema *máster-slave* resulta ligeramente más rápido que el PALACS de dos fases. No obstante, para valores altos del *clock skew* ambos esquemas PALACS presentan una mejor evolución y proporcionan una velocidad de operación significativamente más alta, tendiendo a una mejora del 100 % respecto al esquema *máster-slave*.

### 5.2.3. Análisis del consumo de potencia

Como ya mencionamos en el capítulo 4, el esquema de temporización PALACS de dos fases de reloj permitiría presumiblemente disminuir el consumo de potencia del circuito respecto del esquema *máster-slave*. La razón de este ahorro es doble:

- Con el esquema PALACS en cada ciclo de cómputo cambian de estado la mitad de los *latches* que en el esquema *máster-slave*.
- El esquema PALACS de dos fases tiene el mismo número de señales de reloj que el *máster-slave*, pero su frecuencia se reduce a la mitad para el mismo ratio de cómputo. Por lo tanto la potencia disipada por la red de distribución del reloj también disminuye.

Para comprobar el ahorro de potencia debido a los *latches* se ha medido por simulación eléctrica el consumo de potencia del contador de cuatro bits de la figura 5.7 a distintas frecuencias de cómputo usando los esquemas *máster-slave* y PALACS de dos fases. Los resultados se muestran en la tabla 5.1. Puede observarse una mejora de entre un 6 % a un 10 % al usar el esquema PALACS. Aunque es una mejora apreciable, no se acerca al 50 % debido a que el consumo de la parte combinacional es similar en ambos esquemas.

| Frecuencia de cómputo | Consumo con *máster-slave* | Consumo con PALACS-2 | Ahorro del PALACS respecto al *máster-slave* |
|---|---|---|---|
| 25 MHz | 159 $\mu W$ | 145 $\mu W$ | 8,8 % |
| 100 MHz | 635 $\mu W$ | 576 $\mu W$ | 9,3 % |
| 200 MHz | 1269 $\mu W$ | 1149 $\mu W$ | 9,5 % |
| 348,8 MHz | 2210 $\mu W$ | 1988 $\mu W$ | 10 % |
| 422,12 MHz | 2673 $\mu W$ | 2421 $\mu W$ | 9,4 % |
| 534,76 MHz | 3369$\mu W$ | 3140$\mu W$ | 5,9 % |

Tabla 5.1: Consumo del contador de cuatro bits sin incluir la red de distribución de reloj.

En un circuito secuencial síncrono en el que la mayor parte del consumo se deba a los cambios de estado de los *latches* y no al consumo de la lógica combinacional, es de esperar que el ahorro sea próximo al 50 %. Para comprobarlo se ha implementado el contador de un bit, en el que la parte combinacional se reduce a una puerta, y, por simulación, se ha medido su consumo para distintas frecuencias de cómputo obteniéndose los resultados de la tabla 5.2. Como se esperaba, la mejora ronda el 50 %.

| Frecuencia de cómputo | Consumo con *máster-slave* | Consumo con PALACS-2 | Ahorro del PALACS respecto al *máster-slave* |
|---|---|---|---|
| 25 MHz | 45 $\mu W$ | 12 $\mu W$ | 74 % |
| 50 MHz | 90 $\mu W$ | 48 $\mu W$ | 47 % |
| 100 MHz | 190 $\mu W$ | 100 $\mu W$ | 48 % |
| 200 MHz | 359 $\mu W$ | 183 $\mu W$ | 49 % |

Tabla 5.2: Consumo del contador de un bit sin incluir la red de distribución de reloj.

Para comprobar el ahorro de potencia debido a la red de distribución de reloj, esta se ha implementado con una cadena de cuatro inversores y se ha aplicado de nuevo simulación eléctrica al contador de cuatro bits. Los resultados pueden observarse en la tabla 5.3. Al tener en cuenta el consumo

de la red de distribución de reloj se observa que la mejora es notablemente mayor que la contemplada en la tabla 5.1, siendo superior al 20 % para todas las frecuencias.

| Frecuencia de cómputo | Consumo con *máster-slave* | Consumo con PALACS-2 | Ahorro del PALACS respecto al *máster-slave* |
|---|---|---|---|
| 25 MHz | 245 $\mu W$ | 194 $\mu W$ | 21 % |
| 50 MHz | 489 $\mu W$ | 388 $\mu W$ | 21 % |
| 100 MHz | 978 $\mu W$ | 774 $\mu W$ | 21 % |
| 200 MHz | 1954 $\mu W$ | 1547 $\mu W$ | 21 % |
| 500 MHz | 4877 $\mu W$ | 3860 $\mu W$ | 21 % |

Tabla 5.3: Consumo del contador de cuatro bits incluida la red de distribución de reloj.

# Capítulo 6

# Conclusiones

En la primera parte de esta tesis hemos podido constatar que el usar puertas estáticas CMOS usando terminales *body* independientes (INBO) permite mejorar notablemente el consumo estático y dinámico así como la velocidad de las puertas lógicas CMOS estáticas. Además, las puertas INBO presentan un consumo estático y un retraso pin-a-pin más uniforme a lo largo de sus entradas, lo que es deseable de cara a la síntesis automatizada de circuitos lógicos.

Las mejoras introducidas en la implementación INBO se obtienen a costa de una mayor área. Podría argumentarse que los transistores de las puertas de diseño convencional podrían redimensionarse hasta que ocupasen el área de sus homólogas INBO para aumentar su conductividad y velocidad pero esto es engañoso ya que al aumentar el tamaño de los transistores de una puerta se aumenta también su capacidad de entrada, de modo que las puertas que atacan dicha capacidad (es decir, aquellas cuya salida se ha conectado a las capacidades de entrada sobredimensionadas) aumentarían su consumo y reducirían su velocidad. Esto no ocurre al usar puertas INBO dado que su capacidad de entrada no aumenta respecto a sus homólogas COBO (aunque su tamaño sea mayor). Es más, dado que los terminales *body* de los transistores de las puertas INBO no están polarizados sus capacidades puerta-*body* se cargan a un menor voltaje, con lo que la velocidad de las puertas que generan las entradas de la puerta INBO mejora.

A pesar de todo lo expuesto, la penalización de área introducida por las puertas INBO es muy severa, por lo que resultaría muy costoso que reemplazasen a las puertas tradicionales en todo el diseño. En lugar de eso, se propone usar las puertas INBO selectivamente en caminos críticos, nodos con alta actividad de conmutación o cuando se requieran puertas de gran número de entradas. Los algoritmos actuales de diseño incremental podrían aplicarse para hacer esta sustitución selectiva [49]. Esto permitiría mejorar de forma significativa las prestaciones del circuito con un coste aceptable.

Con respecto al estudio de los esquemas de sincronización, en la segunda parte de la tesis se han propuesto dos nuevos esquemas de sincronización basados en *latches* que operan de forma alternativa (PALACS) empleando dos o cuatro fases de reloj. Hemos establecido un procedimiento para obtener las formas de onda de máxima frecuencia de operación para un determinado *clock skew* tanto en estos esquemas como en el esquema *máster-slave*. Asimismo, hemos verificado que, efectivamente, los esquemas PALACS propuestos son tolerantes al *clock skew*.

Por otra parte, hemos realizado un análisis comparativo de la máxima frecuencia de operación para los esquemas propuestos y el esquema *máster-slave*. Los esquemas PALACS ofrecen una mayor velocidad de operación para prácticamente cualquier valor de *skew*, tendiendo, para valores altos del mismo, a una mejora del 100 % respecto al *máster-slave*.

Por último, hemos comprobado como, efectivamente, los esquemas PALACS suponen un ahorro muy significativo del consumo de energía, fundamentalmente porque reducen la potencia consumida en la red de distribución de reloj, aspecto cada vez más importante en las tecnologías actuales.

# Apéndice A

# Publicaciones resultantes de los trabajos realizados

## A.1. Publicaciones en revistas

- D. Guerrero, M. J. Bellido, J. Juan, A. Millan, P. Ruiz-de-Clavijo, E. Ostua and J. Viejo, "Automated performance evaluation of skew-tolerant clocking schemes". International Journal of Electronics, Volume 93, Issue 12 December 2006, pages 819 - 842

- D. Guerrero, A. Millan, J. Juan, M. J. Bellido, P. Ruiz-de-Clavijo, E. Ostua and J. Viejo, "Static Power Consumption in CMOS Gates Using Independent Bodies". Lecture Notes in Computer Science, 2007, Volume 4644/2007, 404-412, DOI: 10.1007/978-3-540-74442-9_39

- D. Guerrero, A. Millan, J. Juan, M. J. Bellido, P. Ruiz-de-Clavijo, E. Ostua and J. Viejo, "Improving the Performance of Static CMOS Gates by Using Independent Bodies". Journal of Low Power Electronics, Volume 3, Number 1, April 2007, pp. 70-77, DOI: 10.1166/jolpe.2007.120

- D. Guerrero, A. Millan, J. Juan, M. J. Bellido, P. Ruiz-de-Clavijo, E. Ostua and J. Viejo, "Studying the viability of static CMOS gates with a large number of inputs when using separate transistor wells". Journal

of Low Power Electronics, Volume 7, Number 3, August 2011, pp. 444-452, DOI: 10.1166/jolpe.2011.1145

## A.2. Aportaciones en actas de congresos

- D. Guerrero, M. J. Bellido, J. Juan, P. Ruiz-de-Clavijo and A. Millan, "Two Phase Alternating Latches Clocking Scheme for CMOS Sequential Circuits". Proceedings of the XVII, Conference on Design of Circuits and Integrated Systems, DCIS 2002. Universidad de Cantabria, Santander, Spain. 2002. Pag. 159-162. ISBN: 84-8102-311-6

- D. Guerrero, M. J. Bellido, J. Juan, A. Millan, P. Ruiz-de-Clavijo and E. Ostua, "Four Phase Alternating Latches Clocking Scheme for CMOS Sequential Circuits". Proceedings of the XIX Conference on Design of Circuits and Integrated Systems, DCIS 2004. Bordeaux, France. 2004. Pag. 780-783

- D. Guerrero, M. J. Bellido, J. Juan, A. Millan, P. Ruiz-de-Clavijo, E. Ostua and J. Viejo, "Algorithms to Get the Maximum Operation Frequency for Skew-Tolerant Clocking Schemes". Proceedings of SPIE International Symposium on Microtechnologies for the New Millenium (VLSI Circuits and Systems II). Conference on VLSI Circuits and Systems II. Seville, Spain. 2005. SPIE- the International Society for Optical Engineering. Pag. 467-478. ISBN: 0-8194-5832-5

- D. Guerrero, M. J. Bellido, J. Juan, "CMOS Digital Design Techniques for Low Power and High Speed". Phd Forum. Date Conference. Munich, Germany. ACM Sigda. 2005

- D. Guerrero, A. Millan, J. Juan, M. J. Bellido, P. Ruiz-de-Clavijo, E. Ostua and J. Viejo, "The Effect of Using Separated Bodies Over Static Power Consumption in Static Bulk-CMOS Gates". Proceedings of the XXII Conference on Design of Circuits and Integrated Systems, DCIS 2007. Seville, Spain. 2007. Pag. 181-185. ISBN: 978-84-690-86

- D. Guerrero, M. J. Bellido, J. Juan, P. Ruiz, A. Millan, E. Ostua and J. Viejo, "Síntesis Lógica Automatizada para esquemas de Temporización de Latches Alternantes". XIII Workshop IBERCHIP. Lima, Peru. 2007. Pag. 349-350

- D. Guerrero, M. Bellido, J. Juan, A. Millan, P. Ruiz, E. Ostua and J. Viejo, "Automatic Logic Synthesis for Parallel Alternating Latches Clocking Schemes". VLSI Circuits and Systems Conference SPIE. Maspalomas, Spain. 2007. SPIE- the International Society for Optical Engineering

- D. Guerrero, A. Millan, J. Juan, M. J. Bellido, J. Viejo and A. Muñoz, "Using Independent Bodies in Bulk-CMOS Gates". Cool Chips XI Proceedings. IEEE Cool Chips Num. 11. Pag. 121-121. Yokohama, Japan. 2008

- D. Guerrero, A. Millan, J. Juan, M. J. Bellido, P. Ruiz-de-Clavijo and E. Ostua, "Delay and Power Consumption of Static Bulk-CMOS Gates Using Independent Bodies". Faible Tension Faible Consommation. Num. 7. Pag. 105-110. Université catholique de Louvain, Louvain-la-Neuve, Belgium. 2008

- D. Guerrero, A. Millan, J. Juan, M. J. Bellido, P. Ruiz-de-Clavijo and E. Ostua, "Performance Analysis of Bulk-CMOS Gates Using Separated Wells". XV Workshop IBERCHIP. Pag. 54-59. Workshop IBERCHIP. Buenos Aires, Argentina. 2009

- D. Guerrero, A. Millan, J. Juan, M. J. Bellido, P. Ruiz-de-Clavijo and E. Ostua, "Delay and Power Consumption of Static Bulk-CMOS Gates Using Independent Bodies". IEEE DTIS 2009. ISBN: 978-1-4244-43. Num. 4. Pag. 191-196. Cairo, Egypt

# Apéndice B

# *Layouts* de las puertas diseñadas

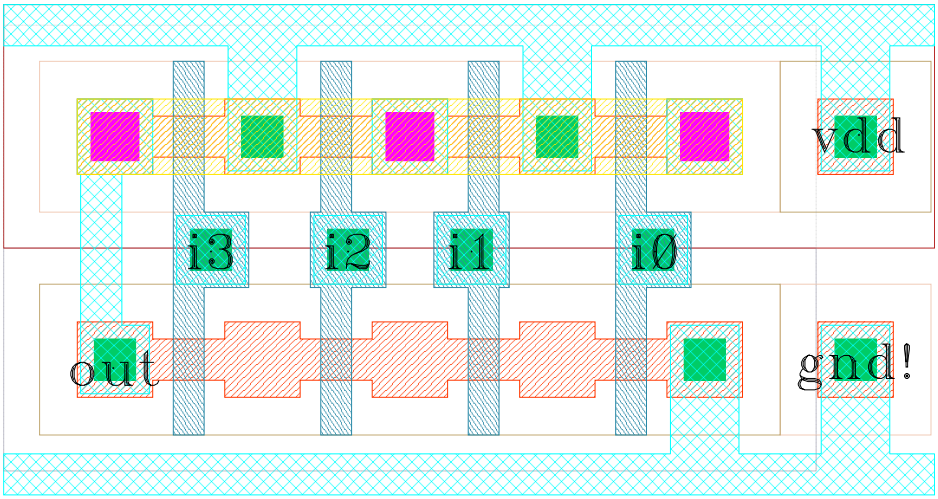



Figura B.1: Layout de una puerta NAND de cuatro entradas con implementación COBO.

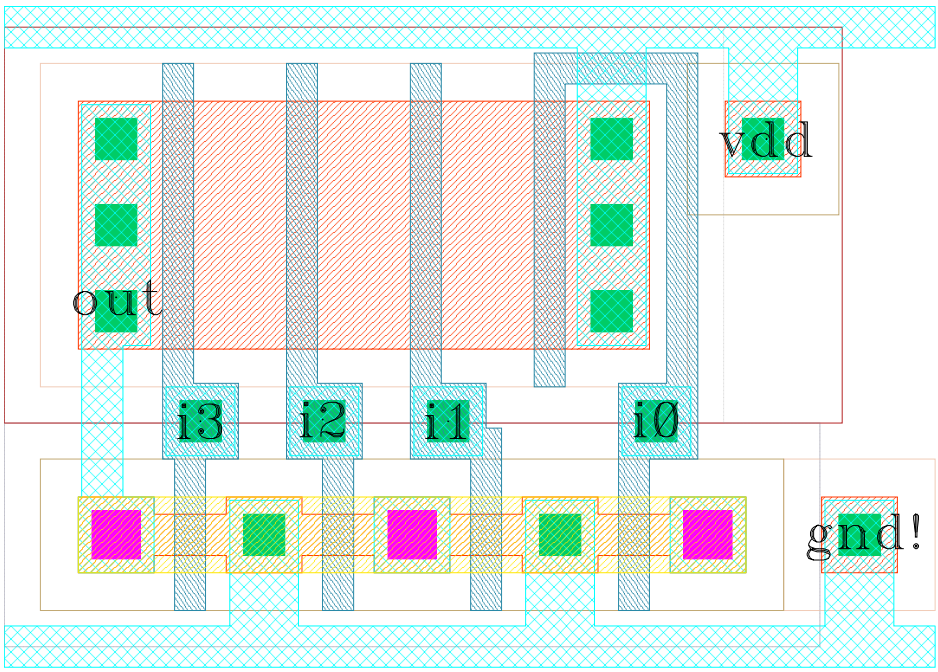



Figura B.2: Layout de una puerta NOR de cuatro entradas con implementación COBO.

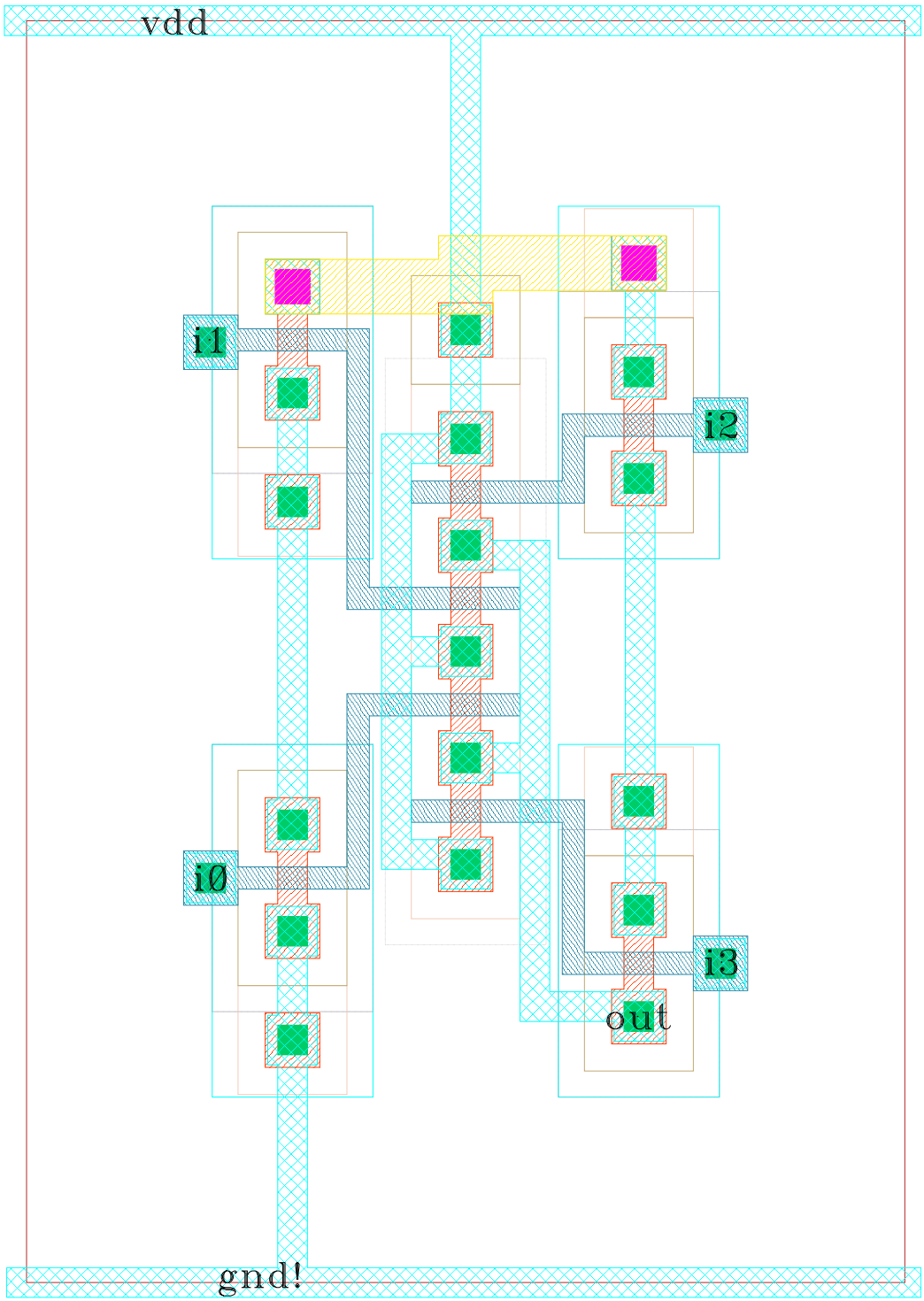



Figura B.3: Layout de una puerta NAND de cuatro entradas con implementación INBO.

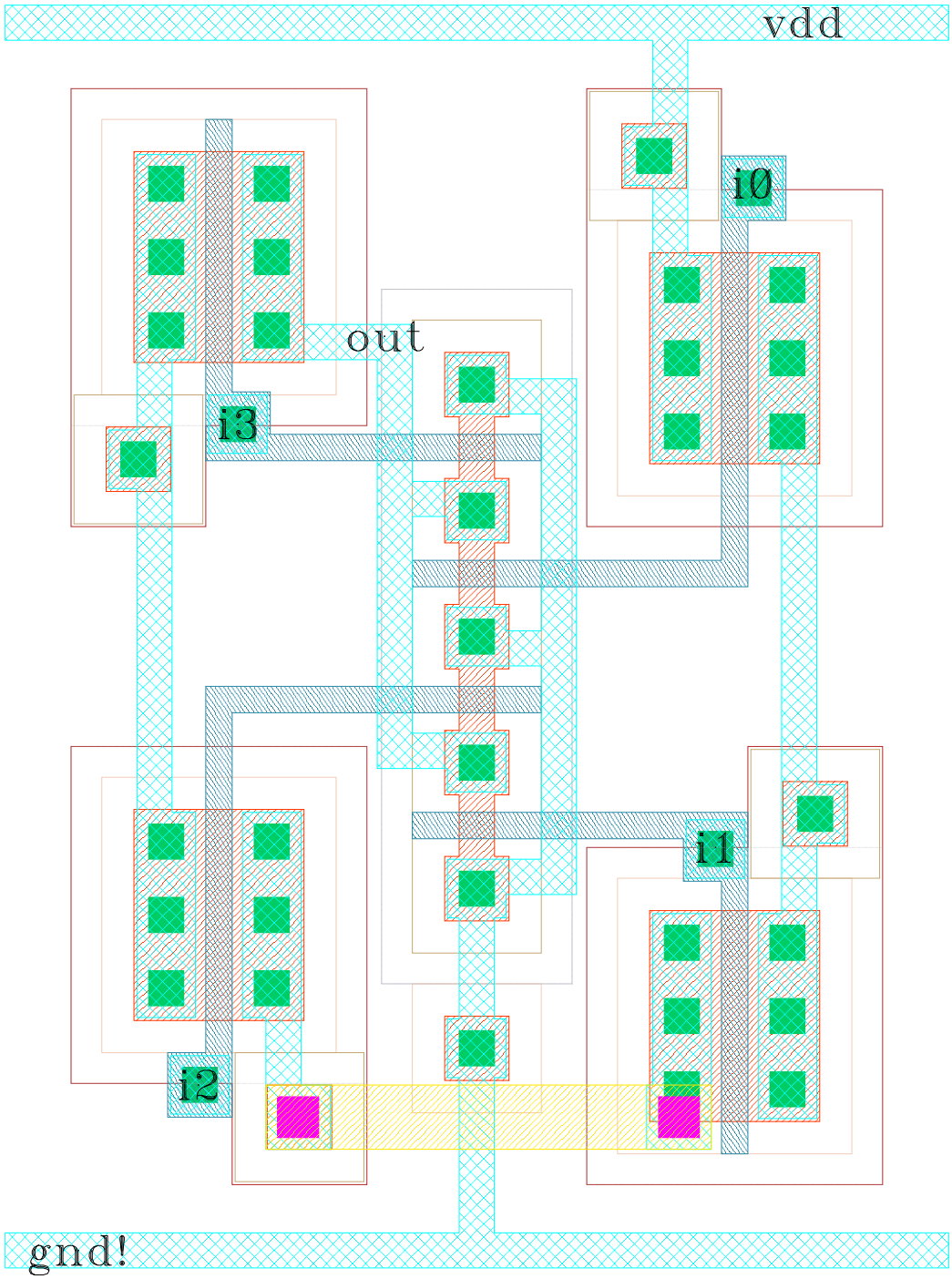



Figura B.4: Layout de una puerta NOR de cuatro entradas con implementación INBO.

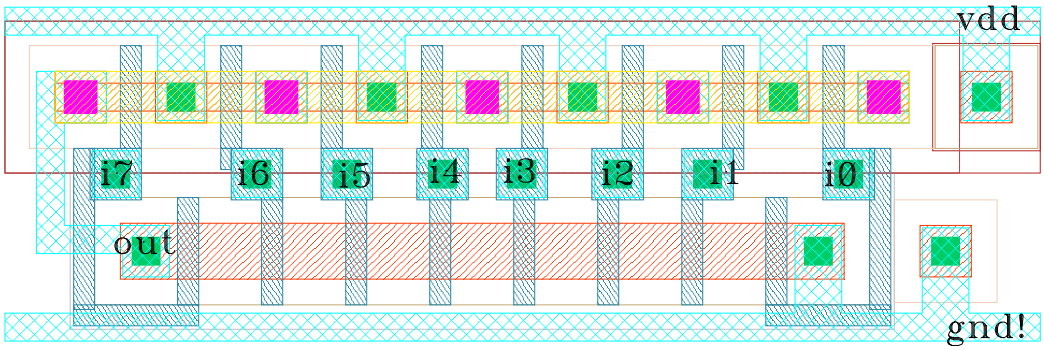



Figura B.5: Layout de una puerta NAND de ocho entradas con implementación COBO.

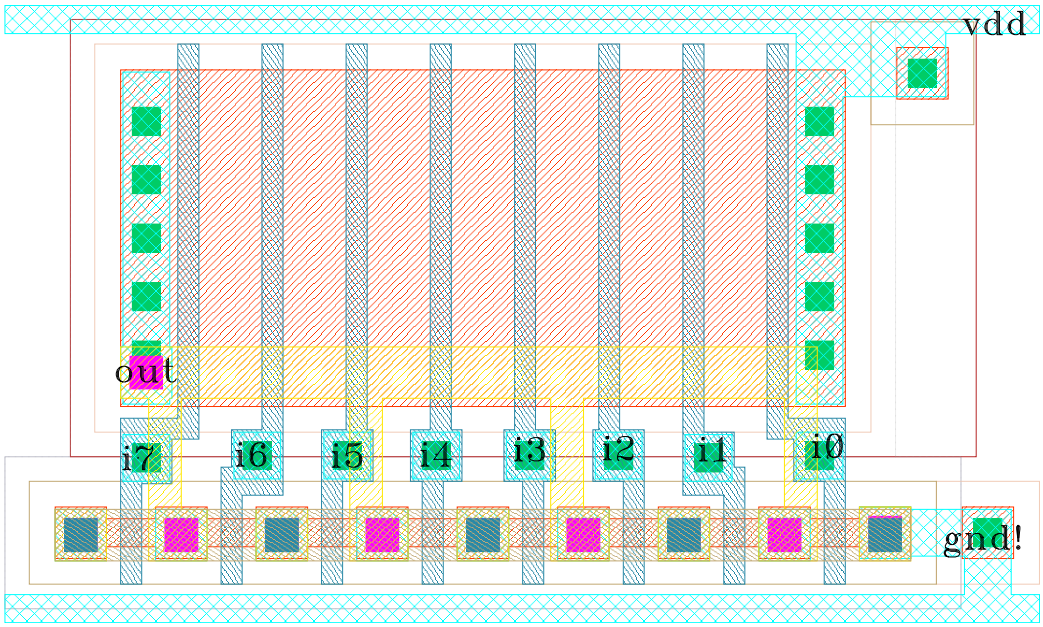



Figura B.6: Layout de una puerta NOR de ocho entradas con implementación COBO.

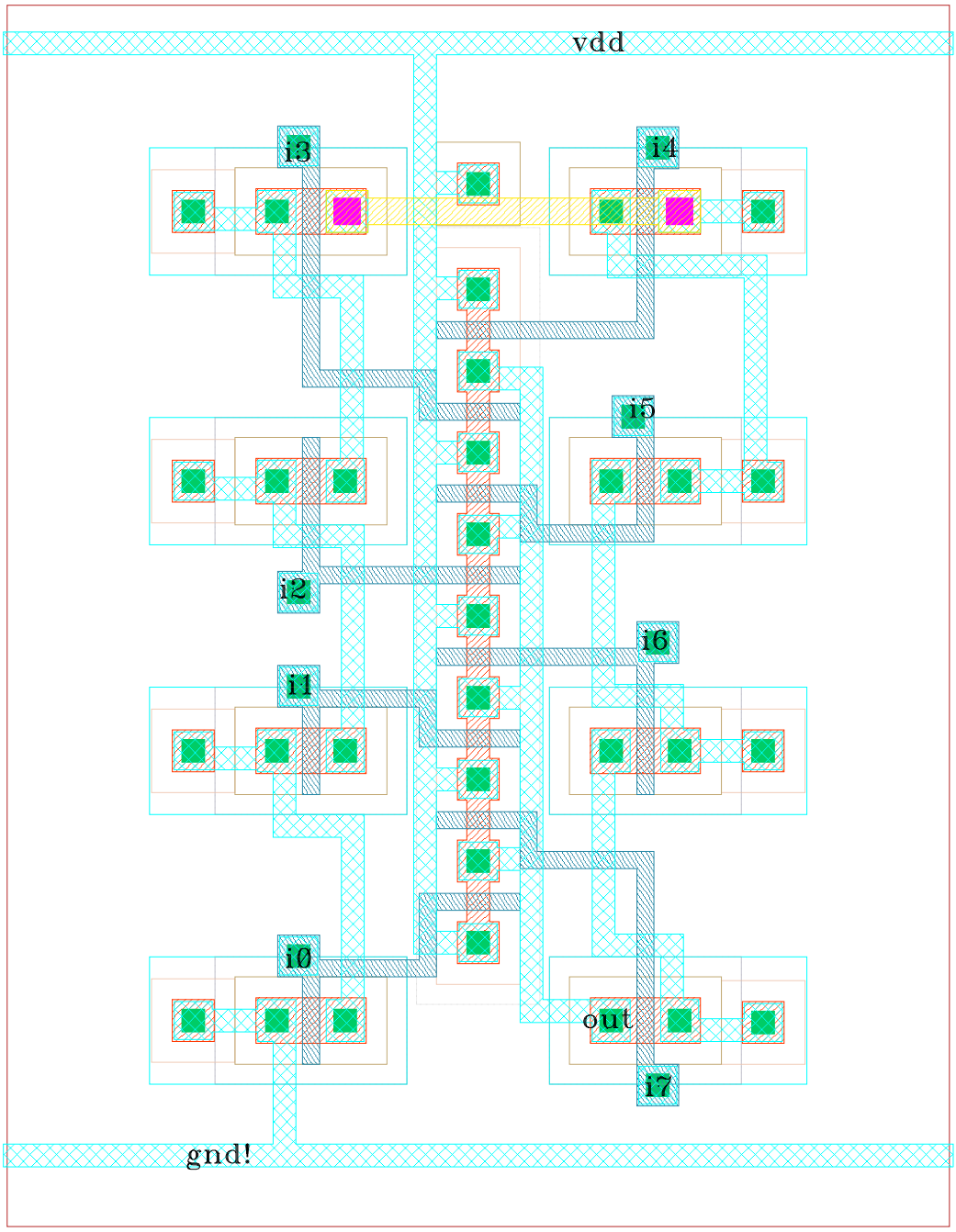



Figura B.7: Layout de una puerta NAND de ocho entradas con implementación INBO.

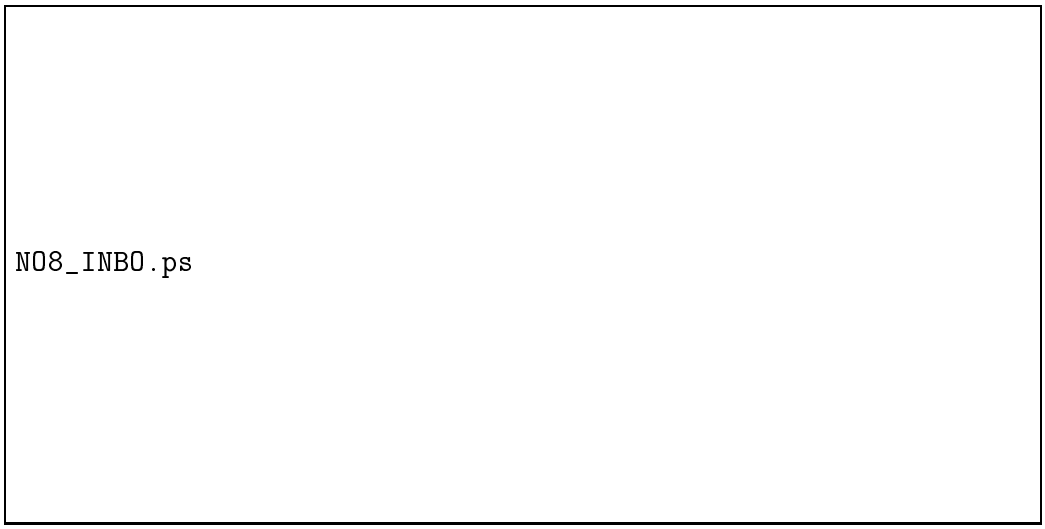



Figura B.8: Layout de una puerta NOR de ocho entradas con implementación INBO.

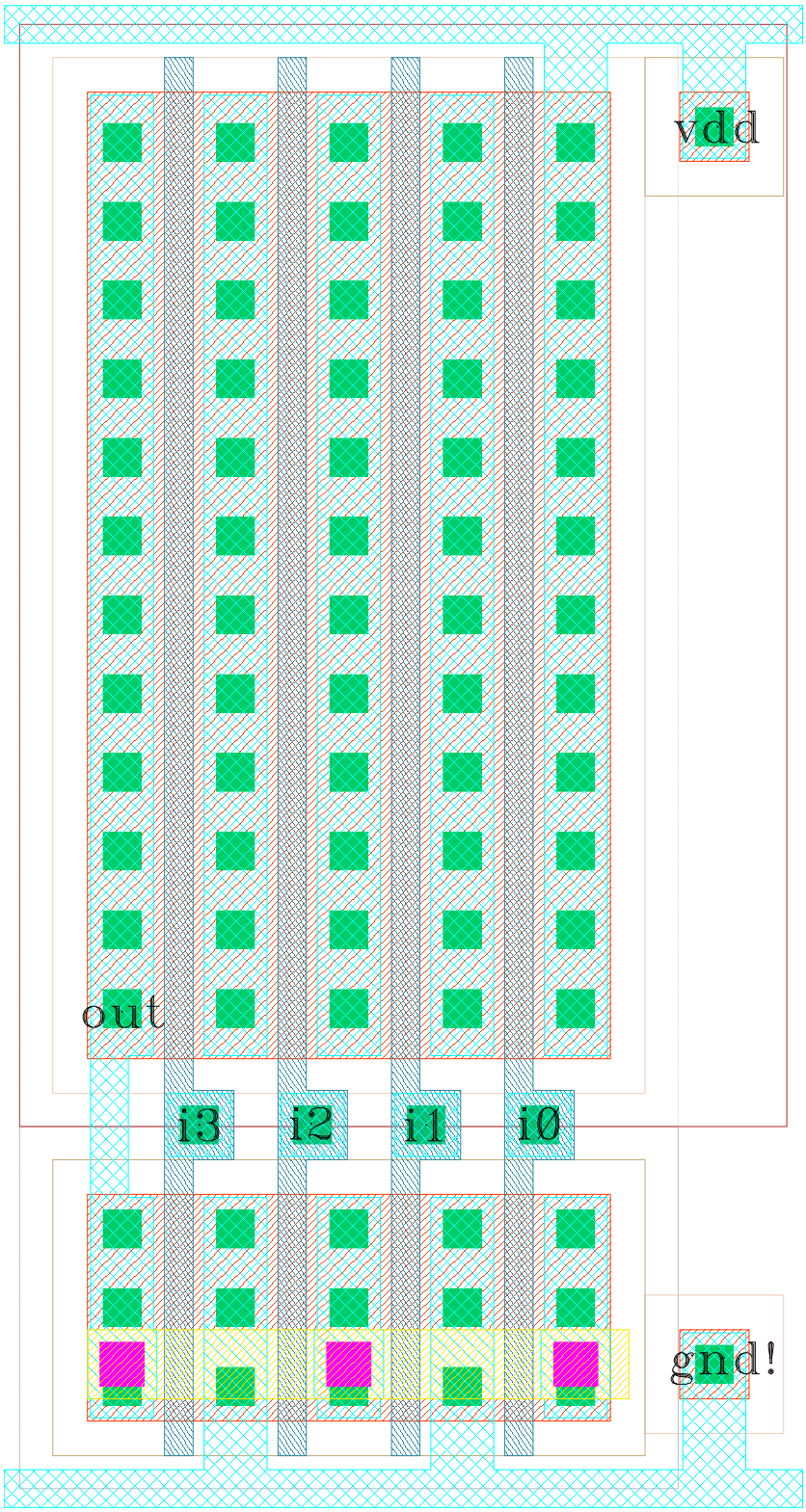



Figura B.9: Layout de una puerta NOR de cuatro entradas con implementación COBO sobredimensionada.

# Apéndice C

# Resultados de simular las puertas no sobredimensionadas

## C.1. Puertas de cuatro entradas

### C.1.1. retraso pin a pin

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 197 ps | 184 ps | 153 ps | 109 ps |
| 2 | 238 ps | 225 ps | 194 ps | 147 ps |
| 3 | 279 ps | 265 ps | 234 ps | 186 ps |
| 4 | 319 ps | 306 ps | 275 ps | 227 ps |

Tabla C.1: Retraso de subida de la puerta NOR COBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 159 ps | 149 ps | 129 ps | 97 ps |
| 2 | 179 ps | 168 ps | 148 ps | 117 ps |
| 3 | 197 ps | 186 ps | 166 ps | 136 ps |
| 4 | 215 ps | 203 ps | 183 ps | 154 ps |

Tabla C.2: Retraso de bajada de la puerta NOR COBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 151 ps | 143 ps | 124 ps | 97 ps |
| 2 | 187 ps | 179 ps | 160 ps | 131 ps |
| 3 | 223 ps | 215 ps | 196 ps | 166 ps |
| 4 | 258 ps | 250 ps | 231 ps | 201 ps |

Tabla C.3: Retraso de subida de la puerta NOR INBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 117 ps | 112 ps | 102 ps | 86 ps |
| 2 | 136 ps | 131 ps | 121 ps | 106 ps |
| 3 | 153 ps | 148 ps | 138 ps | 124 ps |
| 4 | 170 ps | 164 ps | 155 ps | 141 ps |

Tabla C.4: Retraso de bajada de la puerta NOR INBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 191 ps | 177 ps | 160 ps | 141 ps |
| 2 | 212 ps | 198 ps | 181 ps | 163 ps |
| 3 | 233 ps | 218 ps | 202 ps | 184 ps |
| 4 | 253 ps | 238 ps | 222 ps | 204 ps |

Tabla C.5: Retraso de subida de la puerta NAND COBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 116 ps | 113 ps | 106 ps | 99 ps |
| 2 | 135 ps | 131 ps | 125 ps | 118 ps |
| 3 | 153 ps | 150 ps | 144 ps | 136 ps |
| 4 | 171 ps | 168 ps | 162 ps | 154 ps |

Tabla C.6: Retraso de bajada de la puerta NAND COBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 154 ps | 146 ps | 132 ps | 127 ps |
| 2 | 177 ps | 169 ps | 155 ps | 150 ps |
| 3 | 199 ps | 191 ps | 177 ps | 172 ps |
| 4 | 221 ps | 213 ps | 199 ps | 194 ps |

Tabla C.7: Retraso de subida de la puerta NAND INBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 98 ps | 95 ps | 89 ps | 87 ps |
| 2 | 117 ps | 115 ps | 109 ps | 107 ps |
| 3 | 136 ps | 134 ps | 129 ps | 127 ps |
| 4 | 155 ps | 153 ps | 148 ps | 146 ps |

Tabla C.8: Retraso de bajada de la puerta NAND INBO de cuatro entradas.

### C.1.2. consumo dinámico

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 69 fJ | 61 fJ | 51 fJ | 35 fJ |
| 2 | 80 fJ | 72 fJ | 61 fJ | 48 fJ |
| 3 | 94 fJ | 87 fJ | 75 fJ | 62 fJ |
| 4 | 106 fJ | 99 fJ | 88 fJ | 74 fJ |

Tabla C.9: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NOR COBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 56 fJ | 53 fJ | 45 fJ | 35 fJ |
| 2 | 68 fJ | 64 fJ | 58 fJ | 49 fJ |
| 3 | 82 fJ | 78 fJ | 71 fJ | 63 fJ |
| 4 | 95 fJ | 92 fJ | 82 fJ | 76 fJ |

Tabla C.10: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NOR INBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 26 fJ | 24 fJ | 21 fJ | 19 fJ |
| 2 | 31 fJ | 29 fJ | 25 fJ | 23 fJ |
| 3 | 36 fJ | 33 fJ | 30 fJ | 27 fJ |
| 4 | 41 fJ | 38 fJ | 35 fJ | 32 fJ |

Tabla C.11: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NAND COBO de cuatro entradas.

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 22 fJ | 20 fJ | 18 fJ | 17 fJ |
| 2 | 27 fJ | 25 fJ | 22 fJ | 22 fJ |
| 3 | 32 fJ | 30 fJ | 28 fJ | 27 fJ |
| 4 | 37 fJ | 36 fJ | 33 fJ | 32 fJ |

Tabla C.12: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NAND INBO de cuatro entradas.

### C.1.3. producto energía-retraso

| Fan-out | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 13,60 | 11,18 | 7,85 | 3,85 |
| 2 | 19,14 | 16,25 | 11,85 | 7,11 |
| 3 | 26,14 | 23,02 | 17,62 | 11,61 |
| 4 | 33,85 | 30,35 | 24,08 | 16,89 |

Tabla C.13: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NOR COBO de cuatro entradas (medido en picojulio*picosegundo).

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 8,49 | 7,57 | 5,54 | 3,42 |
| 2 | 12,78 | 11,50 | 9,26 | 6,48 |
| 3 | 18,18 | 16,75 | 13,99 | 10,41 |
| 4 | 24,48 | 22,92 | 19,05 | 15,36 |

Tabla C.14: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NOR INBO de cuatro entradas (medido en picojulio*picosegundo).

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 5,04 | 4,18 | 3,32 | 2,65 |
| 2 | 6,60 | 5,75 | 4,61 | 3,67 |
| 3 | 8,36 | 7,20 | 6,03 | 5,02 |
| 4 | 10,34 | 8,96 | 7,83 | 6,59 |

Tabla C.15: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NAND COBO de cuatro entradas (medido en picojulio*picosegundo).

| *Fan-out* | entrada 0 | entrada 1 | entrada 2 | entrada 3 |
|---|---|---|---|---|
| 1 | 3,41 | 2,97 | 2,39 | 2,20 |
| 2 | 4,78 | 4,28 | 3,47 | 3,30 |
| 3 | 6,32 | 5,76 | 5,01 | 4,67 |
| 4 | 8,28 | 7,58 | 6,63 | 6,27 |

Tabla C.16: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NAND INBO de cuatro entradas (medido en picojulio*picosegundo).

### C.1.4. consumo estático

| vector de entrada | consumo COBO | consumo INBO |
|---|---|---|
| 1 | 32.7 pW | 20.4 pW |
| 2 | 27.8 pW | 20.5 pW |
| 3 | 18.4 pW | 7.0 pW |
| 4 | 23.7 pW | 20.7 pW |
| 5 | 13.5 pW | 7.0 pW |
| 6 | 13.5 pW | 7.0 pW |
| 7 | 11.5 pW | 4.8 pW |
| 8 | 20.9 pW | 21.2 pW |
| 9 | 9.0 pW | 7.1 pW |
| 10 | 9.0 pW | 7.1 pW |
| 11 | 7.0 pW | 4.8 pW |
| 12 | 8.6 pW | 7.1 pW |
| 13 | 6.5 pW | 4.8 pW |
| 14 | 6.5 pW | 4.7 pW |
| 15 | 6.1 pW | 3.6 pW |

Tabla C.17: Consumo estático de las puertas NOR de cuatro entradas cuando el árbol serie no está conduciendo.

| vector de entrada | consumo COBO | consumo INBO |
|---|---|---|
| 0 | 9.0 pW | 6.6 pW |
| 1 | 11.9 pW | 9.4 pW |
| 2 | 11.9 pW | 9.4 pW |
| 3 | 19.2 pW | 17.5 pW |
| 4 | 11.9 pW | 9.4 pW |
| 5 | 19.2 pW | 17.5 pW |
| 6 | 19.2 pW | 17.5 pW |
| 7 | 120.7 pW | 121.2 pW |
| 8 | 17.2 pW | 9.4 pW |
| 9 | 24.1 pW | 17.4 pW |
| 10 | 25.0 pW | 17.4 pW |
| 11 | 102.7 pW | 113.9 pW |
| 12 | 29.9 pW | 17.4 pW |
| 13 | 106.0 pW | 111.6 pW |
| 14 | 110.1 pW | 110.2 pW |

Tabla C.18: Consumo estático de las puertas NAND de cuatro entradas cuando el árbol serie no está conduciendo.

## C.2. Puertas de ocho entradas

### C.2.1. retraso pin a pin

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 485 ps | 561 ps | 635 ps | 708 ps |
| entrada 1 | 473 ps | 553 ps | 625 ps | 700 ps |
| entrada 2 | 449 ps | 525 ps | 601 ps | 674 ps |
| entrada 3 | 403 ps | 478 ps | 554 ps | 627 ps |
| entrada 4 | 343 ps | 417 ps | 490 ps | 565 ps |
| entrada 5 | 267 ps | 339 ps | 410 ps | 483 ps |
| entrada 6 | 190 ps | 252 ps | 318 ps | 387 ps |
| entrada 7 | 130 ps | 174 ps | 225 ps | 283 ps |

Tabla C.19: Retraso de subida de la puerta NOR COBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 418 ps | 454 ps | 487 ps | 520 ps |
| entrada 1 | 412 ps | 447 ps | 480 ps | 512 ps |
| entrada 2 | 399 ps | 433 ps | 465 ps | 495 ps |
| entrada 3 | 376 ps | 409 ps | 440 ps | 470 ps |
| entrada 4 | 345 ps | 377 ps | 407 ps | 436 ps |
| entrada 5 | 302 ps | 334 ps | 364 ps | 393 ps |
| entrada 6 | 247 ps | 278 ps | 309 ps | 338 ps |
| entrada 7 | 174 ps | 208 ps | 240 ps | 271 ps |

Tabla C.20: Retraso de bajada de la puerta NOR COBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 343 ps | 407 ps | 470 ps | 533 ps |
| entrada 1 | 338 ps | 403 ps | 466 ps | 528 ps |
| entrada 2 | 323 ps | 389 ps | 451 ps | 514 ps |
| entrada 3 | 300 ps | 365 ps | 428 ps | 491 ps |
| entrada 4 | 266 ps | 330 ps | 393 ps | 456 ps |
| entrada 5 | 227 ps | 288 ps | 351 ps | 413 ps |
| entrada 6 | 181 ps | 239 ps | 300 ps | 361 ps |
| entrada 7 | 136 ps | 186 ps | 240 ps | 298 ps |

Tabla C.21: Retraso de subida de la puerta NOR INBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 303 ps | 336 ps | 367 ps | 397 ps |
| entrada 1 | 298 ps | 330 ps | 361 ps | 391 ps |
| entrada 2 | 291 ps | 322 ps | 352 ps | 381 ps |
| entrada 3 | 280 ps | 311 ps | 341 ps | 370 ps |
| entrada 4 | 261 ps | 292 ps | 321 ps | 350 ps |
| entrada 5 | 238 ps | 269 ps | 299 ps | 327 ps |
| entrada 6 | 208 ps | 240 ps | 270 ps | 299 ps |
| entrada 7 | 173 ps | 206 ps | 238 ps | 268 ps |

Tabla C.22: Retraso de bajada de la puerta NOR INBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 412 ps | 443 ps | 474 ps | 503 ps |
| entrada 1 | 393 ps | 424 ps | 453 ps | 483 ps |
| entrada 2 | 372 ps | 402 ps | 432 ps | 461 ps |
| entrada 3 | 350 ps | 380 ps | 410 ps | 439 ps |
| entrada 4 | 325 ps | 355 ps | 385 ps | 413 ps |
| entrada 5 | 297 ps | 327 ps | 357 ps | 386 ps |
| entrada 6 | 267 ps | 298 ps | 328 ps | 357 ps |
| entrada 7 | 238 ps | 270 ps | 301 ps | 331 ps |

Tabla C.23: Retraso de subida de la puerta NAND COBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 213 ps | 240 ps | 266 ps | 293 ps |
| entrada 1 | 211 ps | 238 ps | 264 ps | 290 ps |
| entrada 2 | 203 ps | 230 ps | 257 ps | 283 ps |
| entrada 3 | 192 ps | 219 ps | 246 ps | 273 ps |
| entrada 4 | 178 ps | 206 ps | 232 ps | 259 ps |
| entrada 5 | 163 ps | 190 ps | 217 ps | 243 ps |
| entrada 6 | 148 ps | 175 ps | 201 ps | 228 ps |
| entrada 7 | 136 ps | 162 ps | 188 ps | 214 ps |

Tabla C.24: Retraso de bajada de la puerta NAND COBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 323 ps | 354 ps | 385 ps | 414 ps |
| entrada 1 | 312 ps | 342 ps | 372 ps | 402 ps |
| entrada 2 | 297 ps | 327 ps | 357 ps | 386 ps |
| entrada 3 | 282 ps | 312 ps | 341 ps | 370 ps |
| entrada 4 | 266 ps | 296 ps | 325 ps | 354 ps |
| entrada 5 | 247 ps | 277 ps | 307 ps | 336 ps |
| entrada 6 | 231 ps | 261 ps | 291 ps | 320 ps |
| entrada 7 | 213 ps | 244 ps | 274 ps | 304 ps |

Tabla C.25: Retraso de subida de la puerta NAND INBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 176 ps | 203 ps | 229 ps | 254 ps |
| entrada 1 | 175 ps | 202 ps | 228 ps | 253 ps |
| entrada 2 | 170 ps | 196 ps | 222 ps | 248 ps |
| entrada 3 | 161 ps | 188 ps | 213 ps | 239 ps |
| entrada 4 | 152 ps | 179 ps | 205 ps | 231 ps |
| entrada 5 | 141 ps | 168 ps | 194 ps | 220 ps |
| entrada 6 | 132 ps | 159 ps | 185 ps | 211 ps |
| entrada 7 | 123 ps | 149 ps | 175 ps | 200 ps |

Tabla C.26: Retraso de bajada de la puerta NAND INBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 215 fJ | 235 fJ | 258 fJ | 281fJ |
| entrada 1 | 210 fJ | 225 fJ | 253 fJ | 269 fJ |
| entrada 2 | 182 fJ | 208 fJ | 230 fJ | 250 fJ |
| entrada 3 | 163 fJ | 190 fJ | 208 fJ | 234 fJ |
| entrada 4 | 140 fJ | 159 fJ | 187 fJ | 206 fJ |
| entrada 5 | 111 fJ | 138 fJ | 154 fJ | 189 fJ |
| entrada 6 | 94 fJ | 113 fJ | 141 fJ | 162 fJ |
| entrada 7 | 69 fJ | 93 fJ | 112 fJ | 133 fJ |

Tabla C.27: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NOR COBO de ocho entradas.

### C.2.2. consumo dinámico

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 163 fJ | 185 fJ | 208 fJ | 231 fJ |
| entrada 1 | 161 fJ | 186 fJ | 209 fJ | 234 fJ |
| entrada 2 | 152 fJ | 174 fJ | 197 fJ | 225 fJ |
| entrada 3 | 141 fJ | 166 fJ | 185 fJ | 214 fJ |
| entrada 4 | 135 fJ | 157 fJ | 181 fJ | 203 fJ |
| entrada 5 | 116 fJ | 143 fJ | 166 fJ | 188 fJ |
| entrada 6 | 97 fJ | 124 fJ | 148 fJ | 173 fJ |
| entrada 7 | 87 fJ | 113 fJ | 143 fJ | 166 fJ |

Tabla C.28: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NOR INBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 60 fJ | 67 fJ | 75 fJ | 82 fJ |
| entrada 1 | 56 fJ | 63 fJ | 71 fJ | 79 fJ |
| entrada 2 | 53 fJ | 60 fJ | 67 fJ | 74 fJ |
| entrada 3 | 49 fJ | 55 fJ | 63 fJ | 70 fJ |
| entrada 4 | 43 fJ | 52 fJ | 59 fJ | 66 fJ |
| entrada 5 | 39 fJ | 48 fJ | 55 fJ | 62 fJ |
| entrada 6 | 36 fJ | 43 fJ | 50 fJ | 59 fJ |
| entrada 7 | 31 fJ | 39 fJ | 45 fJ | 53 fJ |

Tabla C.29: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NAND COBO de ocho entradas.

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 50 fJ | 57 fJ | 64 fJ | 72 fJ |
| entrada 1 | 48 fJ | 55 fJ | 61 fJ | 70 fJ |
| entrada 2 | 44 fJ | 51 fJ | 59 fJ | 66 fJ |
| entrada 3 | 40 fJ | 47 fJ | 56 fJ | 63 fJ |
| entrada 4 | 38 fJ | 44 fJ | 53 fJ | 60 fJ |
| entrada 5 | 35 fJ | 41 fJ | 49 fJ | 57 fJ |
| entrada 6 | 31 fJ | 39 fJ | 46 fJ | 54 fJ |
| entrada 7 | 29 fJ | 36 fJ | 43 fJ | 51fJ |

Tabla C.30: Energía consumida cuando se aplica un pulso a una entrada sensibilizada de la puerta NAND INBO de ocho entradas.

### C.2.3. producto energía-retraso

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 104 | 132 | 164 | 199 |
| entrada 1 | 99 | 124 | 158 | 188 |
| entrada 2 | 82 | 109 | 138 | 169 |
| entrada 3 | 66 | 91 | 115 | 147 |
| entrada 4 | 48 | 66 | 92 | 116 |
| entrada 5 | 34 | 47 | 63 | 91 |
| entrada 6 | 23 | 32 | 45 | 63 |
| entrada 7 | 12 | 19 | 27 | 38 |

Tabla C.31: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NOR COBO de ocho entradas
(medido en picojulio*picosegundo).

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 56 | 75 | 98 | 123 |
| entrada 1 | 54 | 75 | 97 | 123 |
| entrada 2 | 49 | 67 | 89 | 116 |
| entrada 3 | 42 | 60 | 79 | 105 |
| entrada 4 | 36 | 52 | 71 | 93 |
| entrada 5 | 28 | 41 | 58 | 78 |
| entrada 6 | 20 | 30 | 44 | 62 |
| entrada 7 | 15 | 23 | 34 | 49 |

Tabla C.32: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NOR INBO de ocho entradas (medido en picojulio*picosegundo).

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 25 | 30 | 35 | 41 |
| entrada 1 | 22 | 27 | 32 | 38 |
| entrada 2 | 20 | 24 | 29 | 34 |
| entrada 3 | 17 | 21 | 26 | 31 |
| entrada 4 | 14 | 18 | 23 | 27 |
| entrada 5 | 12 | 16 | 20 | 24 |
| entrada 6 | 10 | 13 | 16 | 21 |
| entrada 7 | 7 | 11 | 13 | 18 |

Tabla C.33: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NAND COBO de ocho entradas (medido en picojulio*picosegundo).

| *Fan-out* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| entrada 0 | 16 | 20 | 25 | 30 |
| entrada 1 | 15 | 19 | 23 | 28 |
| entrada 2 | 13 | 17 | 21 | 25 |
| entrada 3 | 11 | 15 | 19 | 23 |
| entrada 4 | 10 | 13 | 17 | 21 |
| entrada 5 | 9 | 11 | 15 | 19 |
| entrada 6 | 7 | 10 | 13 | 17 |
| entrada 7 | 6 | 9 | 12 | 15 |

Tabla C.34: Producto energía-retraso cuando se aplica un pulso a cada una de las entradas de la puerta NAND INBO de ocho entradas (medido en picojulio*picosegundo).

## C.2.4. consumo estático

| | $I_7I_6I_5$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $I_4I_3I_2I_1I_0$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 00000 | 966,85 | 40,28 | 35,66 | 17,58 | 35,40 | 13,14 | 13,05 | 9,03 |
| 00001 | 63,64 | 22,83 | 17,98 | 14,01 | 13,60 | 9,30 | 9,15 | 7,50 |
| 00010 | 41,76 | 19,34 | 14,50 | 12,60 | 9,85 | 7,86 | 7,67 | 6,77 |
| 00011 | 59,00 | 22,75 | 17,90 | 13,99 | 13,51 | 9,28 | 9,13 | 7,49 |
| 00100 | 37,01 | 19,20 | 14,37 | 12,56 | 9,70 | 7,81 | 7,62 | 6,74 |
| 00101 | 33,45 | 17,84 | 13,03 | 11,85 | 8,32 | 7,12 | 6,92 | 6,37 |
| 00110 | 37,09 | 19,22 | 14,39 | 12,57 | 9,72 | 7,82 | 7,63 | 6,75 |
| 00111 | 54,34 | 22,68 | 17,82 | 13,97 | 13,42 | 9,25 | 9,10 | 7,48 |
| 01000 | 32,22 | 19,06 | 14,23 | 12,50 | 9,54 | 7,75 | 7,57 | 6,72 |
| 01001 | 28,68 | 17,65 | 12,86 | 11,79 | 8,14 | 7,05 | 6,85 | 6,34 |
| 01010 | 27,21 | 16,94 | 12,16 | 11,37 | 7,46 | 6,67 | 6,46 | 6,16 |
| 01011 | 28,65 | 17,65 | 12,85 | 11,78 | 8,13 | 7,05 | 6,85 | 6,33 |
| 01100 | 32,22 | 19,06 | 14,23 | 12,50 | 9,54 | 7,75 | 7,57 | 6,72 |
| 01101 | 28,68 | 17,65 | 12,86 | 11,79 | 8,14 | 7,05 | 6,85 | 6,34 |
| 01110 | 28,65 | 17,65 | 12,85 | 11,78 | 8,13 | 7,05 | 6,85 | 6,33 |
| 01111 | 27,21 | 16,94 | 12,16 | 11,37 | 7,46 | 6,67 | 6,46 | 6,16 |
| 10000 | 44,97 | 22,52 | 17,66 | 13,92 | 13,23 | 9,21 | 9,05 | 7,46 |
| 10001 | 27,63 | 18,97 | 14,14 | 12,48 | 9,45 | 7,73 | 7,55 | 6,71 |
| 10010 | 24,10 | 17,55 | 12,75 | 11,75 | 8,02 | 7,02 | 6,82 | 6,33 |
| 10011 | 27,55 | 18,95 | 14,13 | 12,47 | 9,43 | 7,72 | 7,54 | 6,70 |
| 10100 | 23,96 | 17,50 | 12,71 | 11,72 | 7,97 | 7,01 | 6,79 | 6,32 |
| 10101 | 22,56 | 16,78 | 12,02 | 11,32 | 7,30 | 6,63 | 6,42 | 6,15 |
| 10110 | 23,98 | 17,51 | 12,72 | 11,73 | 7,98 | 7,01 | 6,80 | 6,32 |
| 10111 | 27,48 | 18,93 | 14,10 | 12,46 | 9,40 | 7,71 | 7,52 | 6,70 |
| 11000 | 23,81 | 17,45 | 12,66 | 11,70 | 7,92 | 6,97 | 6,77 | 6,31 |
| 11001 | 22,37 | 16,71 | 11,95 | 11,28 | 7,23 | 6,60 | 6,39 | 6,13 |
| 11010 | 21,61 | 16,28 | 11,54 | 11,03 | 6,86 | 6,39 | 6,19 | 6,05 |
| 11011 | 22,35 | 16,70 | 11,94 | 11,27 | 7,22 | 6,60 | 6,39 | 6,13 |
| 11100 | 23,81 | 17,45 | 12,66 | 11,70 | 7,92 | 6,97 | 6,77 | 6,31 |
| 11101 | 22,37 | 16,71 | 11,95 | 11,28 | 7,23 | 6,60 | 6,39 | 6,13 |
| 11110 | 22,35 | 16,70 | 11,94 | 11,27 | 7,22 | 6,60 | 6,39 | 6,13 |
| 11111 | 21,61 | 16,28 | 11,54 | 11,03 | 6,86 | 6,39 | 6,19 | 6,05 |

Tabla C.35: Consumo estático de la puerta NOR COBO de 8 entradas (picowatios).

| | $I_7I_6I_5$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $I_4I_3I_2I_1I_0$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 00000 | 968,04 | 35,73 | 36,06 | 12,92 | 36,75 | 12,96 | 12,99 | 8,35 |
| 00001 | 33,91 | 11,99 | 12,02 | 7,84 | 12,05 | 7,85 | 7,87 | 6,32 |
| 00010 | 11,63 | 7,34 | 7,37 | 6,00 | 7,39 | 6,02 | 6,03 | 5,07 |
| 00011 | 34,06 | 12,04 | 12,07 | 7,86 | 12,10 | 7,88 | 7,90 | 6,33 |
| 00100 | 11,71 | 7,39 | 7,41 | 6,03 | 7,44 | 6,05 | 6,06 | 5,09 |
| 00101 | 7,11 | 5,68 | 5,70 | 4,78 | 5,71 | 4,78 | 4,79 | 4,17 |
| 00110 | 11,66 | 7,37 | 7,39 | 6,01 | 7,41 | 6,03 | 6,05 | 5,08 |
| 00111 | 34,20 | 12,08 | 12,11 | 7,88 | 12,14 | 7,90 | 7,92 | 6,35 |
| 01000 | 11,77 | 7,43 | 7,45 | 6,05 | 7,47 | 6,08 | 6,08 | 5,12 |
| 01001 | 7,14 | 5,70 | 5,72 | 4,80 | 5,74 | 4,80 | 4,81 | 4,18 |
| 01010 | 5,39 | 4,55 | 4,56 | 3,95 | 4,57 | 3,95 | 3,97 | 3,61 |
| 01011 | 7,17 | 5,73 | 5,74 | 4,82 | 5,75 | 4,82 | 4,83 | 4,20 |
| 01100 | 11,77 | 7,43 | 7,45 | 6,05 | 7,47 | 6,08 | 6,08 | 5,12 |
| 01101 | 7,14 | 5,70 | 5,72 | 4,80 | 5,74 | 4,80 | 4,81 | 4,18 |
| 01110 | 7,17 | 5,73 | 5,74 | 4,82 | 5,75 | 4,82 | 4,83 | 4,20 |
| 01111 | 5,39 | 4,55 | 4,56 | 3,95 | 4,57 | 3,95 | 3,97 | 3,61 |
| 10000 | 35,56 | 12,89 | 12,93 | 8,32 | 12,97 | 8,33 | 8,35 | 6,59 |
| 10001 | 11,98 | 7,82 | 7,84 | 6,29 | 7,86 | 6,30 | 6,32 | 5,32 |
| 10010 | 7,34 | 5,99 | 6,00 | 5,05 | 6,02 | 5,06 | 5,07 | 4,37 |
| 10011 | 12,02 | 7,85 | 7,87 | 6,30 | 7,88 | 6,33 | 6,34 | 5,33 |
| 10100 | 7,39 | 6,01 | 6,03 | 5,08 | 6,06 | 5,08 | 5,09 | 4,39 |
| 10101 | 5,68 | 4,77 | 4,78 | 4,16 | 4,79 | 4,16 | 4,17 | 3,78 |
| 10110 | 7,36 | 6,00 | 6,01 | 5,06 | 6,04 | 5,07 | 5,08 | 4,38 |
| 10111 | 12,06 | 7,87 | 7,89 | 6,31 | 7,91 | 6,33 | 6,35 | 5,35 |
| 11000 | 7,43 | 6,04 | 6,06 | 5,10 | 6,07 | 5,11 | 5,12 | 4,41 |
| 11001 | 5,70 | 4,79 | 4,80 | 4,17 | 4,81 | 4,18 | 4,19 | 3,79 |
| 11010 | 4,55 | 3,94 | 3,95 | 3,60 | 3,96 | 3,61 | 3,62 | 3,31 |
| 11011 | 5,72 | 4,81 | 4,83 | 4,19 | 4,82 | 4,19 | 4,20 | 3,80 |
| 11100 | 7,43 | 6,04 | 6,06 | 5,10 | 6,07 | 5,11 | 5,12 | 4,41 |
| 11101 | 5,70 | 4,79 | 4,80 | 4,17 | 4,81 | 4,18 | 4,19 | 3,79 |
| 11110 | 5,72 | 4,81 | 4,83 | 4,19 | 4,82 | 4,19 | 4,20 | 3,80 |
| 11111 | 4,55 | 3,94 | 3,95 | 3,60 | 3,96 | 3,61 | 3,62 | 3,31 |

Tabla C.36: Consumo estático de la puerta NOR INBO de 8 entradas (picowatios).

| | $I_7I_6I_5$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $I_4I_3I_2I_1I_0$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 00000 | 12,43 | 12,53 | 12,71 | 12,94 | 18,03 | 18,31 | 23,69 | 29,31 |
| 00001 | 12,48 | 12,61 | 12,74 | 13,07 | 18,06 | 18,42 | 23,79 | 29,62 |
| 00010 | 12,56 | 12,66 | 12,84 | 13,35 | 18,19 | 18,72 | 24,07 | 30,41 |
| 00011 | 12,46 | 12,59 | 12,74 | 13,02 | 18,06 | 18,39 | 23,79 | 29,60 |
| 00100 | 12,59 | 12,66 | 12,87 | 13,38 | 18,21 | 18,75 | 24,07 | 30,49 |
| 00101 | 12,64 | 12,87 | 13,07 | 14,09 | 18,44 | 19,49 | 24,89 | 32,97 |
| 00110 | 12,56 | 12,69 | 12,87 | 13,35 | 18,19 | 18,72 | 24,04 | 30,44 |
| 00111 | 12,46 | 12,56 | 12,74 | 13,07 | 18,03 | 18,42 | 23,81 | 29,65 |
| 01000 | 12,53 | 12,69 | 12,87 | 13,40 | 18,21 | 18,78 | 24,12 | 30,85 |
| 01001 | 12,64 | 12,87 | 13,10 | 14,15 | 18,47 | 19,54 | 24,94 | 33,25 |
| 01010 | 12,84 | 13,63 | 13,86 | 16,65 | 19,29 | 22,02 | 27,45 | 56,71 |
| 01011 | 12,61 | 12,87 | 13,10 | 14,15 | 18,49 | 19,54 | 24,97 | 33,36 |
| 01100 | 12,53 | 12,69 | 12,87 | 13,40 | 18,21 | 18,78 | 24,12 | 30,85 |
| 01101 | 12,64 | 12,87 | 13,10 | 14,15 | 18,47 | 19,54 | 24,94 | 33,25 |
| 01110 | 12,61 | 12,87 | 13,10 | 14,15 | 18,49 | 19,54 | 24,97 | 33,36 |
| 01111 | 12,84 | 13,63 | 13,86 | 16,65 | 19,29 | 22,02 | 27,45 | 56,71 |
| 10000 | 12,51 | 12,61 | 12,82 | 13,30 | 18,14 | 18,67 | 24,04 | 35,10 |
| 10001 | 12,59 | 12,74 | 12,92 | 13,58 | 18,24 | 18,98 | 24,33 | 35,89 |
| 10010 | 12,64 | 12,92 | 13,12 | 14,32 | 18,49 | 19,75 | 25,09 | 38,42 |
| 10011 | 12,59 | 12,69 | 12,89 | 13,53 | 18,24 | 18,95 | 24,30 | 35,89 |
| 10100 | 12,64 | 12,92 | 13,15 | 14,38 | 18,52 | 19,77 | 25,17 | 38,70 |
| 10101 | 12,87 | 13,66 | 13,89 | 16,81 | 19,29 | 22,15 | 27,60 | 61,88 |
| 10110 | 12,64 | 12,92 | 13,15 | 14,30 | 18,49 | 19,72 | 25,12 | 38,57 |
| 10111 | 12,53 | 12,69 | 12,89 | 13,61 | 18,24 | 18,98 | 24,33 | 36,09 |
| 11000 | 12,69 | 12,97 | 13,20 | 14,73 | 18,57 | 20,08 | 25,50 | 49,22 |
| 11001 | 12,87 | 13,74 | 13,94 | 17,06 | 19,36 | 22,38 | 27,86 | 72,26 |
| 11010 | 13,58 | 16,32 | 16,47 | 47,60 | 21,84 | 46,66 | 51,57 | 158,93 |
| 11011 | 12,87 | 13,74 | 13,97 | 17,24 | 19,39 | 22,54 | 28,01 | 77,48 |
| 11100 | 12,69 | 12,97 | 13,20 | 14,73 | 18,57 | 20,08 | 25,50 | 49,22 |
| 11101 | 12,87 | 13,74 | 13,94 | 17,06 | 19,36 | 22,38 | 27,86 | 72,26 |
| 11110 | 12,87 | 13,74 | 13,97 | 17,24 | 19,39 | 22,54 | 28,01 | 77,48 |
| 11111 | 13,58 | 16,32 | 16,47 | 47,60 | 21,84 | 46,66 | 51,57 | 158,93 |

Tabla C.37: Consumo estático de la puerta NAND COBO de 8 entradas (picowatios).

| | $I_7I_6I_5$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $I_4I_3I_2I_1I_0$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 00000 | 1,68 | 1,93 | 1,93 | 2,28 | 1,93 | 2,25 | 2,25 | 2,65 |
| 00001 | 1,92 | 2,30 | 2,28 | 2,76 | 2,29 | 2,76 | 2,72 | 3,42 |
| 00010 | 2,29 | 2,79 | 2,77 | 3,52 | 2,74 | 3,50 | 3,47 | 4,83 |
| 00011 | 1,92 | 2,29 | 2,32 | 2,78 | 2,29 | 2,76 | 2,73 | 3,43 |
| 00100 | 2,29 | 2,76 | 2,77 | 3,50 | 2,75 | 3,48 | 3,46 | 4,83 |
| 00101 | 2,76 | 3,49 | 3,48 | 4,90 | 3,49 | 4,88 | 4,83 | 7,36 |
| 00110 | 2,27 | 2,76 | 2,78 | 3,50 | 2,77 | 3,50 | 3,49 | 4,83 |
| 00111 | 1,93 | 2,28 | 2,27 | 2,76 | 2,26 | 2,73 | 2,73 | 3,41 |
| 01000 | 2,30 | 2,75 | 2,75 | 3,50 | 2,76 | 3,49 | 3,48 | 4,84 |
| 01001 | 2,77 | 3,50 | 3,49 | 4,91 | 3,48 | 4,87 | 4,85 | 7,35 |
| 01010 | 3,49 | 4,90 | 4,90 | 7,43 | 4,85 | 7,39 | 7,35 | 38,08 |
| 01011 | 2,79 | 3,49 | 3,49 | 4,90 | 3,46 | 4,87 | 4,84 | 7,37 |
| 01100 | 2,30 | 2,75 | 2,75 | 3,50 | 2,76 | 3,49 | 3,48 | 4,84 |
| 01101 | 2,77 | 3,50 | 3,49 | 4,91 | 3,48 | 4,87 | 4,85 | 7,35 |
| 01110 | 2,79 | 3,49 | 3,49 | 4,90 | 3,46 | 4,87 | 4,84 | 7,37 |
| 01111 | 3,49 | 4,90 | 4,90 | 7,43 | 4,85 | 7,39 | 7,35 | 38,08 |
| 10000 | 1,93 | 2,26 | 2,26 | 2,67 | 2,23 | 2,65 | 2,67 | 3,27 |
| 10001 | 2,29 | 2,77 | 2,73 | 3,41 | 2,74 | 3,42 | 3,39 | 4,66 |
| 10010 | 2,78 | 3,48 | 3,49 | 4,88 | 3,48 | 4,83 | 4,80 | 7,15 |
| 10011 | 2,29 | 2,77 | 2,73 | 3,41 | 2,73 | 3,41 | 3,37 | 4,65 |
| 10100 | 2,76 | 3,49 | 3,48 | 4,88 | 3,47 | 4,84 | 4,81 | 7,15 |
| 10101 | 3,48 | 4,90 | 4,88 | 7,41 | 4,86 | 7,37 | 7,33 | 37,57 |
| 10110 | 2,75 | 3,49 | 3,47 | 4,87 | 3,48 | 4,84 | 4,83 | 7,14 |
| 10111 | 2,28 | 2,75 | 2,76 | 3,42 | 2,74 | 3,41 | 3,40 | 4,64 |
| 11000 | 2,76 | 3,49 | 3,48 | 4,88 | 3,46 | 4,85 | 4,82 | 7,19 |
| 11001 | 3,49 | 4,90 | 4,88 | 7,41 | 4,86 | 7,38 | 7,35 | 37,23 |
| 11010 | 4,88 | 7,42 | 7,40 | 41,02 | 7,41 | 39,07 | 38,41 | 151,98 |
| 11011 | 3,49 | 4,90 | 4,89 | 7,42 | 4,85 | 7,38 | 7,34 | 37,10 |
| 11100 | 2,76 | 3,49 | 3,48 | 4,88 | 3,46 | 4,85 | 4,82 | 7,19 |
| 11101 | 3,49 | 4,90 | 4,88 | 7,41 | 4,86 | 7,38 | 7,35 | 37,23 |
| 11110 | 3,49 | 4,90 | 4,89 | 7,42 | 4,85 | 7,38 | 7,34 | 37,10 |
| 11111 | 4,88 | 7,42 | 7,40 | 41,02 | 7,41 | 39,07 | 38,41 | 151,98 |

Tabla C.38: Consumo estático de la puerta NAND INBO de 8 entradas (picowatios).

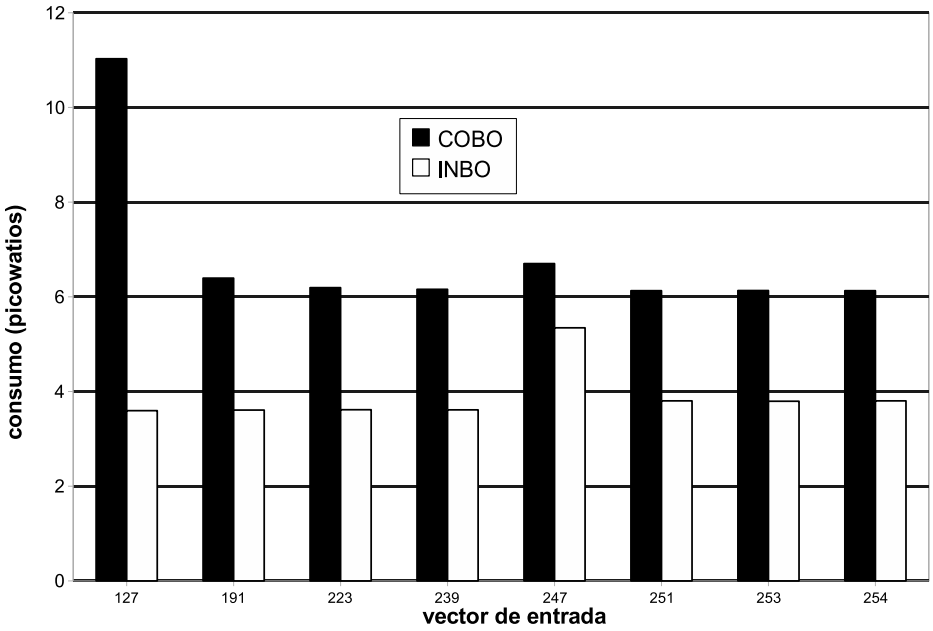



Figura C.1: Consumo estático de las puertas NOR de ocho entradas cuando hay un solo transistor serie activo.

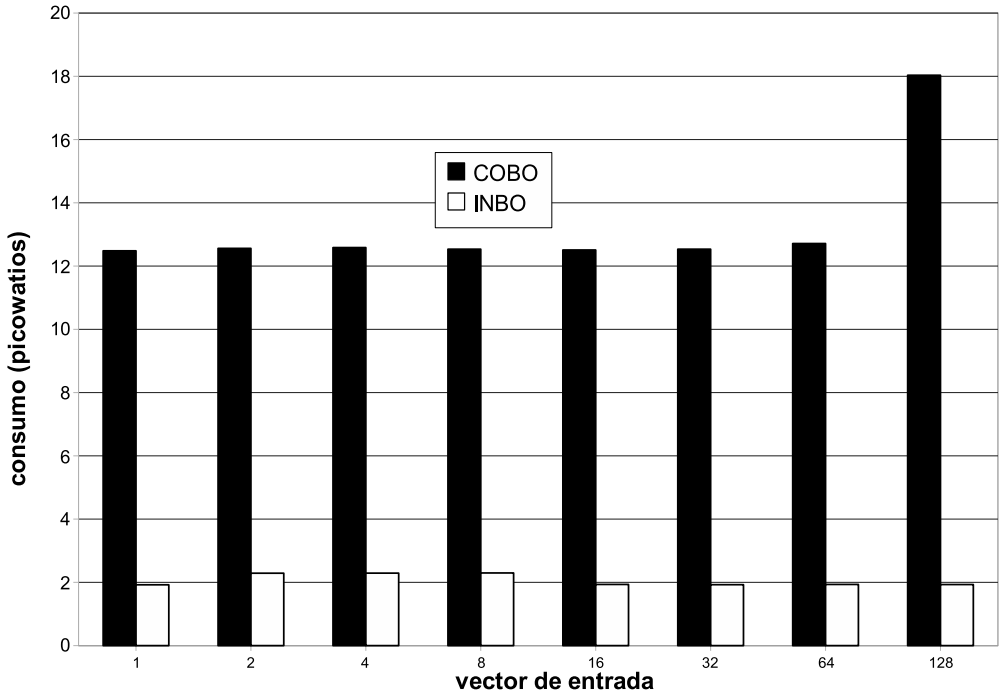



Figura C.2: Consumo estático de las puertas NAND de ocho entradas cuando hay un solo transistor serie activo.

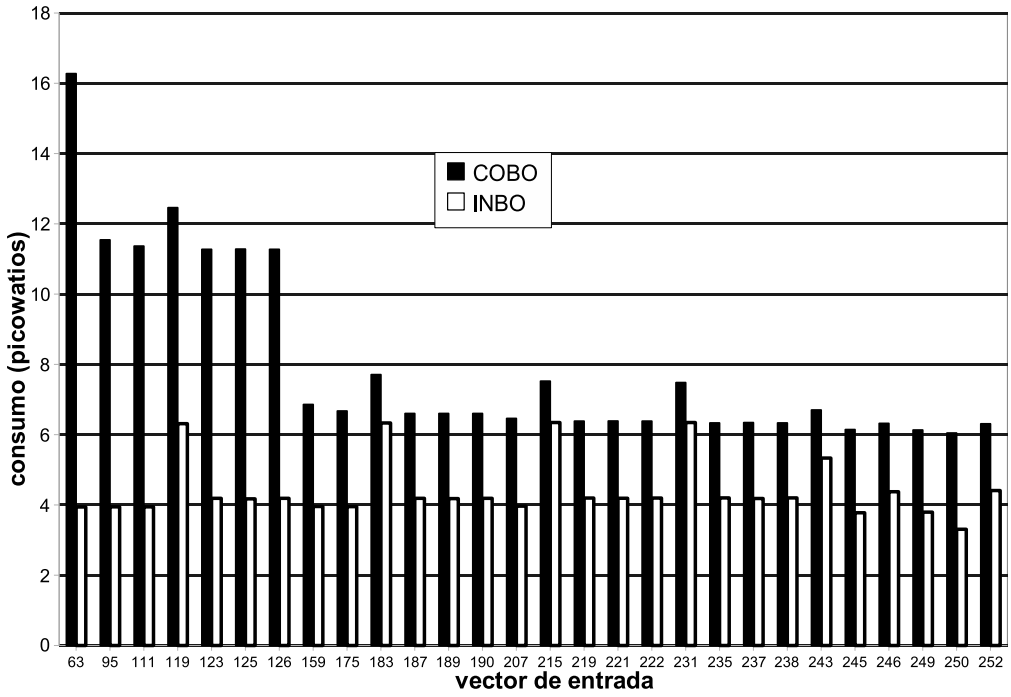



Figura C.3: Consumo estático de las puertas NOR de ocho entradas cuando hay dos transistores serie activos.

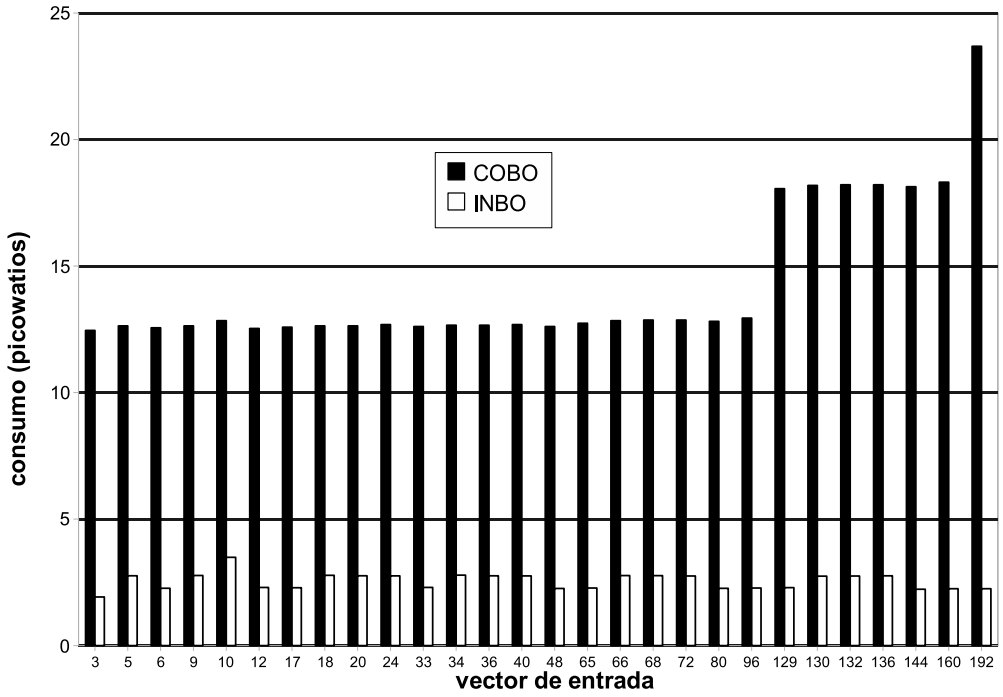



Figura C.4: Consumo estático de las puertas NAND de ocho entradas cuando hay dos transistores serie activos.

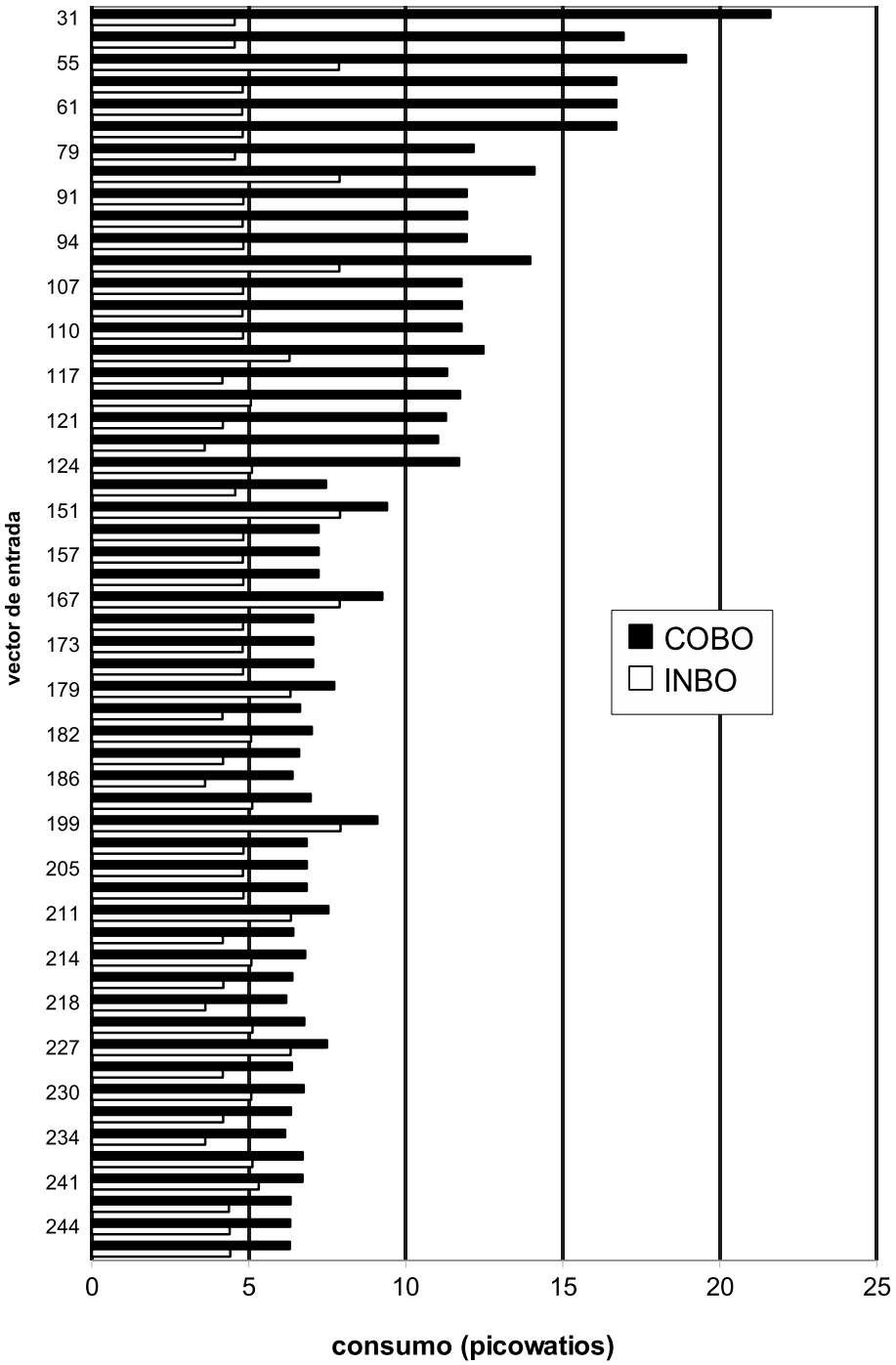



Figura C.5: Consumo estático de las puertas NOR de ocho entradas cuando hay tres transistores serie activos.

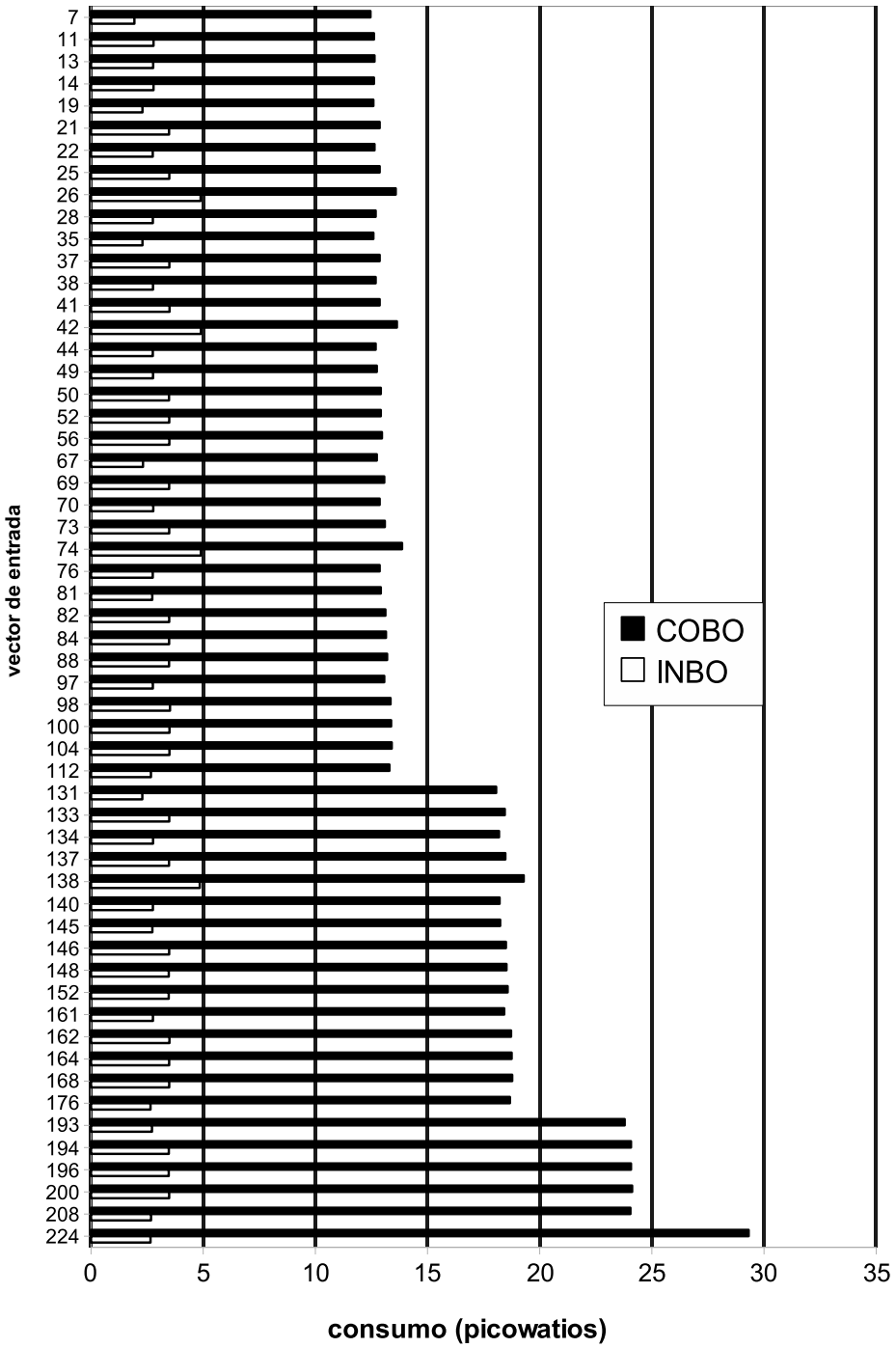



Figura C.6: Consumo estático de las puertas NAND de ocho entradas cuando hay tres transistores serie activos.

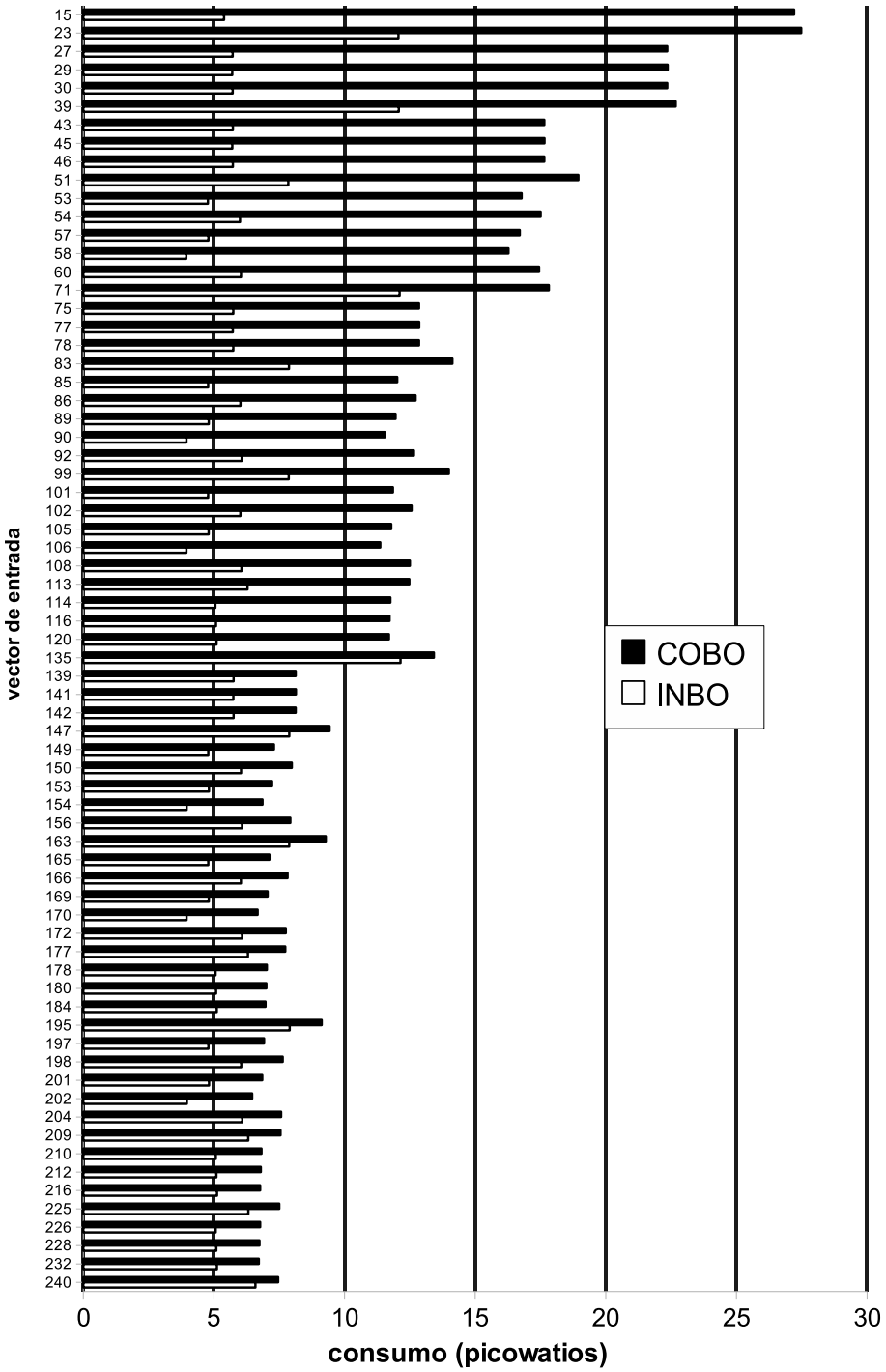



Figura C.7: Consumo estático de las puertas NOR de ocho entradas cuando hay cuatro transistores serie activos.

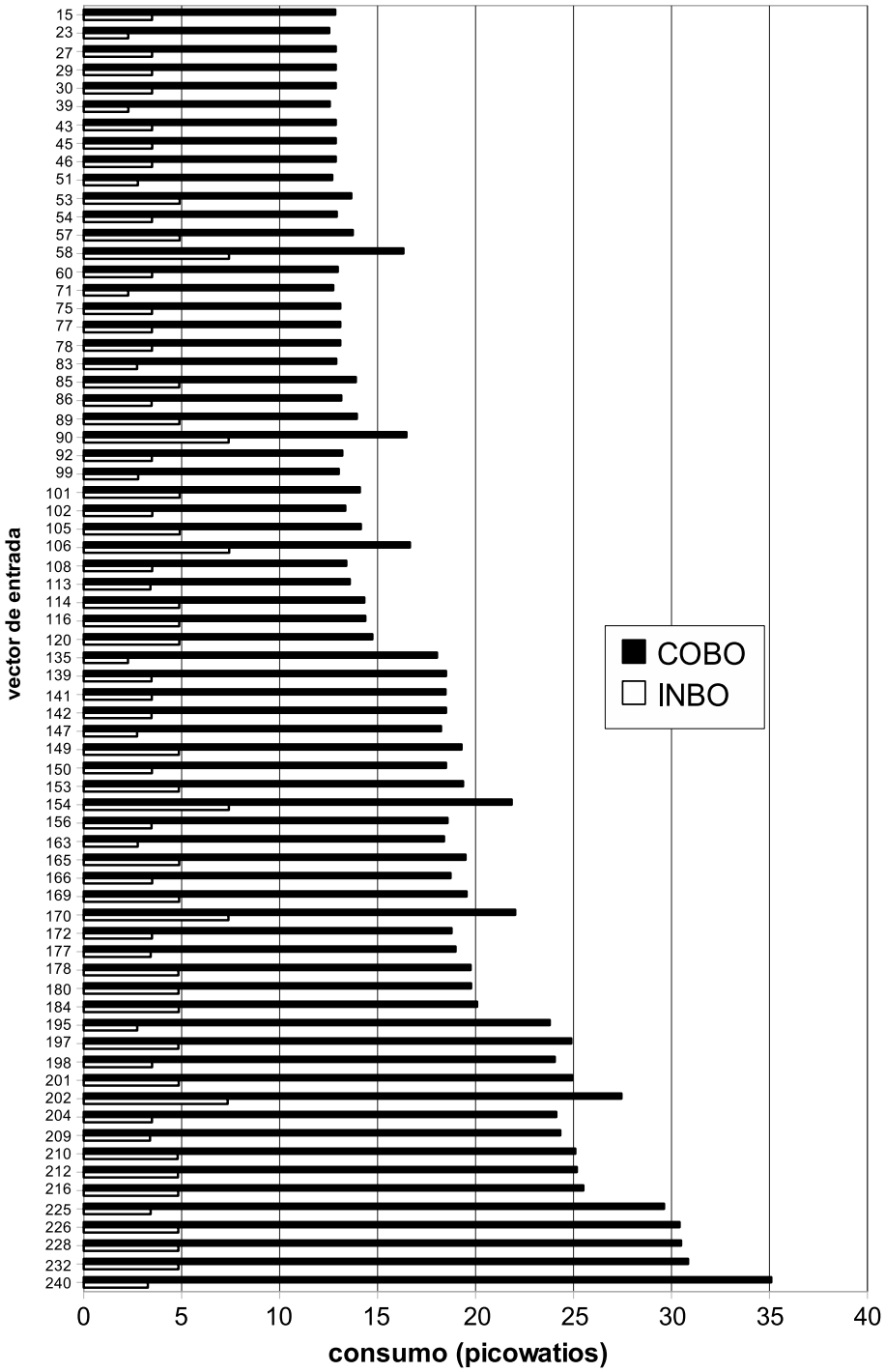



Figura C.8: Consumo estático de las puertas NAND de ocho entradas cuando hay cuatro transistores serie activos.

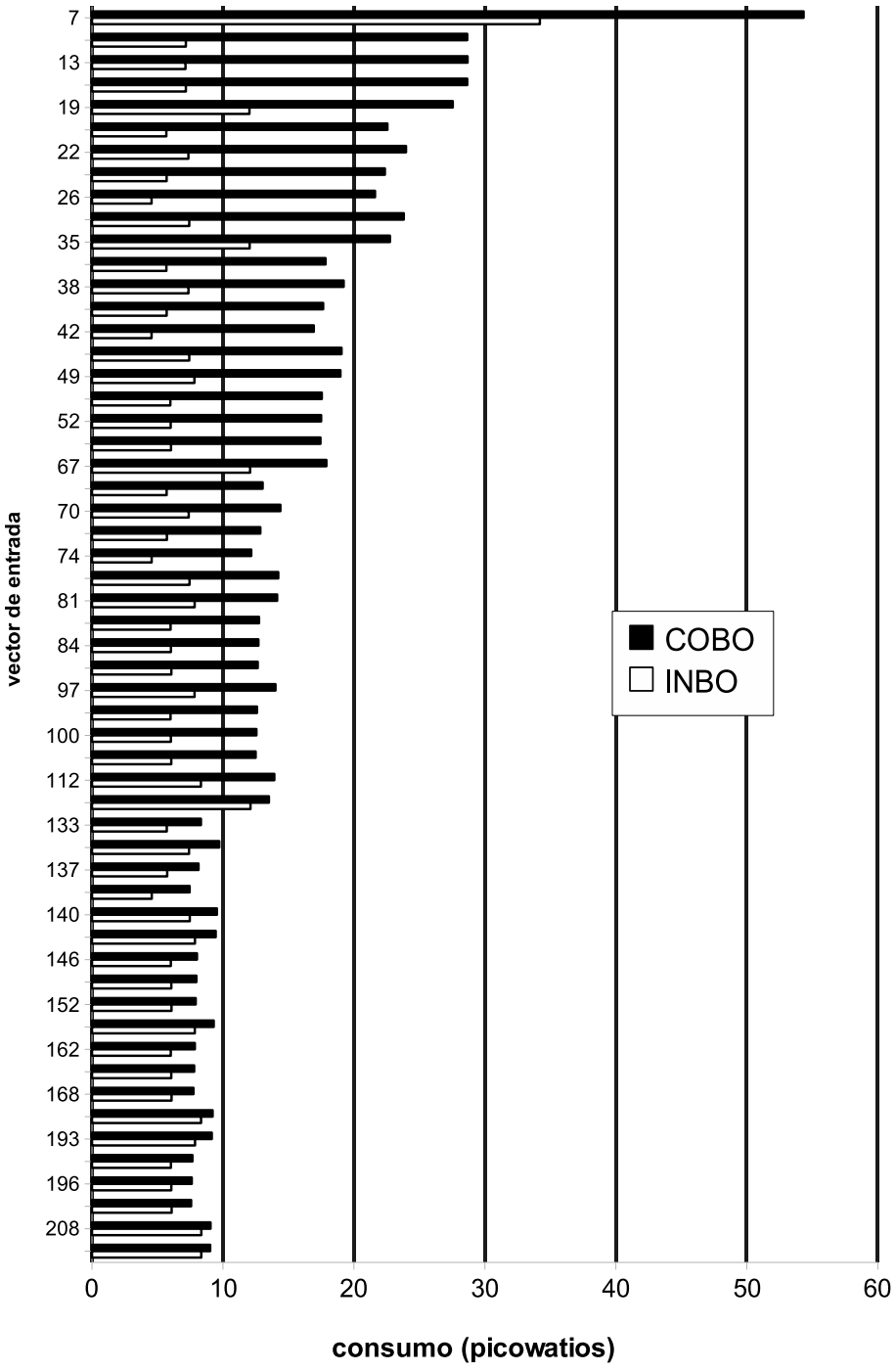



Figura C.9: Consumo estático de las puertas NOR de ocho entradas cuando hay cinco transistores serie activos.

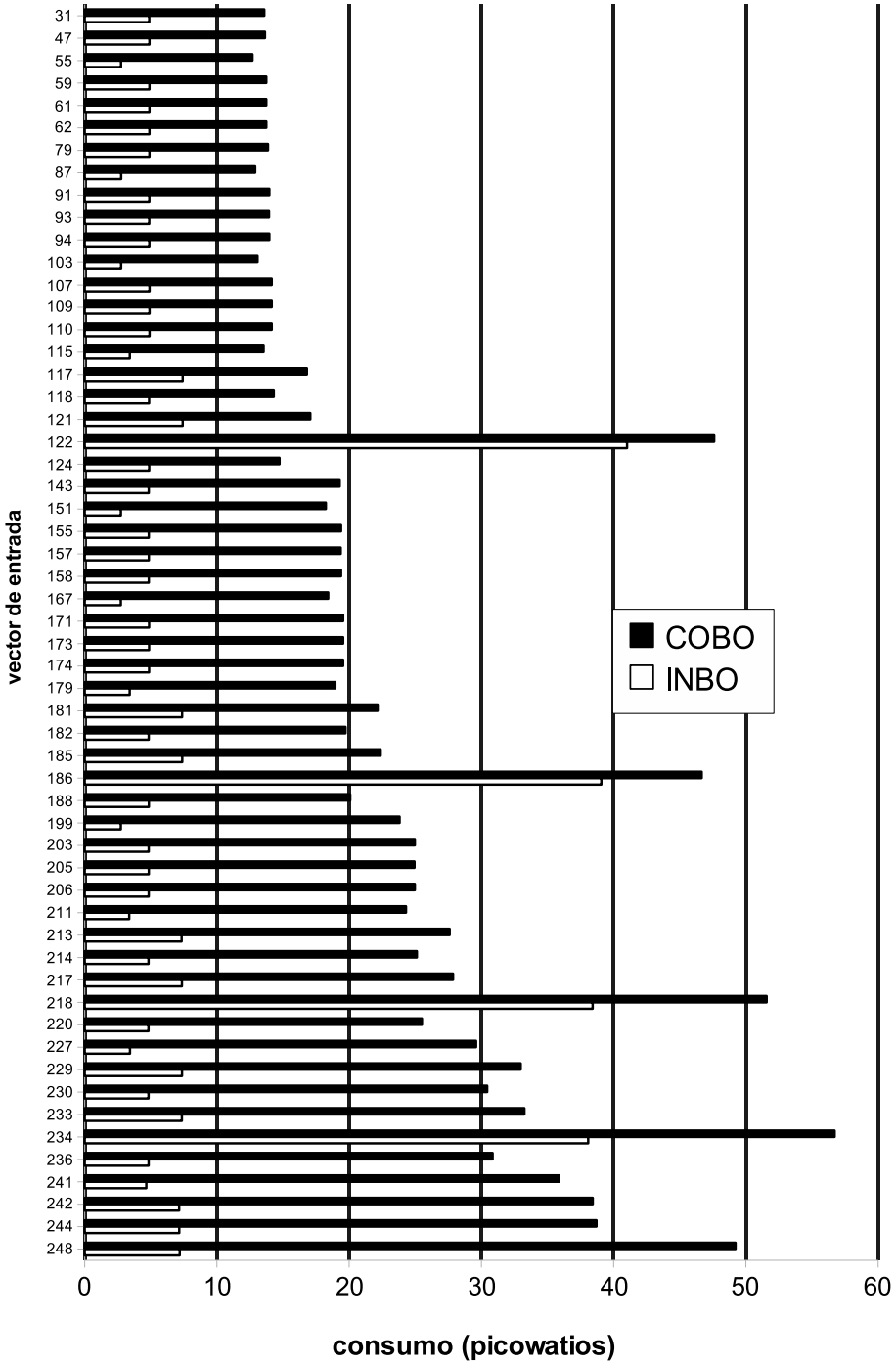



Figura C.10: Consumo estático de las puertas NAND de ocho entradas cuando hay cinco transistores serie activos.

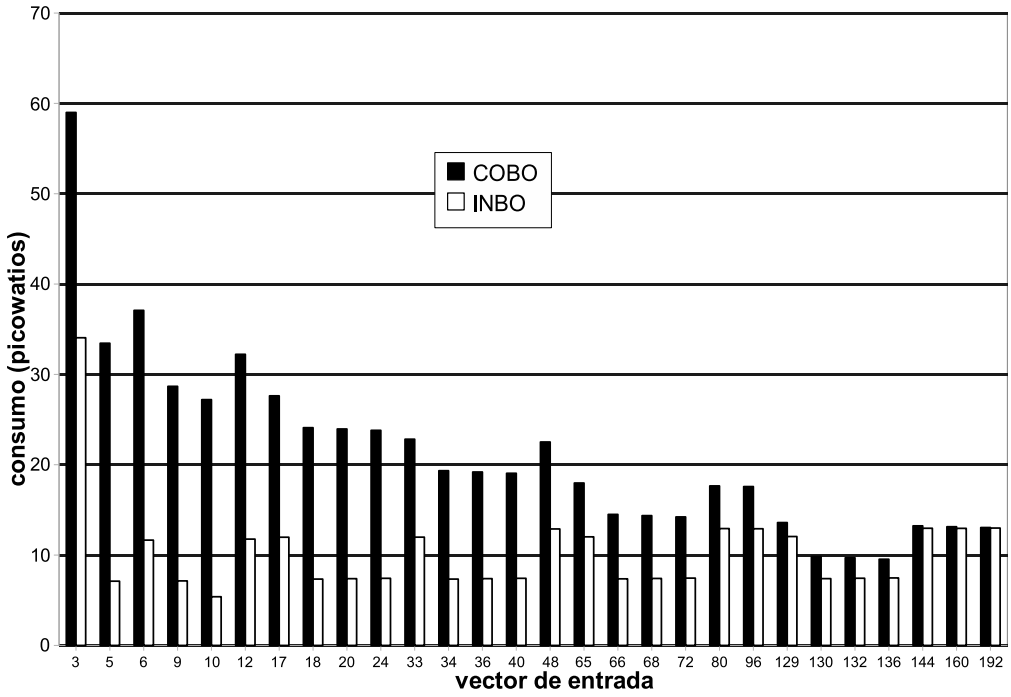



Figura C.11: Consumo estático de las puertas NOR de ocho entradas cuando hay seis transistores serie activos.

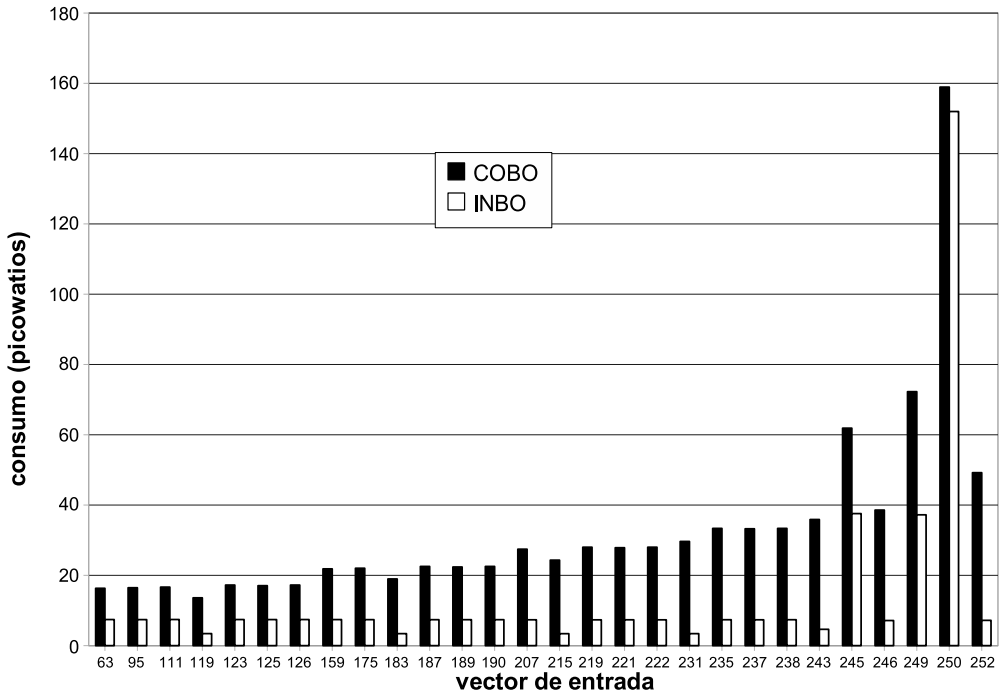



Figura C.12: Consumo estático de las puertas NAND de ocho entradas cuando hay seis transistores serie activos.

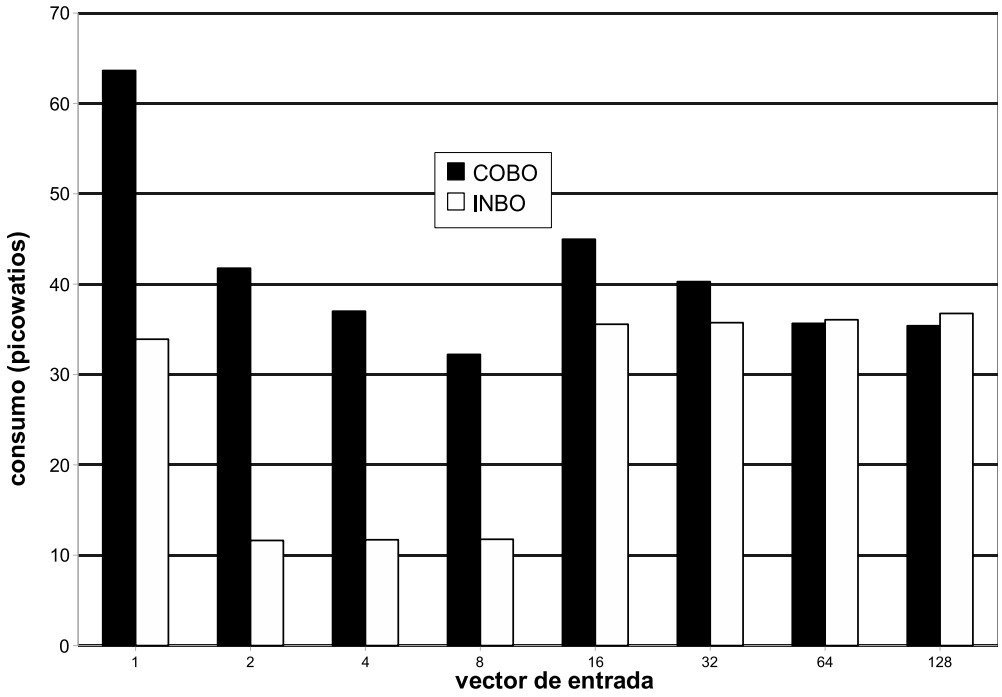



Figura C.13: Consumo estático de las puertas NOR de ocho entradas cuando hay siete transistores serie activos.

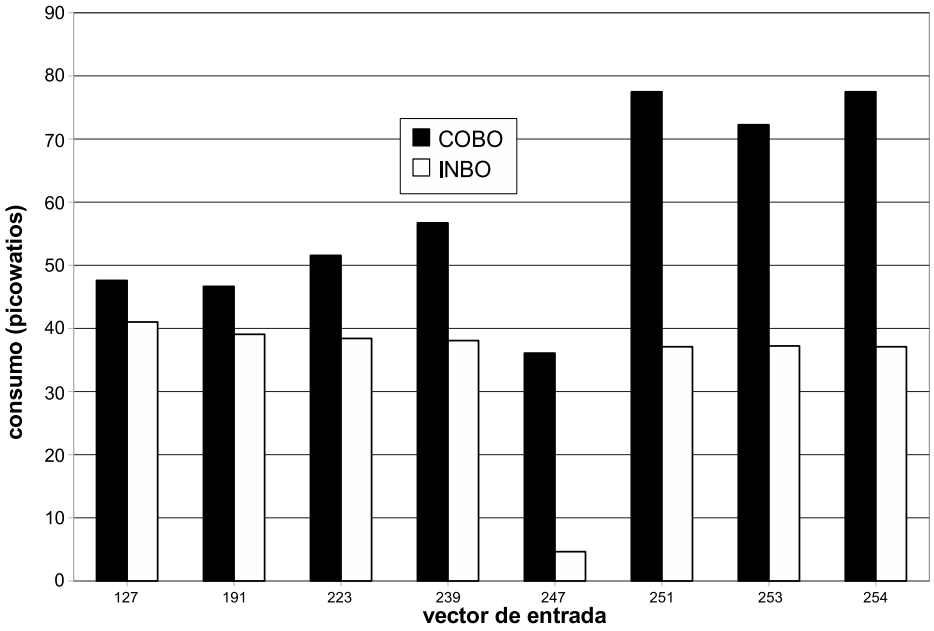



Figura C.14: Consumo estático de las puertas NAND de ocho entradas cuando hay siete transistores serie activos.

# Apéndice D

# Reconocimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia en Innovación a través del proyecto TEC2007-61802/MIC (HIPER) y por el Fondo Europeo de Desarrollo Regional (FEDER).

# Índice de figuras

# Índice de tablas

# Bibliografía


[1] "Diccionario de la lengua española".

[2] "Wikipedia, the free encyclopedia".

[3] D. Harris, *Skew-Tolerant Circuit Design*, Morgan Kaufmann Publisher, 2001.

[4] R. Chau, S. Datta, M. Doczy, B. Doyle, J. Kavalieros, and M. Metz, "High-k/metal-gate stack and its mosfet characteristics", *Electron Device Letters, IEEE*, vol. 25, no. 6, pp. 408–410, June 2004.

[5] H. Shimada, I. Ohshima, T. Ushiki, S. Sugawa, and T. Ohmi, "Tantalum nitride metal gate fd-soi cmos fets using low resistivity self-grown bcc-tantalum layer", *IEEE Transactions on Electron Devices*, vol. 48, no. 8, pp. 1619–1626, August 2001.

[6] Rino Choi, K. Onishi, Chang Seok Kang, S. Gopalan, Renee Nieh, Y.H. Kim, J.H. Han, S. Krishnan, Hag ju Cho, A. Shahriar, and J.C. Lee, "Fabrication of high quality ultra-thin hfo2 gate dielectric mosfets using deuterium anneal", in *Electron Devices Meeting, 2002. IEDM '02. Digest. International*, 2002, pp. 613–616.

[7] L. Manchanda, M.L. Green, R.B. van Dover, M.D. Morris, A. Kerber, Y. Hu, J.-P. Han, P.J. Silverman, T.W. Sorsch, G. Weber, V. Donnelly, K. Pelhos, F. Klemens, N.A. Ciampa, A. Kornblit, Y.O. Kim, J.E. Bower, D. Barr, E. Ferry, D. Jacobson, J. Eng, B. Busch, and H. Schulte, "Si-doped aluminates for high temperature metal-gate cmos: Zr-al-si-o, a novel gate dielectric for low power applications", in *Electron Devices*

*Meeting, 2000. IEDM Technical Digest. International*, December 2000, pp. 23–26.

[8] Andrew Marshall and Sreedhar Natarajan, *SOI Design: Analog, Memory and Digital Techniques*, Kluwer Academic Publishers, 2001.

[9] Jan M. Rabaey, *Digital Integrated Circuits*, Prentice Hall, December 2002.

[10] R. Zimmermann and W. Fichtner, "Low-power logic styles: Cmos versus pass-transistor logic", *IEEE Journal of Solid-State Circuits*, vol. 32, no. 7, pp. 1079–1090, July 1997.

[11] N. Mohankumar, B. Syamal, and C. K. Sarkar, "Performance and optimisation of dual material gate short channel bulk mosfets for analogue/mixed signal applications", *International Journal of Electronics*, vol. 96, no. 6, pp. 603–611, June 2009.

[12] M. J. Bellido, *BIESTABLES CMOS VLSI BAJO ENTRADAS ASÍNCRONAS: PROBLEMAS Y APLICACIONES*, PhD thesis, Facultad de Física, 1994.

[13] John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publisher, 2 edition, 1996.

[14] M. Alshaikh, D. Kinniment, and A. Yakovlev, "On the trade-off between resolution time and delay times in bistable circuits", in *IEEE International Conference on Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th*, December 2009, pp. 355–358.

[15] Adel S. Sedra and Kenneth C. Smith, *Microelectronic Circuits*, Oxford University Press, 6 edition, 2010.

[16] Charles Hawkins and Jaume Segura, *Introduction to Modern Digital Electronics*, SciTech Publishing, 911 Paverstone Drive, Suite B Raleigh, NC 27615, March 2010.

[17] T. Kao J., M. Miyazaki, and A. P. Chandrakasan, "A 175-mv multiply-accumulate unit using an adaptative supply voltage and body bias architecture", *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1545–1554, November 2002.

[18] D. Helms, E. Schmidt, and W. Nebel, "Leakage in cmos circuits – an introduction", in *Proceedings of 14th workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, September 2004, vol. 3254, pp. 17–35.

[19] Ji-Yong J., K. Gil-Su, S. Jong-Pil, R Woo-Jin, and K. Soo-Won, "Body bias generator for leakage power reduction of low-voltage digital logic circuits", in *Proceedings of 16th workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. October 2006, Lecture Notes in Computer Science, pp. 350–559, Springer.

[20] M. Sumita, Sakiyama S., Kinoshita M., Araki Y., Y. Ikeda, and Fukuoka K., "Mixed body-bias techniques with fixed vt and ids generation circuits", in *International Conference on Integrated Circuit Design and Technology (ICICDT) 2005*, May 2005, pp. 233–234.

[21] Y. Tsividis, *Operation and Modeling of the MOS Transistor*, McGraw-Hill, 1987.

[22] Carlos Galup-Montoro and Márcio Cherem Schneider, *MOSFET modeling for circuit analysis and design*, International Series on Advances in Solid State Electronics and Technology. World Scientific Publishing, 2007.

[23] S. Mukhopadhyay, H. Mahmoodi-Meimand, C. Neau, and K. Roy, "Leakage in nanometer scale cmos circuits", in *International Symposium on VLSI Technology, Systems, and Applications, 2003*, 2003, pp. 307–312.

[24] P. Simonen, A. Heinonen, M. Kuulusa, and J. Nurmi, "Comparison of bulk and soi cmos technologies in a dsp processor circuit

implementation", in *Proceedings of the 13th International Conference on Microelectronics*, October 2001, pp. 107–110.

[25] C. Mazure, "Advanced substrate engineering for the nanotechnology era", in *International Symposium on VLSI Technology Systems, and Applications 2006*, April 2006.

[26] James C. Pickel and James T. Blandford, "Cosmic-ray-induced errors in mos devices", *IEEE Transactions on Nuclear Science*, vol. 27, no. 2, pp. 1006–1015, April 1980.

[27] L. Adams, "Cosmic ray effects in microelectronics", *Microelectronics Journal*, vol. 16, no. 2, pp. 17–29, March 1985.

[28] A. J. Acosta, A. Barriga, M. J. Bellido, J. Juan, and M. Valencia, *Temporización en circuitos integrados digitales CMOS*, Marcombo S. A., 2000.

[29] W.J. Dally and J.W. Poulton, *Digital Systems Engineering*, 1998.

[30] K. Bernstein, K.M. Carring, C.M. Durham, P.R. Hansen, D. Hogenmiller, E.J Nowak, and N.J. Rohrer, *High Speed CMOS Design Styles*, 1998.

[31] H.B Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, 1990.

[32] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A System Perspective*, Addison-Wesley Pubs., 2 edition, 1993.

[33] W.C. Elmore, "The transitient response of damped linear networks with particular regard to wide-band amplifiers", *Journal of Applied Physics*, vol. 19, no. 1, pp. 55–63, January 1948.

[34] J. Rubestein, P. Penfield, and M. A. Horowitz, "Signal delay in rc tree networks", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and SystemsIEEE Transactions on*, vol. 2, no. 3, pp. 202–211, July 1983.

[35] S.H. Unger and Chung-Jen Tan, "Clocking schemes for high-speed digital systems", *IEEE Transactions on Computers*, vol. C-35, no. 10, pp. 880–895, October 1986.

[36] M. Afghahi and J Yuan, "Double edge-triggered d-flip-flops for high-speed cmos circuits", *IEEE Journal of Solid-State Circuits*, vol. 26, no. 8, pp. 1168–1170, August 1991.

[37] M. Horowitz, "Clocking strategies in high performance processors", in *Symposium on VLSI circuits, Digest of Technical Pagers*, June 1992, pp. 50–53.

[38] *Clock Skew and Short Paths Timing*, Application Note AC198. Actel Corporation, March 2004.

[39] Kenneth D. Wagner, "Clock system design", *EEE Design & Test of Computers*, vol. 5, no. 5, pp. 9–27, September 1988.

[40] V. G. Oklobdzija, "Clocking and clocked storage elements in multi-ghz environment", in *Proceedings of the 12th International Workshop on Power and Timing Modeling, Optimization and Simulation*, September 2002, pp. 128–145.

[41] M. Pedram, Qing Wu, and Xunwei Wu, "A new design of double edge triggered flip-flops", in *Design Automation Conference 1998. Proceedings of the ASP-DAC '98. Asia and South Pacific*, February 1998, pp. 417–421.

[42] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel, and F. Baez, "Reducing power in high-performance microprocessors", in *Proceedings of the 35th Design Automation Conference*, 1998, pp. 732–737.

[43] Gustavo E. Téllez, Amir Farrahi, and Majid Sarrafzadeh, "Activity-driven clock design for low power circuits", in *Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design*, November 1995, pp. 62–65.

[44] E.L. Hill and M.H. Lipasti, "Transparent mode flip-flops for collapsible pipelines", in *25th International Conference on Computer Design (ICCD) 2007.*, October 2007, pp. 553–560.

[45] Wujin Mao, *Design and Test Generation for Clock Skew Faults of Clock-Delayed Domino Logic Circuits*, PhD thesis, University of Cincinnati ETDs, November 2006.

[46] Peter J. Ashenden and Jim Lewis, *The Student's Guide to VHDL*, Morgan Kaufmann Publishers, 2008.

[47] D. Guerrero, M. J. Bellido, J. Juan, A. Millan, P. Ruiz de Clavijo, and E. Ostúa, "Four phase alternating latches clocking scheme for cmos sequential circuits", in *Proceedings of the XIX Conference on Design of Circuits and Integrated Systems*, November 2004, pp. 780–783.

[48] Cadence Design Systems, "Affirma spectre circuit simulator user guide", 1999.

[49] Harry I. A. Chen, Edward K. W. Loo, James B. Kuo, and Marek J. Syrzycki, "Triple-threshold static power minimization in high-level synthesis of vlsi cmos", in *Proceedings of 17th workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, September 2007, vol. 4644/2007 of *Lecture Notes in Computer Science*, pp. 453–462.