

Tema I.

**Introducción a la
Microelectrónica**

1.1. Introducción

Los circuitos electrónicos tienen una gran influencia en la vida cotidiana. De hecho, rara vez es el instante de tiempo en el que nos encontremos lejos de algunos de estos circuitos, al hablar por teléfono, al despertarnos, al comer, etc. La apariencia de estos circuitos suele ser una pequeña caja negra con una serie de patas, lo que vulgarmente se conoce como cucaracha o circuito integrado. Por lo tanto, la interacción con dichos circuitos no suele ser directa, sino a través de una serie de interfaces con las cuales podemos solicitar la información deseada. Pero antes de seguir vamos a definir la parte de la ciencia que se encarga de su estudio: la MICROELECTRÓNICA.

La MICROELECTRÓNICA se puede definir como el conjunto de ciencias y técnicas con las que se realizan y fabrican circuitos electrónicos sobre una pastilla de un semiconductor, lo cual formará un circuito integrado (CI).

Dentro de estos circuitos integrados, podemos encontrar diferentes estrategias de diseño, como pueden ser los circuitos integrados en los que se deben construir tanto las puertas como las conexiones, denominados habitualmente ASICs (*Application Specific Integrated Circuits*), circuitos programables en los que se encuentran ya contruidos todas las puertas y conexiones de tal forma que únicamente hay que indicar cuales están habilitadas mediante una programación, denominados dispositivos programables (un ejemplo de los cuales pueden ser las FPGAs, *Field Programmable Gate Arrays*).

La división existente en los circuitos electrónicos también es válida para la Microelectrónica, es decir, podemos diferenciar entre Microelectrónica Analógica y Digital, según la naturaleza de las señales tratadas. No obstante, el auge de la Microelectrónica surgió gracias a una propiedad de los sistemas digitales: la jerarquía.

La propiedad de jerarquía es aquella por la cual un sistema puede estar compuesto de bloques conectados entre sí, de tal forma que dichos bloques son independientes entre sí y de su conexión.

A pesar de que un circuito analógico también puede ser construido por una serie de bloques conectados entre sí; estos bloques no serán independientes de su conexión (e incluso su situación en la base del semiconductor) debido a la naturaleza real de las señales analógicas (pueden tener un rango infinito de valores). En cambio, el carácter digital de las señales de los sistemas digitales provee dicha independencia y así la posibilidad de desarrollar un sistema completamente jerárquico. A modo de ejemplo, podemos ver en la figura 1.1 dos ASICs: el caso (a) sería un diseño analógico; y el caso (b) sería un diseño digital. En el caso analógico podemos ver unas estructuras formadas por los mismos bloques dispuestas de una manera especial (parecida a una circunferencia). Bien, esta disposición no es aleatoria sino que debe ser así para garantizar el mismo comportamiento en todos los bloques. En cambio, en el circuito digital podemos apreciar bloques comunes, cuya disposición obedece únicamente a contribuir a que el sistema completo sea lo más compacto posible.

En la figura 1.2 podemos apreciar una clasificación de los sistemas digitales desde un punto de vista jerárquico. Según dicho esquema los sistemas digitales se pueden representar desde bloques de muy alto nivel, como pueden ser procesadores y/o memorias, hasta un cir-

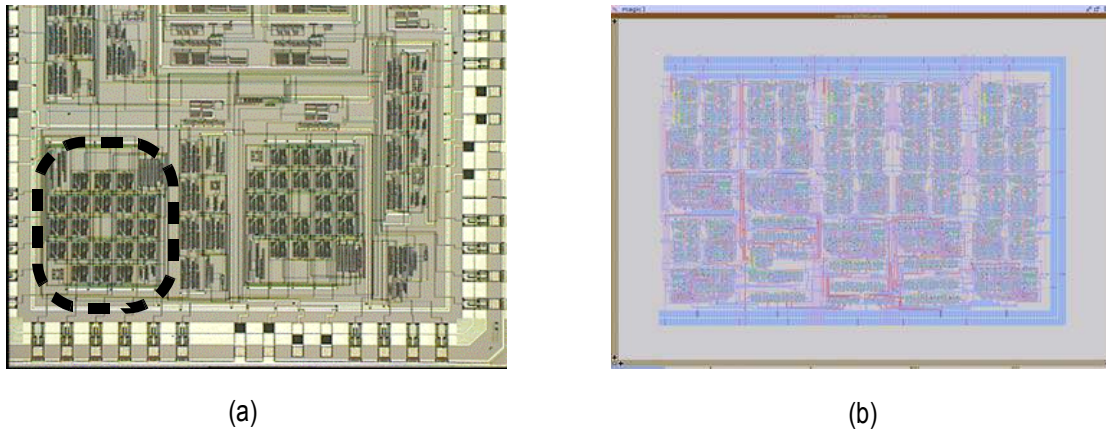


Figura 1.1.- Ejemplos de diseños microelectrónicos (a) analógicos y (b) digitales.

cuito integrado, el chip. Normalmente en el campo digital se suele utilizar un proceso de diseño top-down, en el cual partimos de las especificaciones de más alto nivel, y llegamos al circuito integrado pasando por todos los niveles de jerarquía. Dentro de este proceso podemos distinguir dos fases bien diferenciadas.

En la primera fase se obtiene el circuito lógico a partir de descripciones de alto nivel. El resultado de esta fase será un diagrama o una descripción de en un HDL (Lenguaje de Descripción de Hardware) con las conexiones de puertas, registros, etc, es decir, bloques abstractos en función del nivel en el que nos encontremos. Esta fase ya ha sido tratada a lo largo de diferentes asignaturas por lo que no será tratada.

En la segunda fase, se construirán los bloques mencionados anteriormente según una familia lógica, de tal forma que están completamente determinados el tamaño, la forma y la estructura interna de cada uno de los bloques. En esta fase, el diseñador debe obtener los diferentes esquemas a nivel de transistores (moviéndonos en el nivel eléctrico); y a partir del cual se obtendrá el layout del circuito (moviéndonos en el nivel geométrico).

El layout se puede definir como una representación física de un circuito electrónico que está sujeto a limitaciones derivadas del proceso de integración, el flujo de diseño y requerimientos de prestaciones.

Una vez que tenemos un layout correcto, se pasa a la *foundry* (que es la fábrica encargada de obtener el circuito físico) para que lleve a cabo todos los procesos de integración. La corrección del layout se debe llevar a cabo en un doble sentido:

- una verificación funcional, en la cual ha de verificarse que las capas incluidas en el layout, así como su distribución, sean equivalentes al esquema eléctrico de partida. dicha verificación se suele conocer como LVS (*Layout Versus Schematic*)
- una verificación geométrica, en la cual ha de verificarse que las capas de un determinado material están lo suficientemente separadas como para que, al finalizar el proceso de integración, dichas capas estén realmente separadas. Dicha verificación se suele conocer como DRC (*Design Rules Checking*). Esta verificación es lo suficientemente importante como para que la *foundry* vuelva a realizarla para evitar fallos.

Por lo tanto nos vamos a centrar en los niveles eléctricos y geométricos, por ser aquellos que no se han visto y en los que la acción del diseñador es importante.

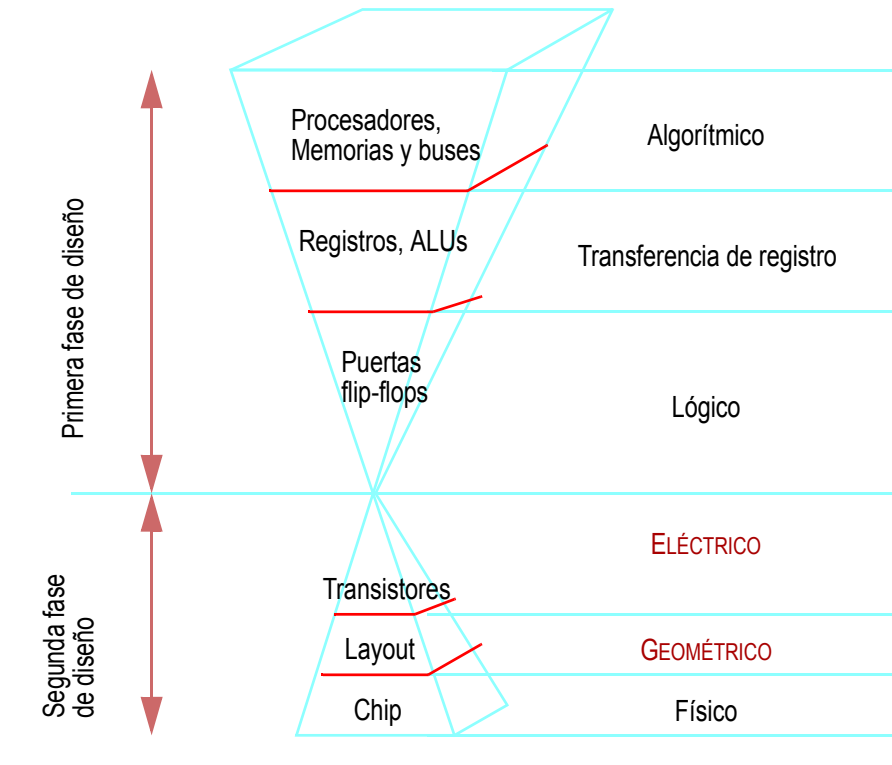


Figura 1.2.- Clasificación jerárquica de los sistemas digitales.

1.1.1. Historia

Antes de continuar, vamos a realizar un breve recorrido por la historia que nos trajo la Microelectrónica.

Todo ello empezó con la necesidad de realizar cálculos matemáticos intensivos. Esta necesidad no es reciente. De hecho en 1834, Babbage desarrolló una máquina calculadora (con las operaciones básicas de suma, resta, multiplicación y división) utilizando dispositivos mecánicos; a pesar de lo cual su filosofía no se diferencia mucho de los ordenadores de hoy día, es decir, realizaba las operaciones en dos ciclos: almacenar y ejecutar, y tenía una estructura en *pipeline* para acelerar las operaciones. No obstante, dicha máquina fue impracticable para la época debido al coste excesivamente elevado.

Con la llegada de las soluciones eléctricas, el coste se abarató. Como muestras de esta época podemos indicar la utilización de los relés y de los tubos de vacío. Uno de los campos en los que la utilización de los relés fue mayor es el campo ferroviario, de hecho, aun en nuestros días se suelen utilizar estos dispositivos. Se puede decir que la era digital comenzó con los tubos de vacío, ya que con ellos se desarrollaron las primeras computadoras: ENIAC y UNIVAC I (la primera de ellas con fines militares y la segunda fue la primera computadora comercial). No obstante, esta tecnología alcanzó pronto su límite debido al elevado consumo de potencia que tenían.

Seguidamente entraron en escena los semiconductores con los transistores bipolares y MOSFET. A pesar de que los transistores MOSFET (1925) fueron desarrollados antes que los

transistores bipolares (1947), al principio sufrieron una serie de problemas por los cuales su aplicación al procesado digital fue frenado. Así que continuamos con los transistores bipolares, con los cuales se empezaron a integrar los primeros circuitos integrados en la familia TTL (Transistor-Transistor-Logic). De nuevo se llegó al techo con el consumo de potencia, a pesar de ello sigue siendo muy utilizado en dispositivos SSI e incluso MSI. Con la solución de los problemas mencionados anteriormente a partir de 1970, los transistores MOSFET entraron en la escena del procesado de señal digital. Esta entrada culminó con la fabricación del primer microprocesador en 1972, el 4004.

La tendencia desde entonces concuerda con la denominada *ley de Moore* (cofundador de Intel)

El número de transistores que pueden ser integrados en un mismo circuito crecerá exponencialmente con el tiempo.

Dicha ley se muestra mejor gráficamente. En la figura 1.3 podemos apreciar el número de transistores de determinados sistemas (en concreto procesadores) en función del año en el que fueron desarrollados. Como podemos apreciar, dicha nube de puntos se ajusta a una línea recta cuando el número de transistores se encuentra en un escala logarítmica. Lo más sorprendente de esta ley es que fue enunciada en los años 60, y sigue vigente hasta nuestros días. De hecho, esta ley es una de las bases para las planificaciones de las industrias de semiconductores.

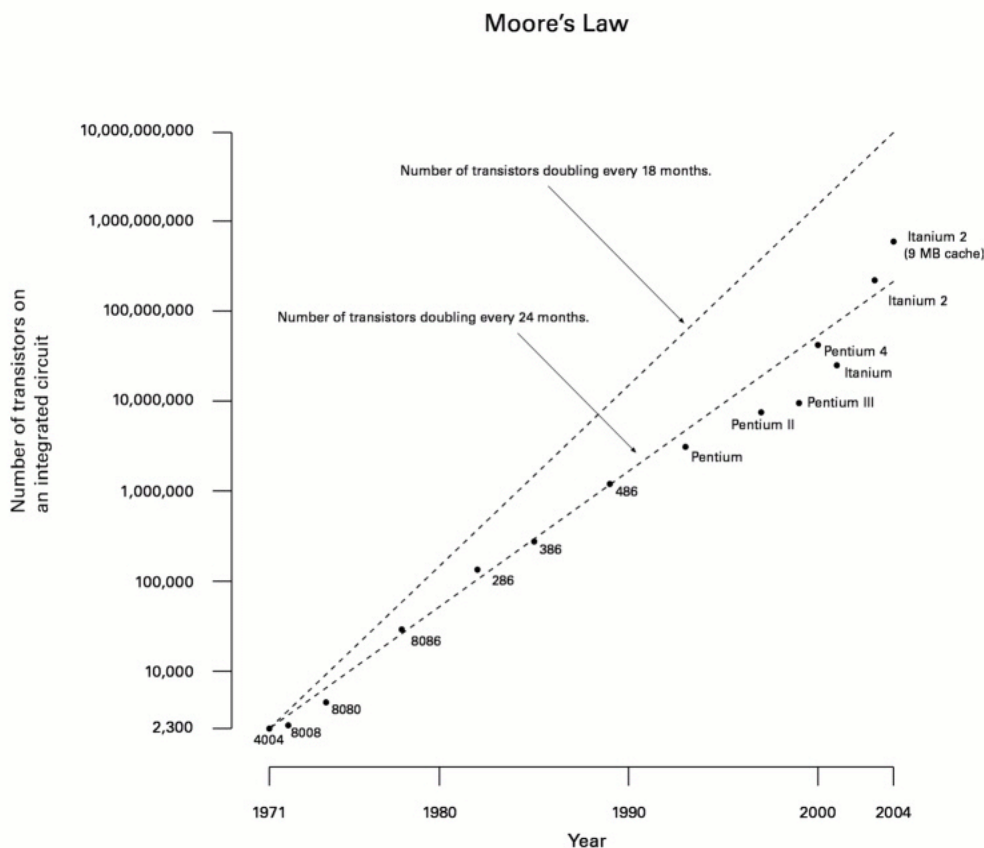


Figura 1.3.- Tendencia del número de transistores en función de los años.

1.2. Procesos de diseño

La fabricación de un circuito integrado consiste de una serie de pasos en un orden específico. El material base de los circuitos integrados es una oblea o disco de semiconductor. La tecnología más desarrollada es la basada en silicio (Si), aunque también existen tecnologías a base de otros semiconductores como arseniuro de galio (GaAs) o germanio y silicio (SiGe). No obstante, todas las tecnologías siguen unos pasos similares. El semiconductor no está en equilibrio eléctricamente, sino que unas cantidades controladas de impurezas son añadidas para dotar al cristal de las propiedades eléctricas requeridas. Dichas impurezas pueden ser donadores (cargados eléctricamente con signo negativo), que se corresponden al tipo n; o aceptoras (cargadas eléctricamente con signo positivo), que corresponden al tipo p. En función del tipo de estas impurezas podemos encontrarnos con diferentes tipos de procesos:

- Procesos de n-well (utilizado principalmente en Europa) en el que el dopado de la oblea es de tipo p.
- Procesos de p-well (utilizado principalmente en Estados Unidos) en el que el dopado de la oblea es de tipo n.
- Procesos de well gemelos (el cual se está extendiendo cada vez más) en el que el signo del dopado de la oblea no tiene demasiada importancia ya que se van a generar los dos tipos de well (n-well y p-well).

Como los tres tipos de procesos son similares, nos vamos a centrar en uno de ellos, en particular en el proceso de n-well. Dicho proceso se muestra en la figura 1.4, más concretamente la oblea sería el paso (a).

La primera máscara define el n-well (figura 1.4b). Esta zona, el n-well, es el lugar en el que se van a implementar los transistores PMOS. En esta etapa también se depositan capas de óxido grueso (denominado de campo), SiO_2 , para separar cada una de las regiones (figura 1.4c).

La siguiente capa que se deposita es la llamada óxido de puerta (o fino), la cual se obtendrá por el crecimiento de las zonas de óxido de campo (figura 1.4d). Dicha capa constituirá con el polisilicio el terminal de puerta de los transistores.

Una vez que se ha depositado el óxido de puerta, se coloca el polisilicio (figura 1.4e) y se despeja el resto de óxido de puerta para permitir la creación de los terminales de fuente y drenador (figura 1.4f).

El siguiente paso es difundir los terminales de fuente y drenador de los transistores NMOS (figura 1.4g), y seguidamente los terminales correspondientes a los transistores PMOS (figura 1.4h). También se crean los contactos con las zonas de polarización: regiones n^+ sobre el n-well, y regiones p^+ sobre el resto del CI.

Una vez creadas las diferentes regiones n^+ y p^+ , se pasa a la inserción de la primera capa de metal (figura 1.4i). El contacto físico de este material (por lo general aluminio, aunque se está probando con cobre en los últimos tiempos) con el polisilicio y las zonas de difusión provocará un contacto eléctrico. No obstante hay que indicar que aquellas zonas en las que no se requiere contacto, habían sido ocupadas con óxido de campo evitando de esta forma el contacto.

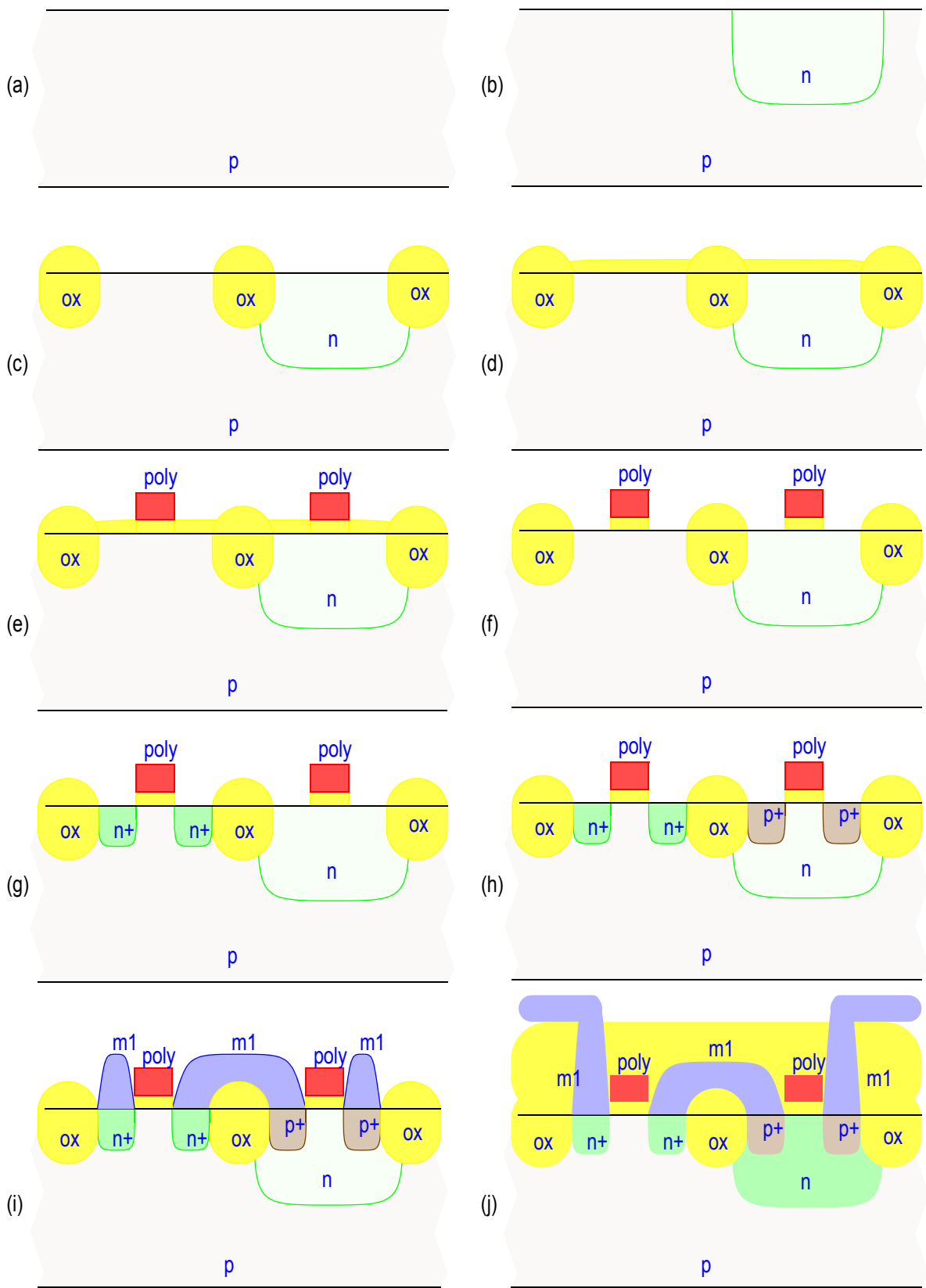


Figura 1.4.- Pasos del proceso de fabricación de un circuito CMOS con el proceso de n-well.

Para la segunda (y restantes capas) capa de metal (figura 1.4j) se coloca una nueva capa de óxido de campo por todo el CI excepto en los lugares donde exista un contacto entre la capa actual y la anterior. Después se deposita la capa de metal en los lugares correspondientes. Cabe destacar que el material de todas las capas de metal existentes, en la tecnología utilizada, es el mismo, es decir, aluminio (o cobre).

En todos estos pasos podemos distinguir dos tipos de acciones: crear zonas de difusión y de well, que alterará la composición interna de la oblea; y la deposición de material sobre la oblea. La primera acción se puede conseguir a través de dos procesos diferentes: difusión e implantación iónica.

- El proceso de difusión consiste en depositar sobre la oblea un material desde el cual obtener las impurezas deseadas y calentarla oblea a una temperatura elevada. De esta forma, los espacios intersticiales del semiconductor aumentan, y así las impurezas pueden ocupar estos espacios.
- El proceso de implantación iónica consiste en bombardear la oblea con las impurezas que se quieren difundir. Dicha difusión sólo se producirá en las zonas que no se encuentren protegidas por una máscara de material.

En cuanto a la segunda acción, el proceso más utilizado es la fotolitografía. Dicha técnica consiste en depositar por todo el circuito una capa de material en cuestión (polisilicio o metal, también se utiliza con el óxido para separar las diferentes capas, pero dicho proceso es transparente para el diseñador, la única acción del diseñador en esta capa son los contactos en los que no debe haber dicha capa de óxido), y encima de ella una máscara fotorresistiva, la cual evitará la pérdida del material que se encuentre bajo ella. Después de haber eliminado el material sobrante del circuito, se elimina la máscara dejando el circuito preparado para una nueva capa.

Por lo tanto, para cualquiera de los procesos anteriores, es necesario conocer y verificar una serie de características geométricas como son el tamaño del material depositado sobre el circuito. Dichas características son denominadas reglas de diseño, y gracias a ellas se asegura que los dispositivos descritos en el layout estarán en el circuito físico. Si algunas de estas reglas son violadas no se asegura la correcta creación del circuito electrónico.

A modo de ejemplo veamos una porción de un fichero de reglas de diseño para una tecnología CMOS estándar de $0.12\ \mu\text{m}$ en la figura 1.5. En él podemos ver las diferentes capas con las reglas geométricas correspondientes a cada una de ellas. Es interesante notar que las dimensiones están referenciadas a un parámetro, λ , para que la portabilidad de dichas reglas a otras tecnologías sea más sencilla. En dicho fichero, que lo utiliza la herramienta Microwind, podemos distinguir tres partes en cada línea: la referencia de la regla, la dimensión mínima que se ha de verificar (en términos de λ) y un breve comentario sobre dicha regla. También se suele adjuntar un esquema en el que se muestran de forma gráfica las diferentes reglas. En este caso particular únicamente se muestran las reglas del nwell, áreas activas o difusiones y del polisilicio.

1.2.1. Ejemplo de un layout simple

Como ejemplo de la creación de un layout vamos a abordar el layout de una celda de una memoria RAM dinámica, mostrada en la figura 1.6. En dicho esquema se ha utilizado las


```

*
NAME CMOS 0.12µm - 6 Metal
*
lambda = 0.06 (Lambda is set to half the
gate size)
*
* Design rules associated to each layer
*
* Well
*
r101 = 10 (well width)
r102 = 11 (well spacing)
*
* Diffusion
*
r201 = 4 (diffusion width)
r202 = 4 (diffusion spacing)
r203 = 6 (border of nwell on diffp)
r204 = 6 (nwell to next diffn)
r205 = 0 (diffn to diffp)
*
* Poly
*
r301 = 2 (poly width)
r302 = 2 (gate length)
r303 = 4 (high voltage gate length)
r304 = 3 (poly spacing)
r305 = 1 (spacing poly and unrelated diff)
r306 = 4 (width of drain and source diff)
r307 = 3 (extra gate poly)
*
* Poly 2
*
r311 = 2 (poly2 width)
r312 = 2 (poly2 spacing)
*
* Contact
r401 = 2 (contact width)
r402 = 4 (contact spacing)
r403 = 1 (metal border for contact)
r404 = 1 (poly border for contact)
r405 = 1 (diff border for contact)
r406 = 2 (contact to gate)
r407 = 1 (poly2 border for contact)
*
* metal
r501 = 3 (metal width)
r502 = 4 (metal spacing)
r510 = 16 (minimum surface)
*
* via
r601 = 2 (Via width)
r602 = 4 (Spacing)
r604 = 1 (border of metal)
r605 = 1 (border of metal2)
*
* metal 2
r701 = 3 (Metal 2 width)
r702 = 4
r710 = 16 (minimum surface)
    
```

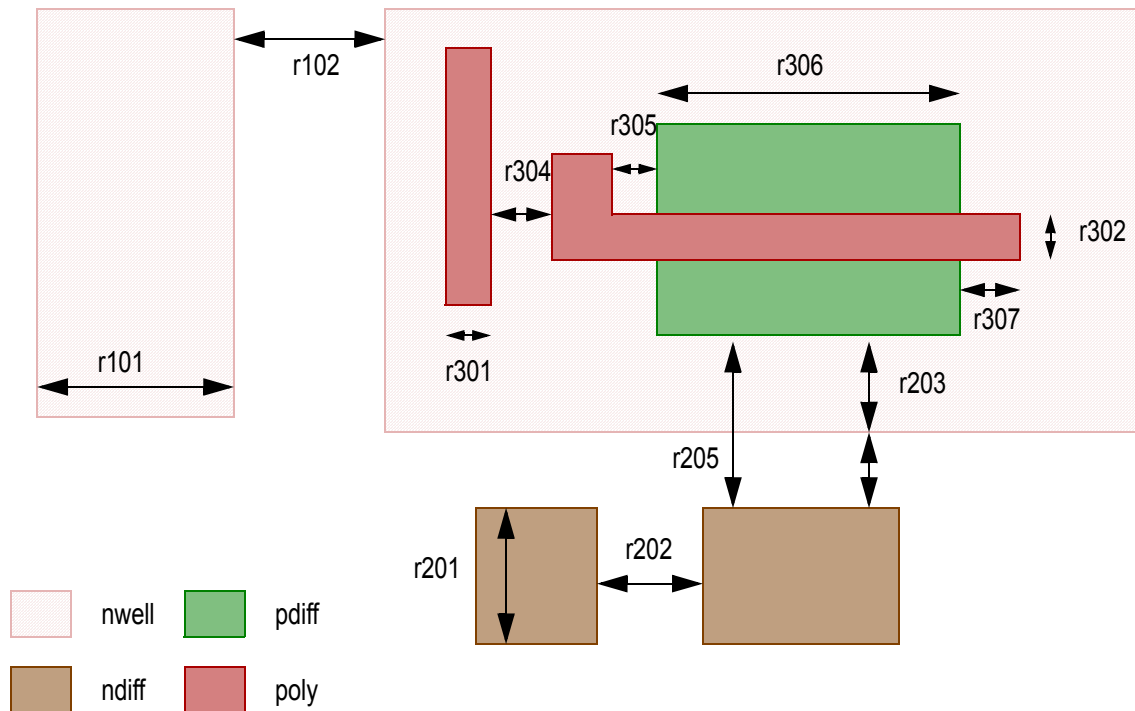


Figura 1.5.- Reglas de diseño de un proceso CMOS en tecnología de 0.12 µm.

dimensiones mínimas de una tecnología de $0.12\ \mu\text{m}$, de tal forma que la velocidad de los transistores NMOS y PMOS sea lo más parecida posible. Antes de continuar cabe destacar que las dimensiones importantes de un transistor MOS son la anchura y longitud de la puerta o canal ya que la intensidad del transistor será proporcional a su relación. Cualitativamente hablando, el canal representa la resistencia a que la corriente pase del drenador a la fuente; por lo tanto, la corriente será mayor cuanto más ancho (W) y corto (L) sea.

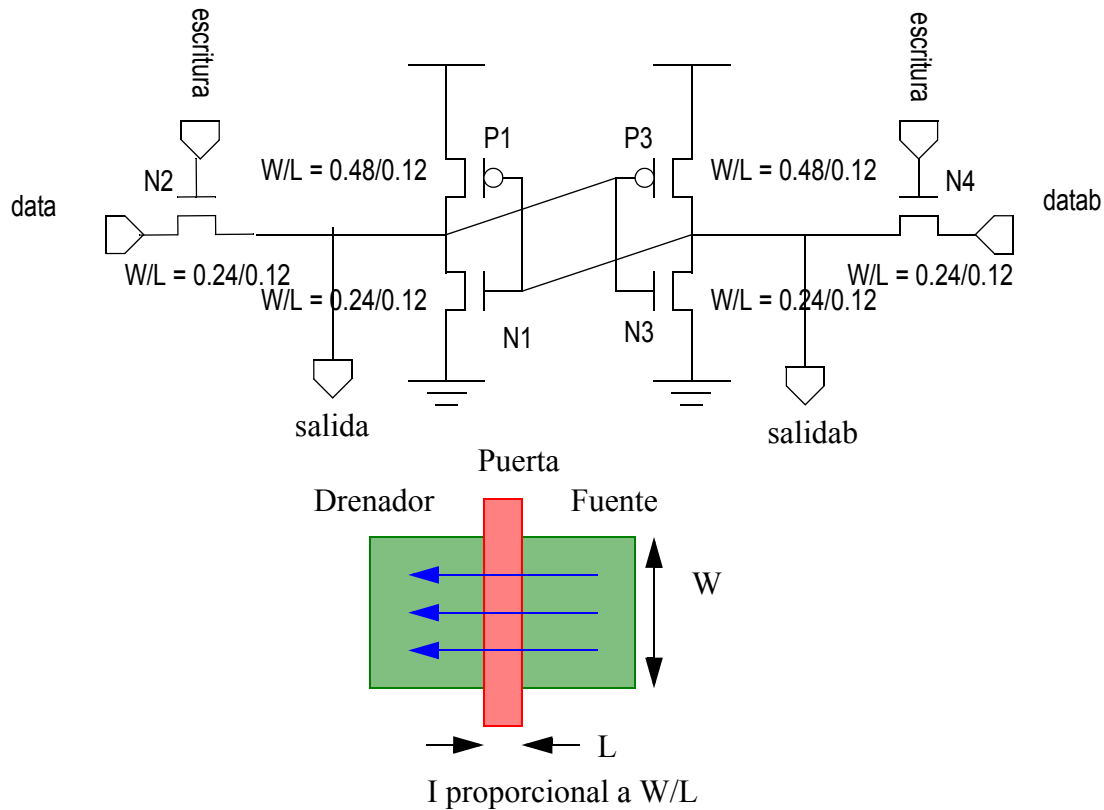


Figura 1.6.- Esquema de una celda de una memoria RAM dinámica con un dimensionado de una tecnología de $0.12\ \mu\text{m}$.

Como podemos apreciar de las reglas de diseño, la longitud mínima de la puerta es de $2 \cdot 0.06 = 0.12\ \mu\text{m}$ (regla r301), y la anchura mínima de la puerta es de $4 \cdot 0.06 = 0.24\ \mu\text{m}$ (regla r306). En el caso de los transistores PMOS se sabe que su movilidad (parámetro que mide la velocidad de los electrones) es menor que en el caso de los transistores NMOS (típicamente la mitad), así que dicha disminución es contrarrestada aumentando al doble la razón W/L con respecto a la de los transistores NMOS, es decir, $W = 0.48\ \mu\text{m}$.

Una posibilidad del layout se muestra en la figura 1.7. Existen muchas estrategias para la construcción de un layout, no obstante, particularmente suelo abordar dicha construcción siguiendo la estructura del esquemático (en este caso particular figura 1.6). Se suele empezar colocando un transistor, y continuar con otros transistores que estén conectados al ya colocado, así como sus conexiones. Debido a la existencia o no del nwell, se suelen colocar transistores del mismo tipo (NMOS o PMOS).

Con respecto a las conexiones hay que indicar que dos capas del mismo material que se tocan están conectadas, así que no hace falta realizar todas las conexiones con capas de metal. Cuando se quiera conectar dos capas de diferente material es necesario recurrir a los contactos

(ya sean de difusión o polisilicio con la primera capa de metal) o a las vías (para conectar las diferentes capas de metal entre sí).

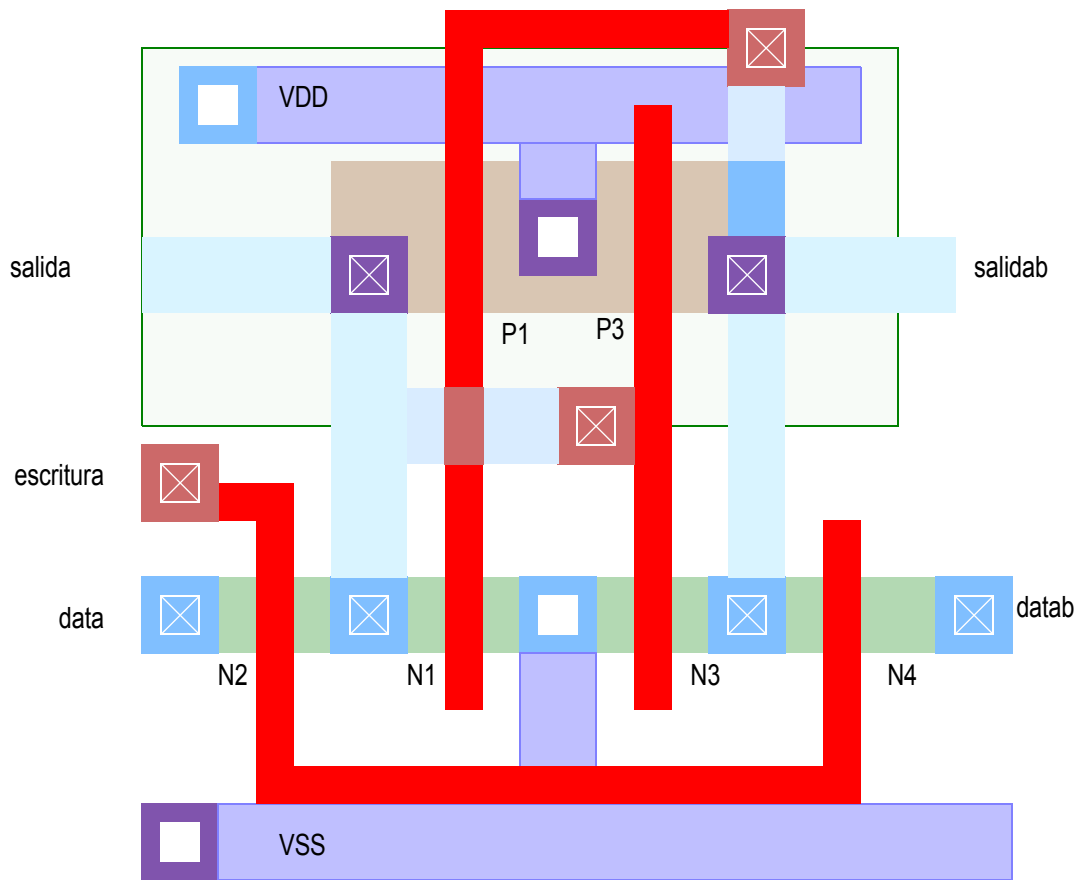


Figura 1.7.- Layout de una celda correspondiente a una memoria RAM dinámica.

Centrándonos en la celda que nos ocupa, en primer lugar colocaríamos los transistores NMOS. Como todos están conectados entre sí, los podemos colocar en una misma tira de difusión n atravesada por cuatro tiras de polisilicio (las cuales van a formar las puertas de los cuatro transistores). Los terminales de drenador y fuente que estén conectados a más de un transistor (en su drenador o fuente) necesitan disponer de un contacto de difusión con metal para realizar la posterior conexión; la excepción de esta regla son las conexiones de masa (y polarización) ya que necesitan conectarse al sustrato (y al nwell) para mantener la polaridad (dicha conexión nunca aparece en los esquemáticos). Cuando los terminales de drenador y fuente están conectados, únicamente es necesario una capa de difusión para los dos (como es el caso de los transistores N2 y N1), en el cual se ha elegido poner un solo contacto. La influencia de colocar más de un contacto es que puede pasar una cantidad mayor de corriente, y por lo tanto, tendrá mayor rapidez; así que hay que llegar a un compromiso entre la rapidez y el área ocupada.

Seguidamente, colocaríamos los transistores PMOS con su nwell correspondiente. En este caso seguiríamos las directrices de los transistores NMOS. La ubicación de los transistores PMOS ha sido elegida de tal forma que nos facilite la conexión de las puertas con las de los transistores NMOS. En este caso, como la anchura del drenador es mayor podríamos intentar

poner más de un contacto de difusión; no obstante, no podría ser (sin aumentar el área) debido a la restricción de espaciado entre contactos. En este caso, tenemos cierta libertad para mover el contacto a lo largo de la capa de difusión.

Ahora realizaremos las conexiones (que nos faltan ya que algunas ya se han realizado: drenadores y fuentes adyacentes). Por lo general las conexiones se realizarán con capas de metal (por tener menor resistividad que el polisilicio), pero cuando tengamos que conectar dos capas de polisilicio que se encuentren relativamente cerca, dicha conexión se hará con el mismo polisilicio (como el caso de los transistores N1-P1, N3-P3 y N2-N4). Para realizar las conexiones de diferentes capas, habrá que incluir contactos y/o vías para llegar a los niveles deseados (como puede ser las conexiones de drenador y fuente de N1-P1, N3-P3, y drenador y puerta de N1-N3). Hasta ahora solamente hemos utilizado una capa de metal.

Por último, colocamos los raíles de polarización y de masa. Para un mejor manejo de estas señales (y poder “olvidarnos” de ellas), se suelen utilizar capas de metal dedicadas (que solamente se utilizan para dichos raíles), de tal forma que puedan atravesar un bloque asegurando que no se producirá ninguna conexión interna. En este caso hemos utilizado una segunda capa de metal, por lo que es necesario la introducción de vías y contactos (para conectar difusión-metal1-metal2), los cuales pueden solaparse en esta tecnología. Dichos raíles deben estar conectados a los llamados pozos, que no son más que la conexión de los sustratos de cada transistor. Dichos pozos estarán formados por una capa de difusión n en el nwell y una capa de difusión p fuera del nwell.

Siempre es importante sacar a los extremos los terminales del circuito que estemos tratando (en este caso: escritura, data, datab, salida, salidab). Dichos terminales suelen realizarse con capas de metal (por la condición de resistividad a la que aludimos anteriormente).

1.3. Técnicas de diseño microelectrónico

De forma previa a la creación de un layout, debemos conocer la estructura a nivel de transistores del circuito, para lo cual debemos descender al nivel de jerarquía eléctrico.

Las técnicas de diseño no son más que unos procedimientos para desplazarnos a través de los diferentes niveles de jerarquía del circuito. En función del sentido del movimiento tenemos flujos *top-down*, que van desde el nivel superior al inferior, y flujos *bottom-up*, que van del nivel inferior al superior.

El flujo de diseño *top-down* parte de una descripción global del sistema (sin precisar las diferentes partes que lo forman ni especificaciones cuantitativas); para a partir de dicha descripción ir refinando cada vez más las diferentes partes.

El flujo de diseño *bottom-up* parte de unas especificaciones muy detalladas para realizar las diferentes partes del sistema, las cuales serán conectadas entre sí con posterioridad.

En el caso de diseño, el flujo más utilizado es el *top-down*, ya que cuando la descripción es compleja las especificaciones detalladas son muy difíciles de cumplir; por lo tanto, nos centraremos en el primero. En la figura 1.8 se muestran los procesos típicos del flujo *top-down* junto con el nivel jerárquico asociado y su dominio.



Figura 1.8.- Procesos en el flujo de diseño top-down.

El flujo de diseño tratará de pasar desde el nivel superior del dominio de comportamiento al nivel inferior del dominio físico pasando por el dominio estructural. Veremos todo este paso con el ejemplo de un multiplicador, mostrado en la figura 1.9. El nivel superior estará compuesto por algoritmos (en su dominio de comportamiento) o por procesadores, memorias y/o buses (en su dominio estructural). En nuestro caso particular necesitaremos un procesador dedicado y un controlador (que no sería más que otro procesador dedicado); mientras que el dominio de comportamiento estaría descrito por cualquiera de los algoritmos de multiplicación disponibles, por ejemplo la multiplicación por sumas sucesivas. Obviamente, el punto de partida sería el algoritmo, ya que el conjunto procesador-controlador no nos suministra la suficiente información para continuar.

En el siguiente nivel, el comportamiento estaría descrito por un diagrama ASM en el cual estarían descritas todas las operaciones así como la secuencia en la que se han de realizar dichas operaciones. En dicho nivel se tiene la información suficiente para que la descripción estructural sea lo suficientemente precisa para que tenga un sentido lógico. Por lo tanto, en este nivel se suele realizar el proceso de síntesis.

El proceso de síntesis se puede definir como el proceso por el cual se pasa del dominio de comportamiento al estructural siguiendo una serie de restricciones.

En la figura 1.9 sólo mostramos el esquema correspondiente al procesador. De hecho, para no saturar la figura, sólo se ha considerado una celda para descender de nivel en la jerarquía.

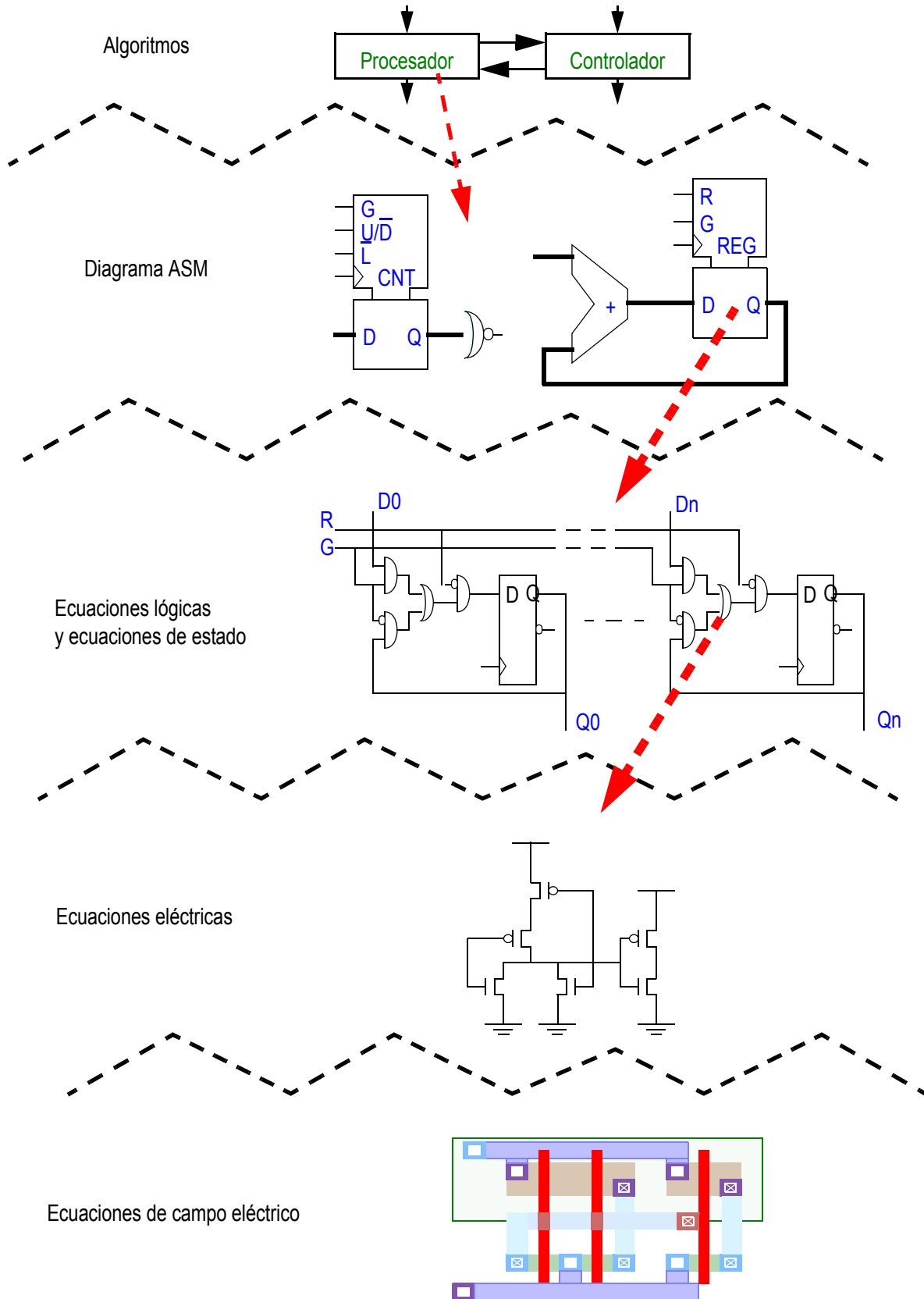


Figura 1.9.- Diferentes pasos del flujo de diseño top-down.

Una vez que tenemos los elementos de procesado del nivel RTL, debemos implementarlo con puertas lógicas y biestables. El comportamiento de este nivel estaría descrito a partir de ecuaciones lógicas (booleanas) y diagramas de estado; no obstante, la complejidad (en cuanto al número de componentes) de este nivel es lo suficientemente grande como para que el sistema de estas ecuaciones sea intratable. Como ejemplo mostramos como estaría formado el registro acumulador con las condiciones de habilitación y reset síncrono.

En el siguiente nivel se obtendrá la estructura eléctrica de los componentes lógicos: puertas y biestables; para ello, se elegirá la familia lógica que mejor se adapte a las especificaciones. En el ejemplo se ha considerado una familia lógica CMOS estática.

Por último, se debe pasar de este esquema a nivel de transistores al layout en el cual se indicarán las capas que se necesitan así como su forma y tamaño.

Un flujo de diseño ampliamente utilizado es el mostrado en la figura 1.10. En él podemos distinguir los siguientes pasos:

- partimos de una descripción informal del comportamiento deseado del circuito que queremos diseñar.
- A partir de esta descripción, generaremos una descripción en un lenguaje de descripción de hardware (como puede ser VHDL). Dicha descripción puede ser a nivel de comportamiento o directamente a nivel de estructura; no obstante, lo usual es obtener una descripción a nivel de comportamiento que con los algoritmos adecuados proporcionará una descripción a nivel de estructura.
- Esta estructura es traducida a componentes de niveles inferiores en la jerarquía como son las puertas lógicas y biestables.
- Las puertas lógicas son diseñadas o elegidas de una librería de celdas (ya se verá posteriormente la diferencia) hasta el nivel físico o layout.
- Con estos layout se hará una planificación del espacio para poder determinar el lugar en el que se colocarán las diferentes celdas.
- Una vez que se han colocado todas las celdas, se realizará el conexionado de todos los nodos del circuito.

En este curso nos vamos a centrar en los tres últimos puntos, es decir, la generación de las puertas lógicas, la planificación, colocación y conexión.

En el proceso anterior no se ha comentado nada sobre ninguna estrategia a la hora de pasar de un nivel a otro. Realmente, las únicas diferencias se encuentran en las dos últimas etapas: el nivel eléctrico y físico. Todas las estrategias se pueden dividir en dos grandes grupos:

- Circuitos *full-custom* o completamente a medida. En este caso, como su propio nombre indica, todas las puertas son realizadas por el diseñador de tal forma que se obtenga una mejor adaptación a las especificaciones del diseño. Esta realización propia será a nivel eléctrico (esquema de transistores con las dimensiones específicas) y físico (layout, que podemos crear layouts diferentes para una misma celda con el fin de que *encaje* mejor en el circuito reduciendo el área ocupada).
- Circuitos *semi-custom* o semi-medida. En este caso, se utiliza una librería de celdas que impone el esquema a nivel eléctrico y el layout de los diferentes componentes. La

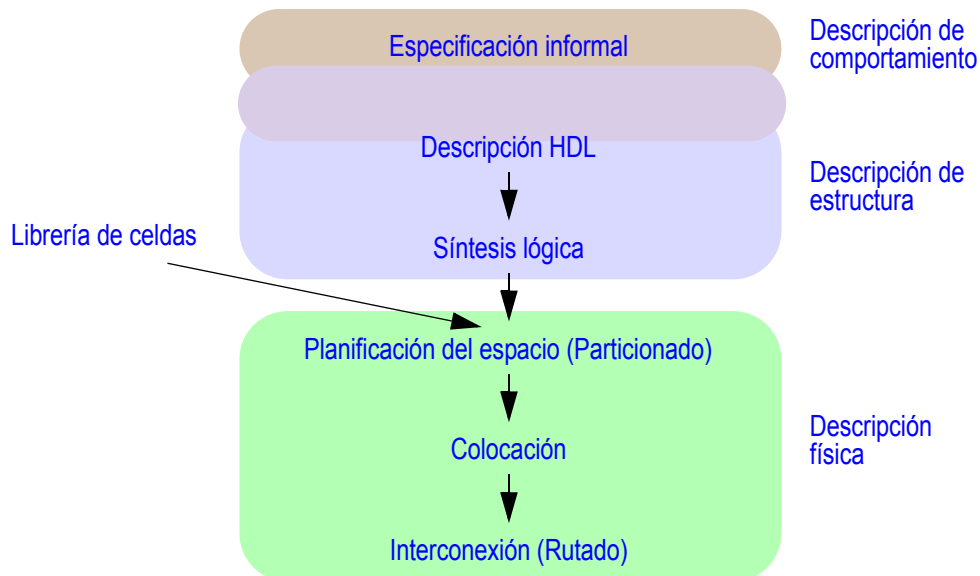


Figura 1.10.- Flujo de diseño.

única libertad que le queda al diseñador en estos niveles es la colocación de las celdas en el layout final (siempre y cuando se cumplan una serie de restricciones, como es su colocación en hileras). El destino de esta estrategia puede ser un circuito integrado, denominado entonces basado en celdas, o un dispositivo programable, denominado entonces basado en matrices.

Como es obvio, ambas estrategias tendrán características diferentes. Así los circuitos *full-custom* son los que presentan mejores prestaciones a costa de un mayor tiempo de diseño, debido a la realización de todas las celdas. Por el contrario, los circuitos *semi-custom* mejoran los tiempos de diseño a costa de un empeoramiento en las prestaciones. Dentro de los circuitos *semi-custom*, aquellos basados en matrices son los que tienen un tiempo de diseño menor ya que no es necesaria el proceso de integración ya que se dispone del dispositivo, que con un proceso de programación adquirirá el comportamiento adecuado. Luego, no podemos encontrar el circuito ideal en el que se optimicen simultáneamente el tiempo de diseño y sus prestaciones, por lo que hay que llegar a un compromiso para decidir la estrategia con la cual abordar la tarea de diseño.

Como alternativa a los circuitos integrados (ya sean *full-custom* o basados en celdas) se encuentran los dispositivos configurables. Entre los dispositivos configurables básicos podemos distinguir los CPLDs y las FPGAs, cuyas estructuras se muestran en la figura 1.11.

Los dispositivos CPLDs (Complex Programmable Logic Devices) ofrecen una cantidad menor de puertas equivalentes. Su arquitectura está distribuida en columnas incluyendo tres tipos de componentes: IOC (Input-Output Control block), LAB (Logic Array Block) y PIA (Programmable Interconnect Array). La arquitectura de estos dispositivos producen circuitos con características lógicas predecibles, por lo que son utilizados en componentes críticos en los que el conocimiento a priori de las características temporales es fundamental.

Los dispositivos FPGAs (Filed Programmable Gate Arrays) suelen ser los dispositivos con una mayor cantidad de puertas equivalentes. Su arquitectura distribuye bidimensional-

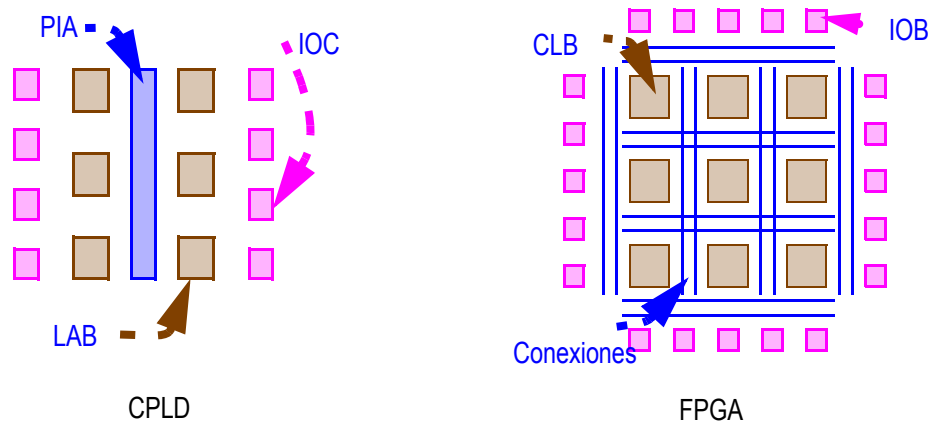


Figura 1.11.- Dispositivos configurables.

mente sus tres elementos fundamentales: IOB (Input-Output Block), CLB (Computation Logic Block) y la matriz de conexiones (esta matriz, a diferencia con los CPLDs, está distribuida en todo el dispositivo ente los CLB). En este caso la predicción de las características temporales no es simple ya que el camino que tienen que recorrer las señales no tiene porqué ser el mismo. Los CLB suelen ser mucho más simples que los LAB por lo que se encuentran en mayor cantidad. Esta situación otorga a las FPGAs una mayor versatilidad que los CPLDs, luego las aplicaciones típicas son aquellas que requieran un gran número de puertas y una arquitectura con un mayor grado de *pipeline*. También se suelen utilizar como plataforma de prototipos para sistemas que posteriormente serán implementados en un ASIC, aunque a menudo la versión final permanece en la FPGA si cumple con las especificaciones necesarias.

En ambos casos, CPLD y FPGA, la matriz de conexiones no implementa la conectividad total de los componentes lógicos. Esta situación se traduce en que la utilización del dispositivo raramente llega al 100% de puertas.

Otra alternativa también programable son los denominados ASICs estructurados o SASICs (Structured ASICs). La estructura general de un SASIC, mostrada en la figura 1.12, consiste en una matriz bidimensional o mar de módulos a los que se le unen una serie de bloques prefabricados como pueden ser bloques de entrada/salida, microprocesadores, memorias, etc. La constitución de los módulos puede variar en gran medida dependiendo de la granularidad del SASIC; estos pueden abarcar el rango desde componentes eléctricos (como transistores y resistencias) a las denominadas *lookup tables* (que no son más que módulos de memoria que implementan una función combinacional con un número determinado de entradas). En muchos aspectos, este dispositivo es muy similar a un ASIC implementado con celdas estándares ya que los módulos pueden ser este tipo de celdas. La principal diferencia es que todas las celdas y muchas conexiones (como las de tierra y polarización) ya están integradas, por lo que el proceso de integración se limita a especificar unas cuantas capas de metal (aquellas correspondientes a conexiones específicas).

Estos nuevos dispositivos, los SASICs, vienen a cubrir el espacio de aplicaciones existente entre los ASICs y los dispositivos programables comunes (FPGAs y CPLDs). Un SASIC tendrá unas prestaciones similares a un ASIC basado en celdas y un coste considerablemente menor debido a que únicamente se integran unas pocas capas de metal.

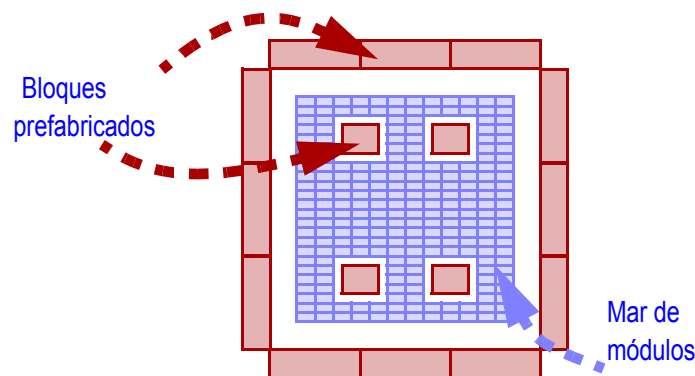


Figura 1.12.- Estructura de SASIC.

1.3.1. Parámetros de calidad de un circuito digital

Una vez que se ha diseñado un circuito digital, hay que medir la calidad de dicho circuito, puesto que circuitos con poca calidad serán despreciados en beneficio de aquellos con una calidad mayor. Entre los principales parámetros de calidad, podemos destacar el coste, la funcionalidad, la robustez, la velocidad y el consumo; alguno de ellos tienen un carácter cualitativo y otros cuantitativo.

1.3.1.1. Coste de un circuito integrado

El coste de un circuito integrado se puede dividir en dos componentes: un coste variable, que es directamente proporcional a la cantidad de circuitos integrados, y un coste fijo, que es independiente de dicha cantidad. Luego el precio total de la producción será:

$$\text{coste total} = \text{coste variable} \times \text{cantidad de circuitos} + \text{coste fijo}$$

y el coste de una unidad será:

$$\text{coste por unidad} = \text{coste variable} + \text{coste fijo} / \text{cantidad}$$

En el caso de los circuitos integrados ASIC, la componente predominante es el coste fijo, por lo tanto un circuito será más rentable cuanto mayor sea la producción. Si consideramos un dispositivo programable, la componente predominante es el coste variable, por lo que serán rentables las producciones pequeñas (puesto que se aumentaría el coste de forma considerable en caso contrario). La situación intermedia estaría ocupada por los SASICs, como se muestra en la figura 1.13. Los límites de producción en los que cada una de las soluciones tendría mejor coste vendrían dados por las siguientes ecuaciones:

$$\text{coste variable}_{\text{FPGA}} + \text{coste fijo}_{\text{FPGA}} / \text{cantidad}_1 = \text{coste variable}_{\text{SASIC}} + \text{coste fijo}_{\text{SASIC}} / \text{cantidad}_1$$

$$\text{coste variable}_{\text{SASIC}} + \text{coste fijo}_{\text{SASIC}} / \text{cantidad}_2 = \text{coste variable}_{\text{ASIC}} + \text{coste fijo}_{\text{ASIC}} / \text{cantidad}_2$$

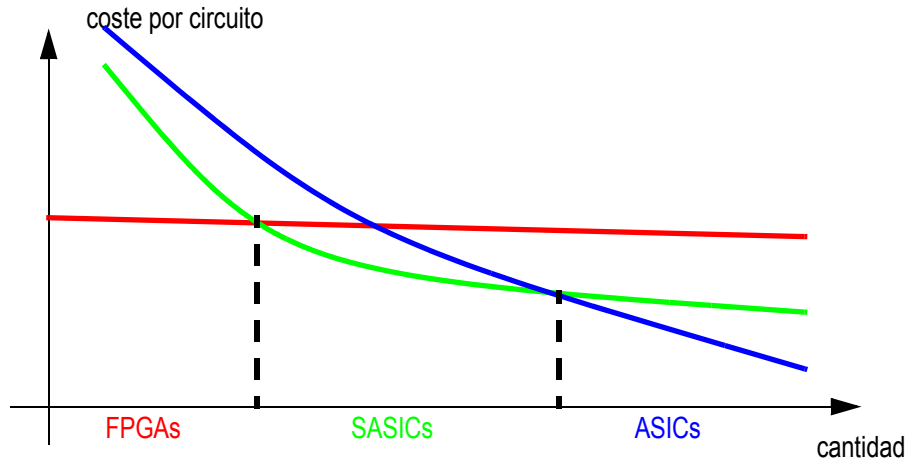


Figura 1.13.- Coste por circuito para las tres alternativas: dispositivos programables, SASICs y ASICs.

1.3.1.2. Funcionalidad y robustez

Un requerimiento primordial es que el circuito tenga el mismo funcionamiento para el que fue diseñado. No obstante, las variaciones en el proceso de diseño e integración pueden provocar variaciones con respecto al comportamiento esperado (obviamente sólo se permiten variaciones cuantitativas, en los parámetros de diseño, y no cualitativas, en la funcionalidad lógica). Otra fuente de variación es el ruido entendido como cualquier variación no deseada de tensión y/o intensidad en los nodos lógicos.

En cambio, la robustez es un parámetro cualitativo que nos da idea del rango de condiciones de operación para las cuales el circuito tiene una funcionalidad adecuada. Algunos parámetros de más bajo nivel que nos mide la robustez son los márgenes de ruido, la inmunidad al ruido y la propiedad regenerativa.

Los márgenes de ruido se pueden definir como aquellos intervalos en los cuales la señal de entrada puede variar (por ejemplo a causa del ruido) sin que su valor lógico cambie, como se muestra en la figura 1.14. Luego tendremos un margen de ruido por cada nivel lógico, los cuales no tienen porqué coincidir, de hecho, en la mayoría de los casos serán diferentes. Luego, para que el circuito sea lo más robusto posible, debe disponer de márgenes de ruido lo más grandes posible para que puedan absorber variaciones grandes de señal.

La inmunidad al ruido se puede definir como la capacidad de rechazar al ruido, en lugar de absorberlo (que sería el caso de los márgenes de ruido). Un ejemplo de sistemas altamente inmunes al ruido son las familias diferenciales que trabajan con la diferencia de dos señales. Al realizar la diferencia de dos señales, también se obtiene la diferencia del ruido de ambas señales lo cual provocará una reducción del ruido (sin tener en cuenta los márgenes de ruido).

La propiedad regenerativa es aquella por la cual partiendo de señales de entrada degradadas se obtiene una señal de salida no degradada. Esta propiedad es fundamental para que la degradación de las señales de una puerta por efecto del ruido no afecte a las que se encuentren conectadas a ella.

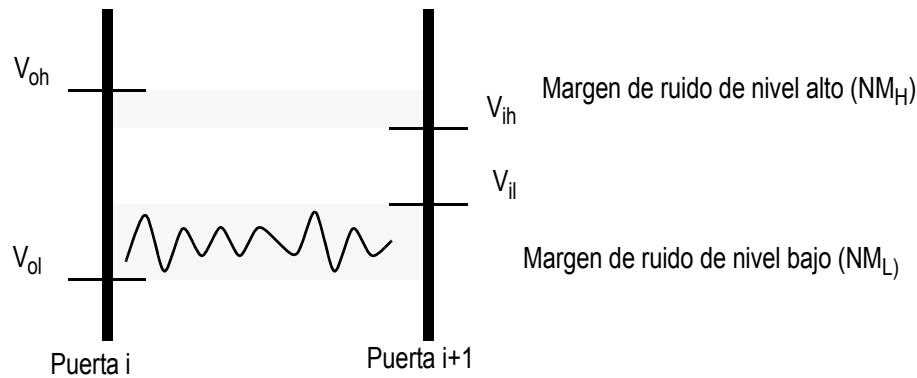


Figura 1.14.- Márgenes de ruido de un circuito digital.

1.3.1.3. Velocidad

La velocidad de un circuito digital se suele medir por la tasa de datos de salida que es capaz de procesar, que equivaldría en un circuito síncrono a la frecuencia de la señal de reloj, o bien su inversa como el periodo de la señal de reloj. Este valor vendrá dado por la siguiente ecuación:

$$\text{retraso total} = \text{retraso de propagación de la lógica} + \text{retraso de propagación de registros} + \text{restricciones temporales}$$

donde el retraso de propagación (ya sea de la lógica o de los registros) es el tiempo necesario para que se realice la operación de la etapa de procesado o del registro del pipeline, y las restricciones temporales son las restricciones impuestas por los registros para asegurar el correcto almacenamiento de los datos (tiempo de *setup* y de *hold*).

1.3.1.4. Consumo de potencia

El consumo de potencia nos va a indicar la cantidad de energía necesaria por cada operación, la cual usualmente será disipada como calor. Por lo tanto, nos enfrentamos a un doble problema en el caso de un elevado consumo: la capacidad de la fuente de polarización debe ser alta, lo cual está en completa contradicción con los sistemas denominados sin cable que operan con baterías (ya que se reduciría el periodo autónomo del sistema); y se deben elegir técnicas para la disipación del calor sin que afecte al funcionamiento del circuito. Luego, siempre que sea posible trataremos de utilizar soluciones de baja potencia.

No obstante, el comportamiento del consumo de potencia y la velocidad son complementarios, es decir, si la velocidad aumenta (mayor optimización), el consumo de potencia también aumenta (menor optimización). Ante esta situación, lo usual es tratar de optimizar un parámetro que los relaciona a ambos como es el producto retraso-potencia (PDP), que encuentra la solución de menor consumo de potencia a la máxima velocidad posible.