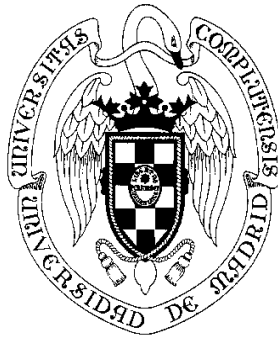


PRACTICAS DE FULL-CUSTOM



JUAN LANCHARES DÁVILA

**DEPARTAMENTO
DE
ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA**

UNIVERSIDAD COMPLUTENSE DE MADRID

1. LOGICA COMBINACIONAL ESTÁTICA

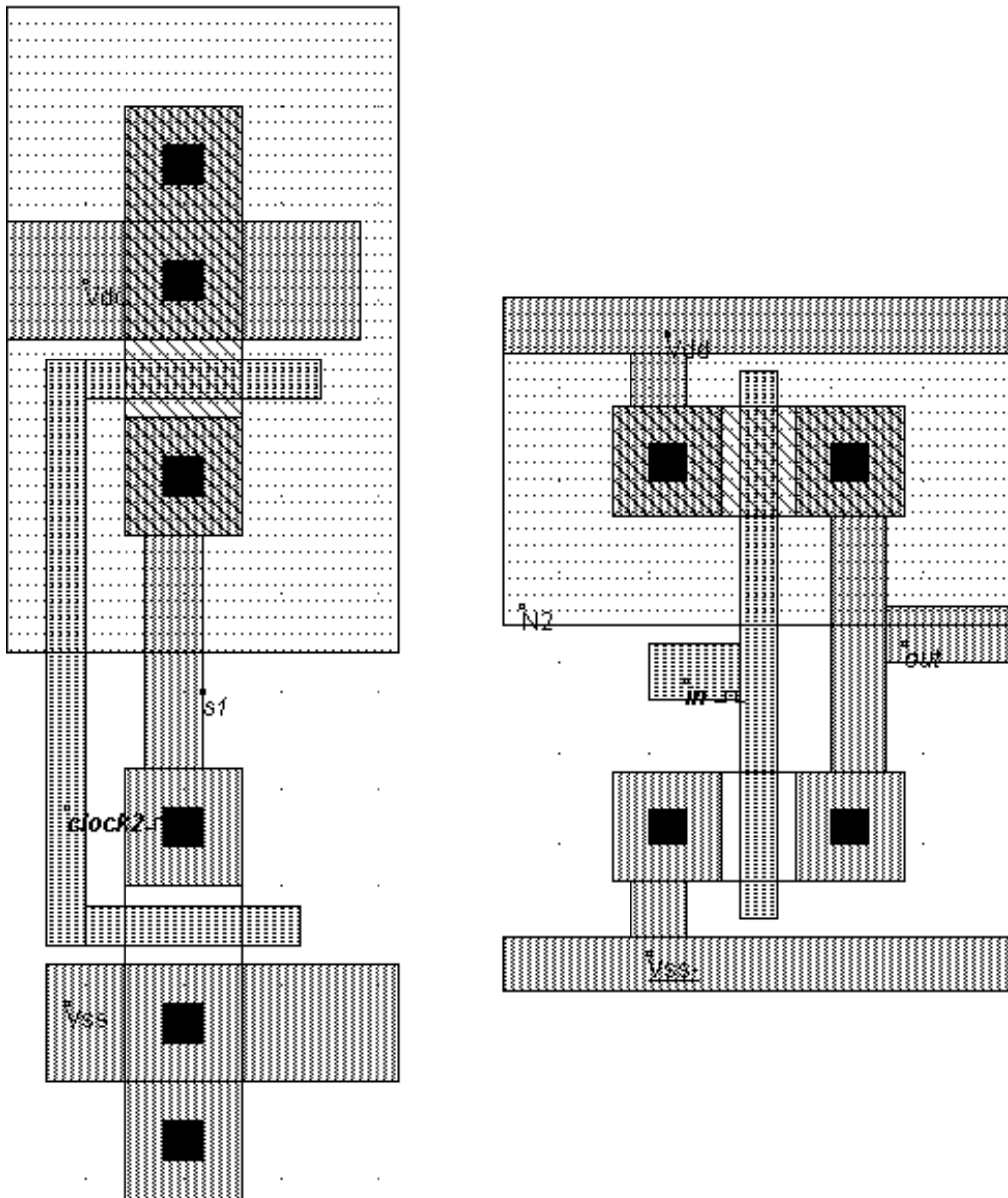
1.1 LÓGICA DE CARGA DINÁMICA

1.2 INVERSOR CMOS COMPLEMENTARIO

La lógica CMOS complementaria es la más extendida en el diseño de circuitos integrados. Aproximadamente el 80% de los circuitos se diseñan con esta lógica. Los motivos por lo que esto es así son la robustez y el bajo consumo de potencia estática.

a.- Comprobar estos extremos diseñando y simulando un inversor CMOS de lógica complementaria de tamaños mínimos.

b.- En la simulación anterior se puede comprobar que los tiempos de subida y de bajada del inversor no coinciden. Corrige el layout para que estos tiempos queden aproximadamente simétricos.



1.2.1 PUERTA COMPLEJA CMOS COMPLEMENTARIA

A) Implementa una puerta NAND de fan-in 4. (figura 1)

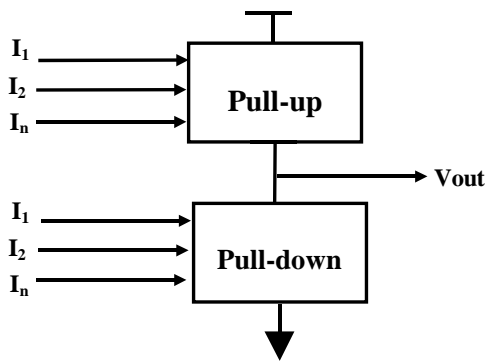


Figura 1

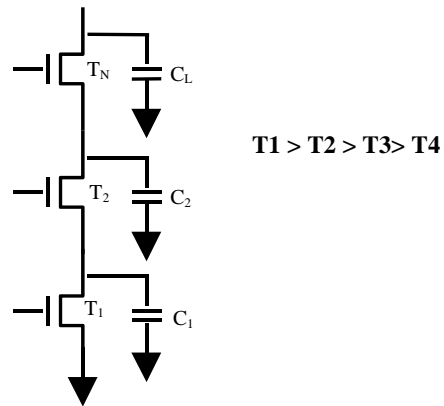
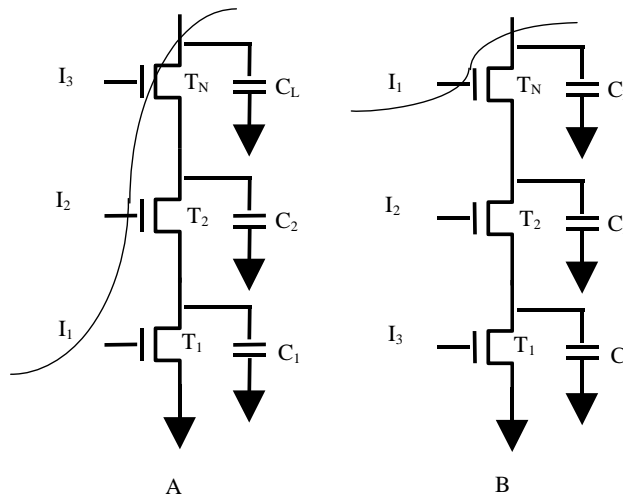
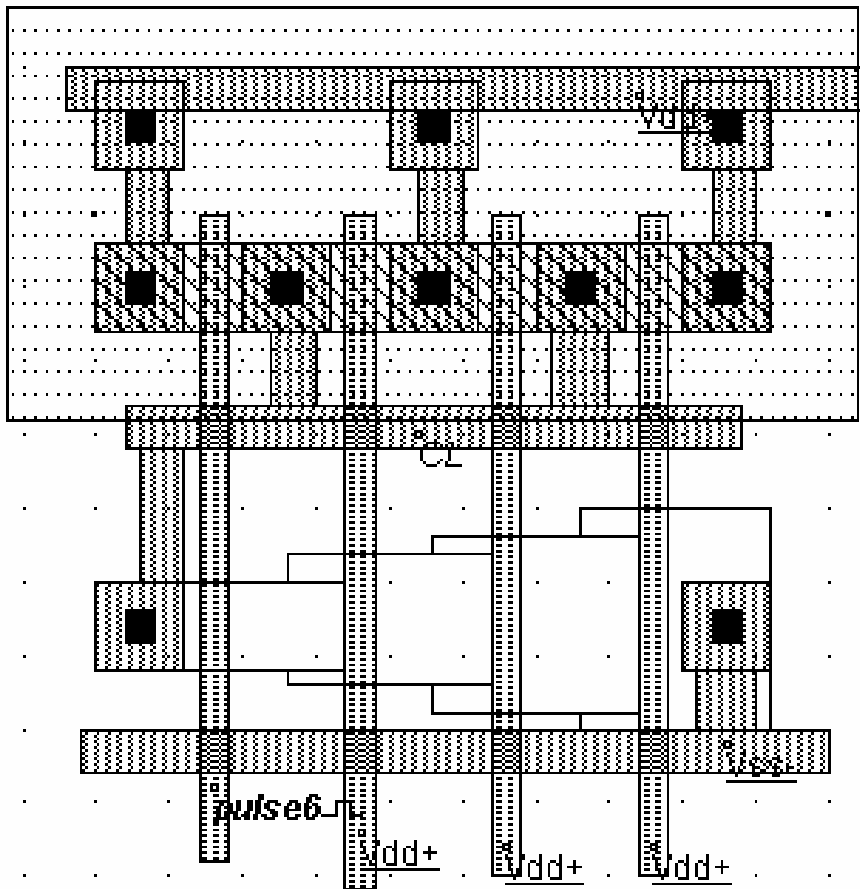
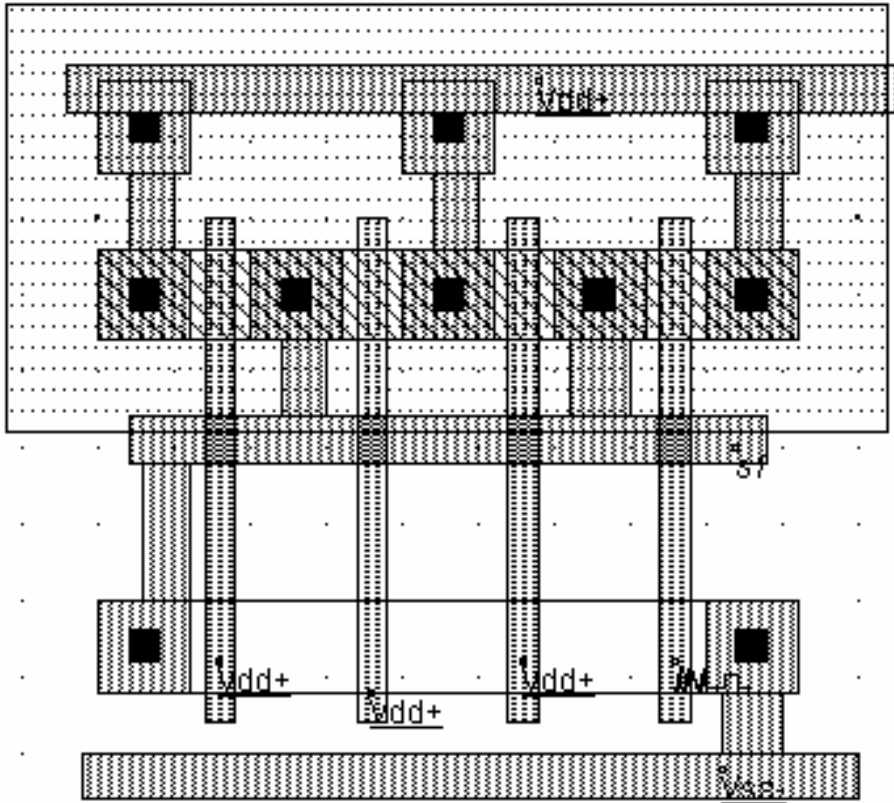


Figura 2

B) En este tipo de puertas en las que existen mucho transistores en serie existe el problema del retardo que se genera. Para evitarlo se podrían hacer los transistores serie más anchos. El problema de esta solución es el consumo de área. La solución ideal es aumentar el canal de los transistores progresivamente. Esta técnica se describe en la página 197 de [Rabaey] y 290 de [weste] Weste indica que esta técnica aplicada a tecnologías antiguas (es decir, tecnologías no submicra) puede conseguir mejoras entre el 20 y 30 de rendimiento. En cambio en tecnologías submicra no consigue mejoras superiores al 4%. Aplicarla a la puerta NAND diseñada (figura 2). Para ello suponer que $C_L=2C$ y que $C_1=C_2=C_3=C_4=C$

c) Por otro lado, el orden de llegada de las señales a las puertas puede afectar al rendimiento del sistema. Supón que llegan a la puerta NAND cuatro señales en tiempos diferentes. Busca cual será el mejor orden de llegada de las señales y cual será el peor orden de llegada.(figura3) Atención: se debe tener cuidado al elegir el paso de simulación.





1.3 LÓGICA DE CARGA ESTÁTICA

1.3.1 LÓGICA PSEUDO NMOS

A) Implementa una puerta NAND de cuatro entradas en lógica pseudonmos.

B) Observa como se modifican los márgenes de ruido de baja, los tiempos de subida y de bajada, el área y el consumo de potencia cuando voy modificando los canales p desde su tamaño mínimo a un tamaño 8 veces el tamaño mínimo. Supón los transistores N de tamaño mínimo.

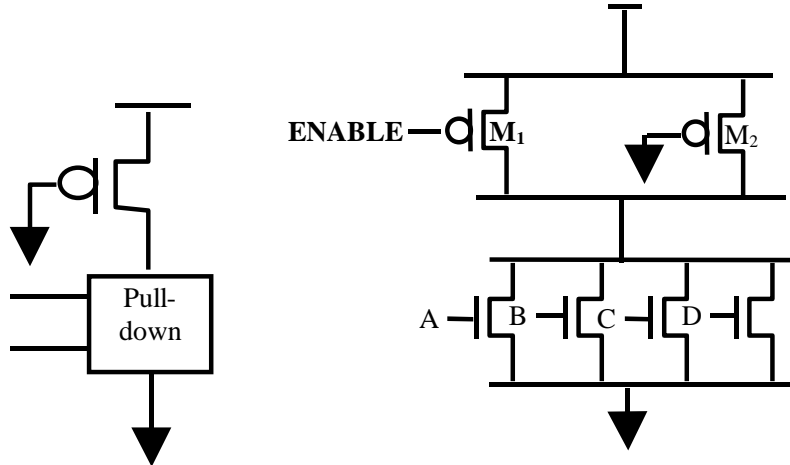
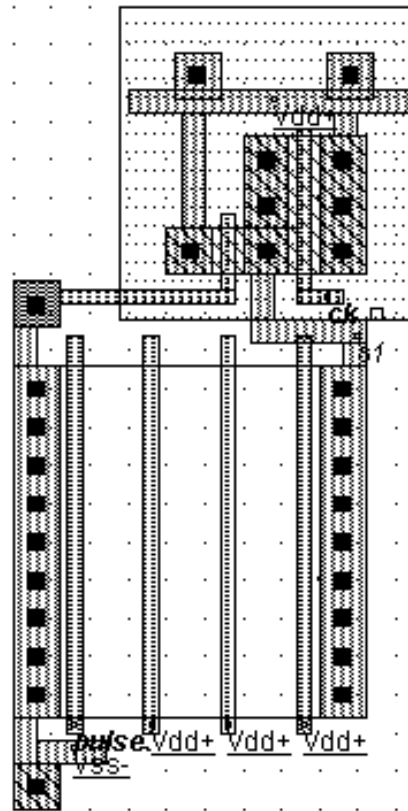
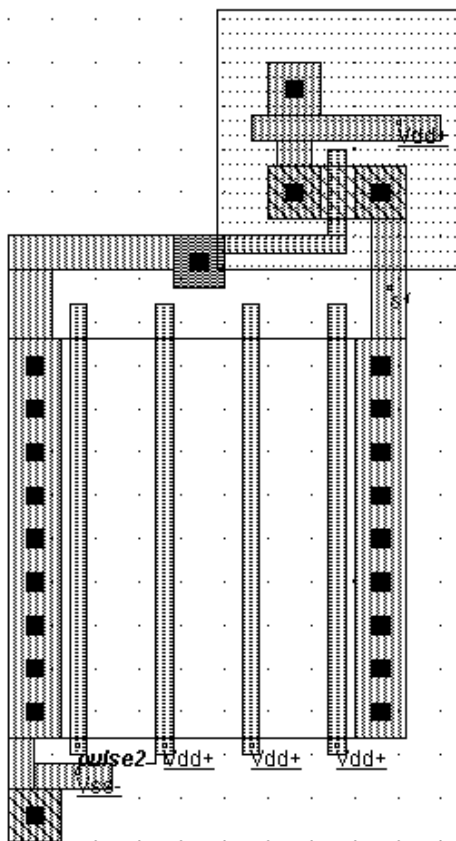
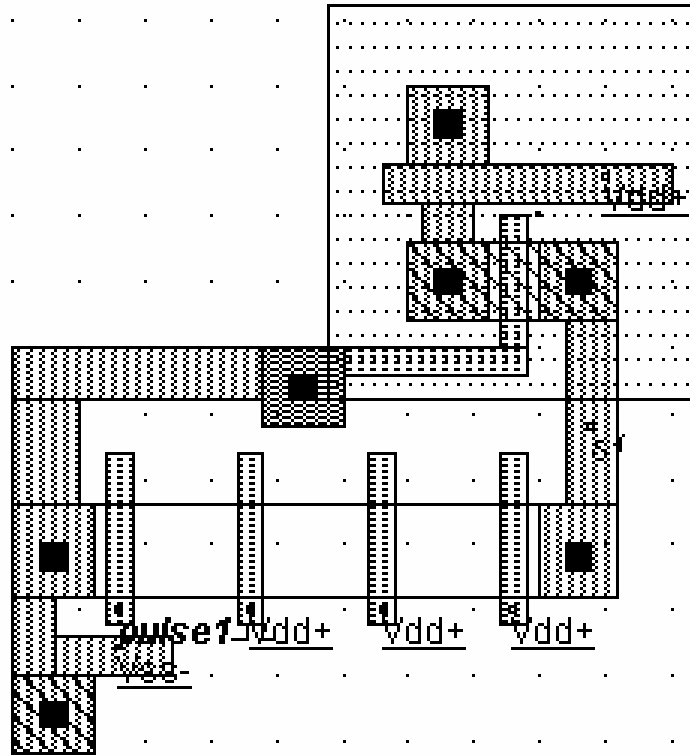


Figura X

Figura y

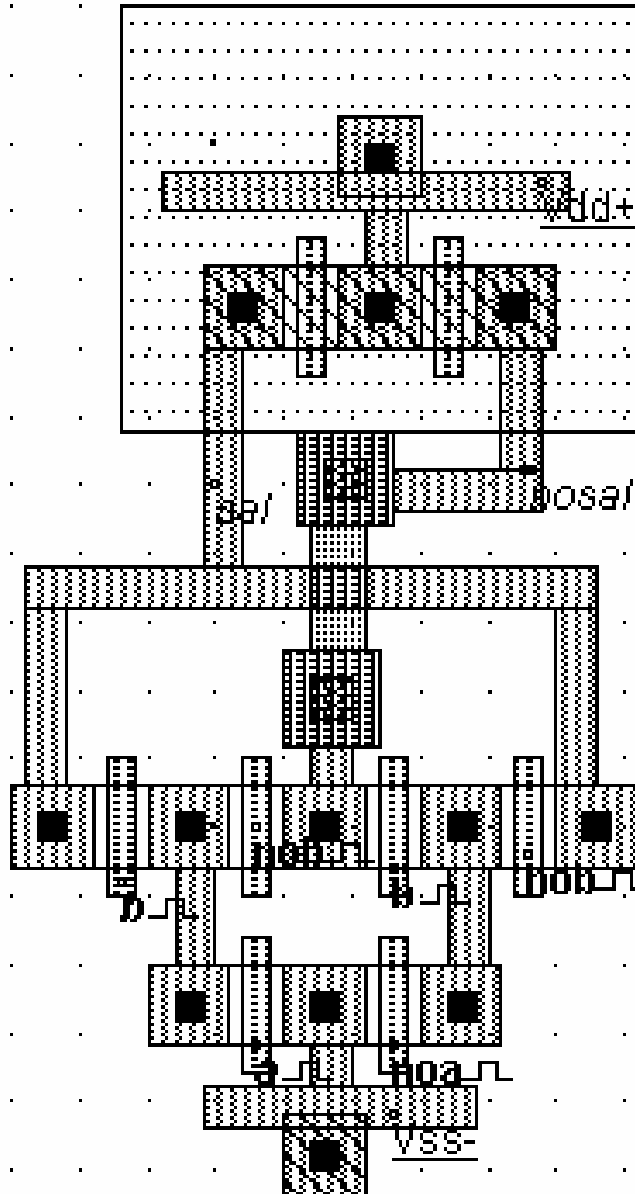
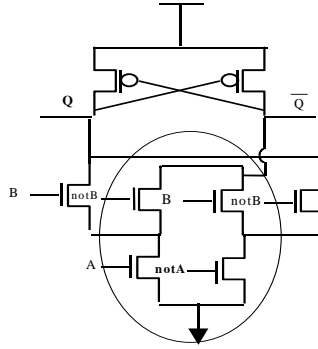
C) Como se ha podido comprobar en las simulaciones de la puerta ya vistas, los tiempos de subida de la señal son extremadamente largos comparados con los de bajada. Esto se debe a la gran capacidad que hay que cargar a la salida, sobre todo cuando se han escalado los transistores NMOS para conseguir buenos ceros lógicos. Esto se podría solucionar haciendo el transistor P más grande pero entonces tendríamos de nuevo un mal cero lógico, además de un mayor consumo de potencia.

Otra solución es la que se presenta en la figura y. Se añade un transistor p controlado por una señal ck en paralelo al inicial. Este esquema es muy útil para los decodificadores de memoria. A nosotros nos interesa que la señal que selecciona la palabra de memoria cambie muy rápidamente de 0 a 1 incluso a expensas de que se consuma mucha potencia. Una vez que la señal ya está a uno se desactiva la señal de reloj y el decodificador queda en una fase de standby. Esta señal de reloj se podría activar cuando variara la dirección a la que se desea acceder.



1.3.2 DIFERENTIAL CASCADE VOLTAGE SWITCH (DCVS)

Un tipo particular de carga estática. Particularmente interesante para implementar puertas **XOR** como la de la figura. Su principal ventaja reside en generar las dos fases de la puerta y en que comparte lógica las dos ramas.



1.4 LÓGICA DE TRANSISTORES

A) Implementa el circuito de la figura x. Comprueba que la señal de entrada se degenera en V_1 . ¿qué cantidad se degenera y porque? ¿aparece consumo de potencia estática? ¿Por qué? Modifica los potenciales umbrales e los p y de los n hasta conseguir que aparezca consumo de potencia estática. Razonalo.

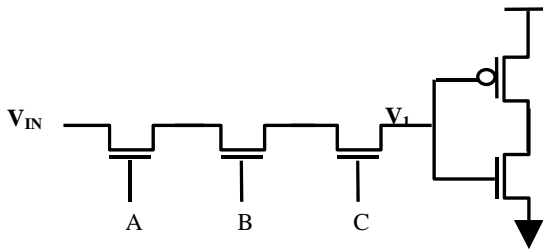


Figura X

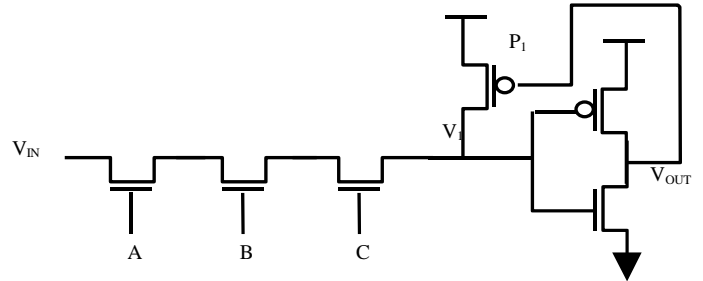
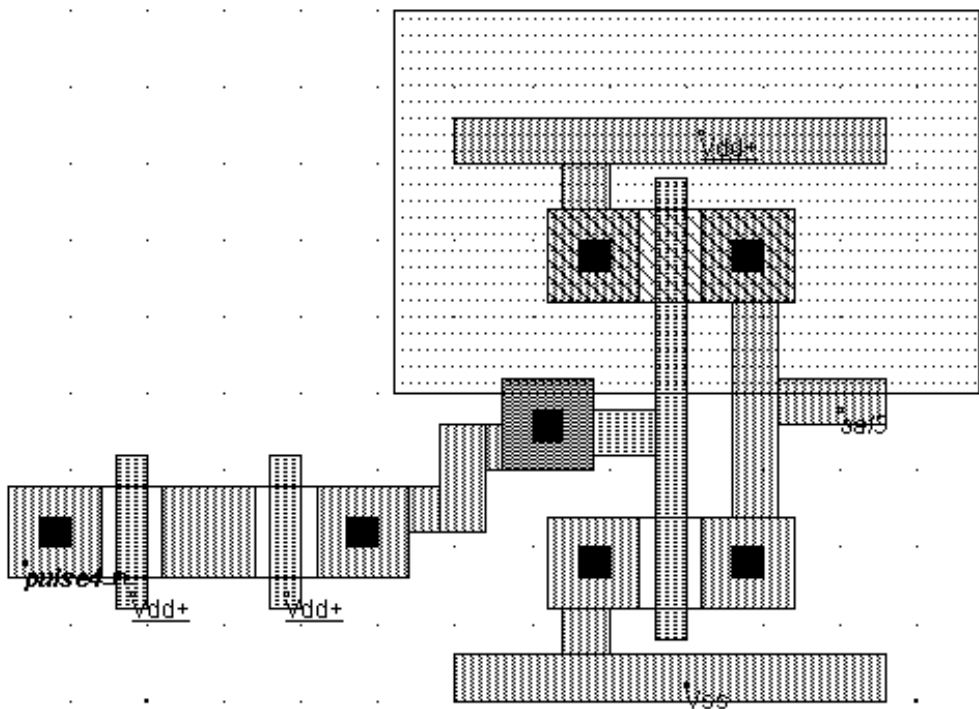
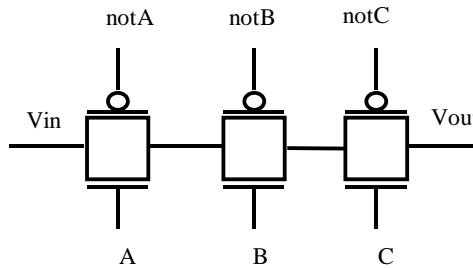
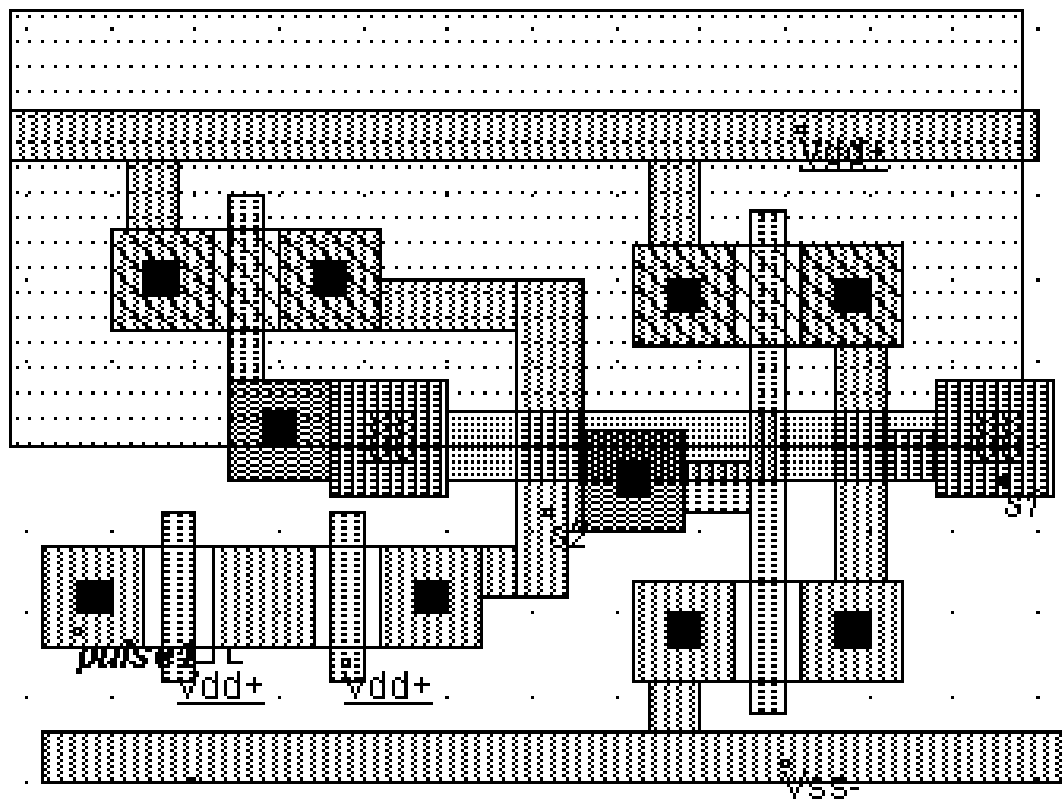


Figura y

B) Implementa la solución al problema que aparece en la figura y. Realizar el diseño de manera que sea válido tanto para $V_{in}=0$ como para el $V_{in}=1$.

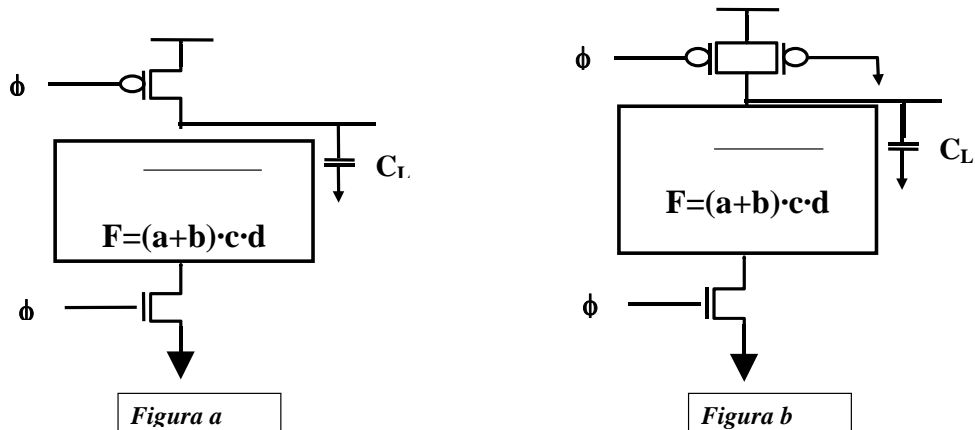
C) Implementar la misma función con puertas de transmisión y comparar resultados (Figura z).





2. LÓGICA COMBINACIONAL DINÁMICA

2.1 LÓGICA DE PRECARGA Y EVALUACIÓN



- Implementa una función $F=(a+b) \cdot c \cdot d$ con una CMOS complementaria y técnica de diseño de celdas estándar. Para ello calcula el mejor orden de llegada de las entradas utilizando el grafo lógico.
- Implementar la función con lógica de precarga y evaluación (figura a). Indica claramente que ocurre en las fases de precarga y evaluación
- Estudia en que casos se podría producir distribución de carga en el caso anterior y simúlalo.
- Aplica la solución al problema de la distribución de carga que aparece en la figura B y comenta las repercusiones que tiene sobre la evaluación, la precarga, el área y el consumo

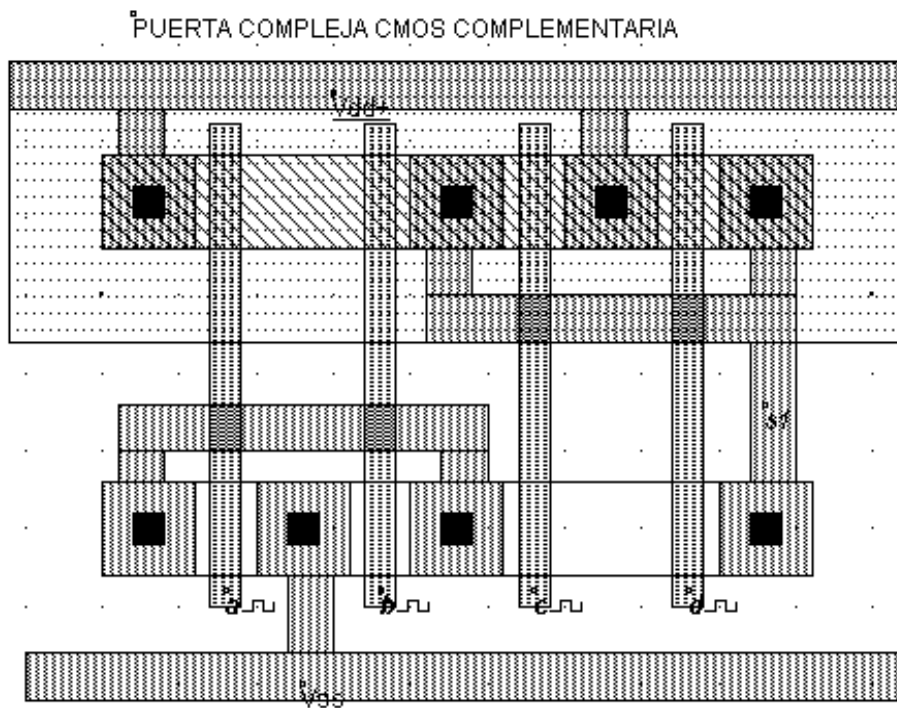
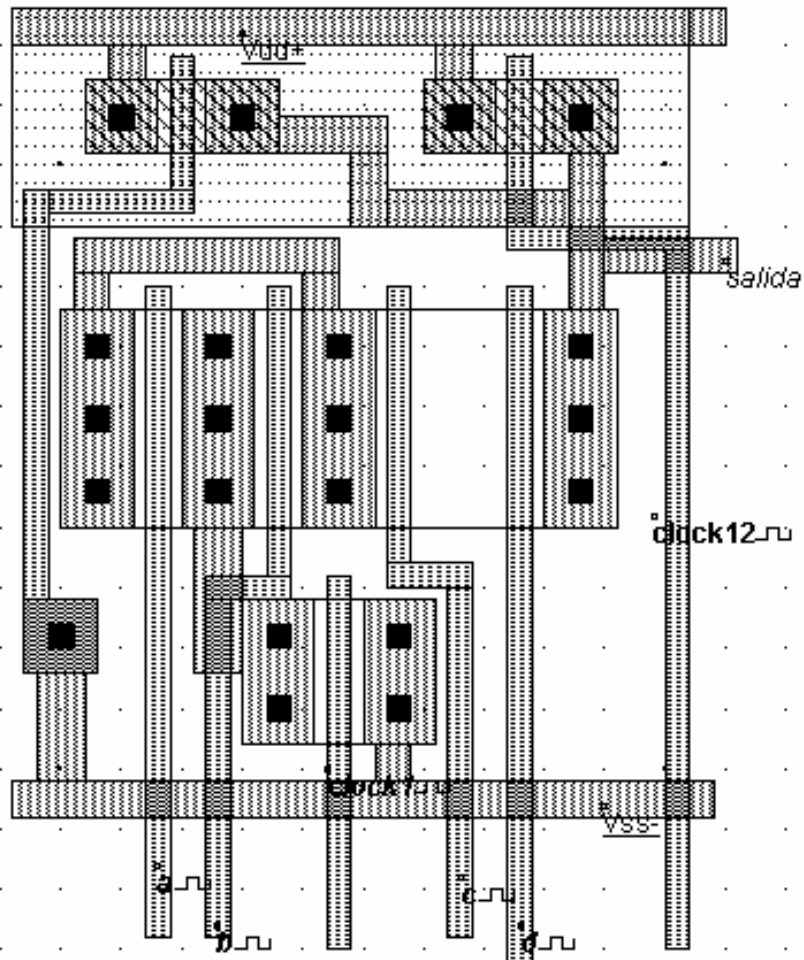
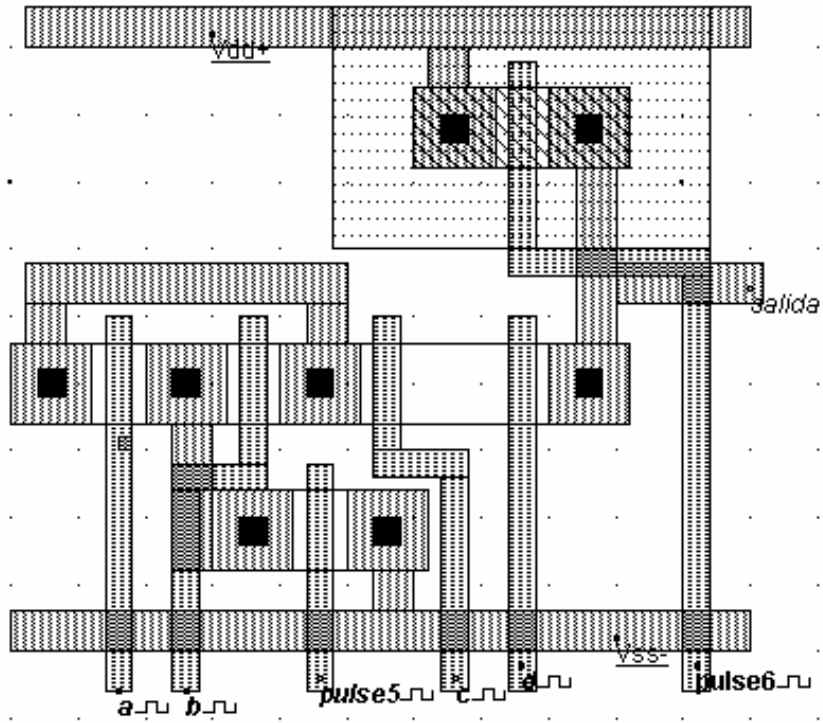


Fig y puerta lógica cmos complementaria usando tecnicas de layout de celdas estandar

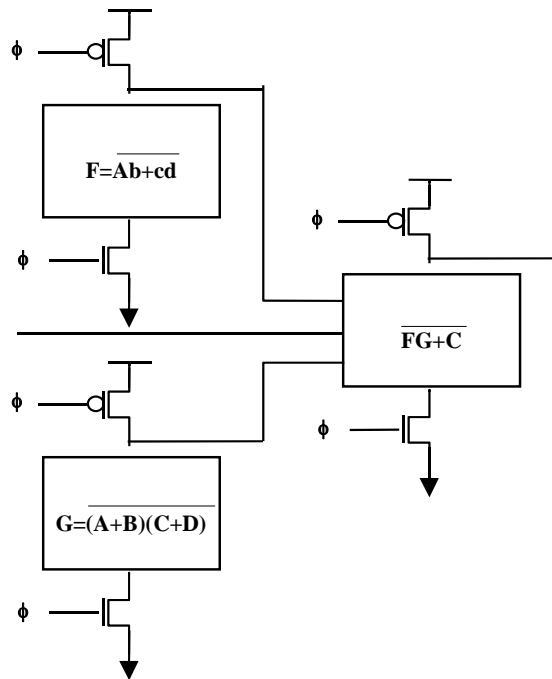
El conjunto de puertas NMOS se puede utilizar para realizar el diseño de precarga y evaluación

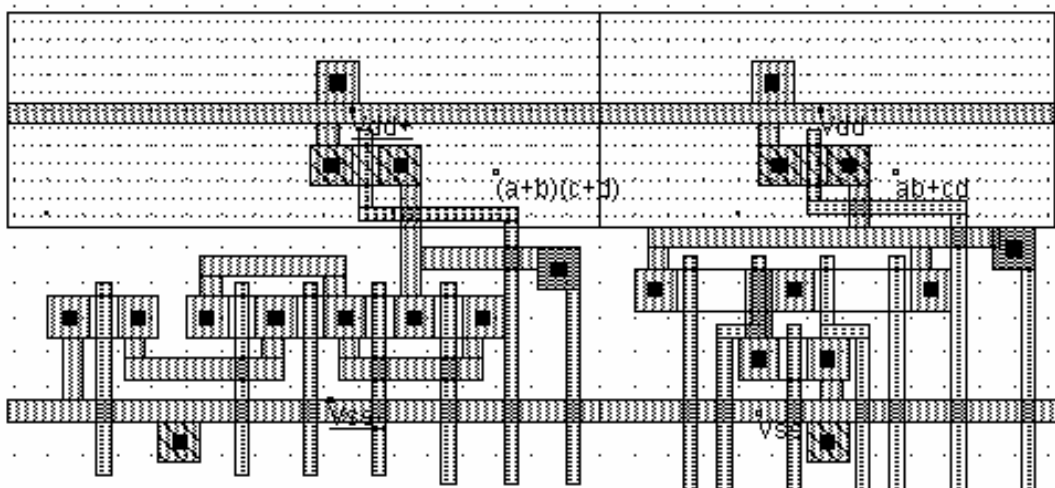
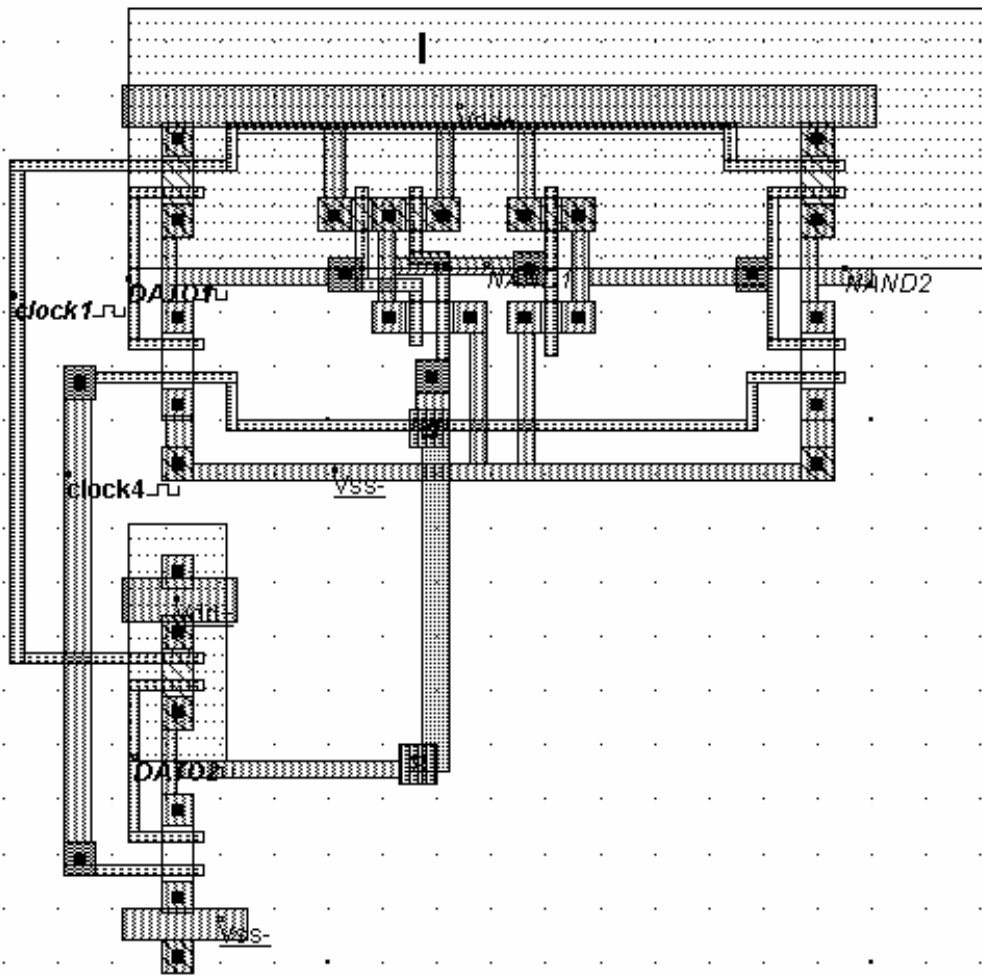


2.2 PUERTAS DINÁMICAS EN CASCADA

Las puertas implementadas mediante lógica de precarga y evaluación no se pueden conectar directamente en cascada debido a que se pueden descargar las capacidades de salida de la segunda etapa debido a retardos en la propagación de la evaluación de la primera etapa.

- A) Implementa el circuito de la figura. Utiliza la técnica de layout de celdas estándar.
- B) Calcula las señales de reloj y de entrada para que se produzca el fenómeno de descarga de la capacidad. Simúlalo para que se vea claramente el problema de la descarga de la capacidad



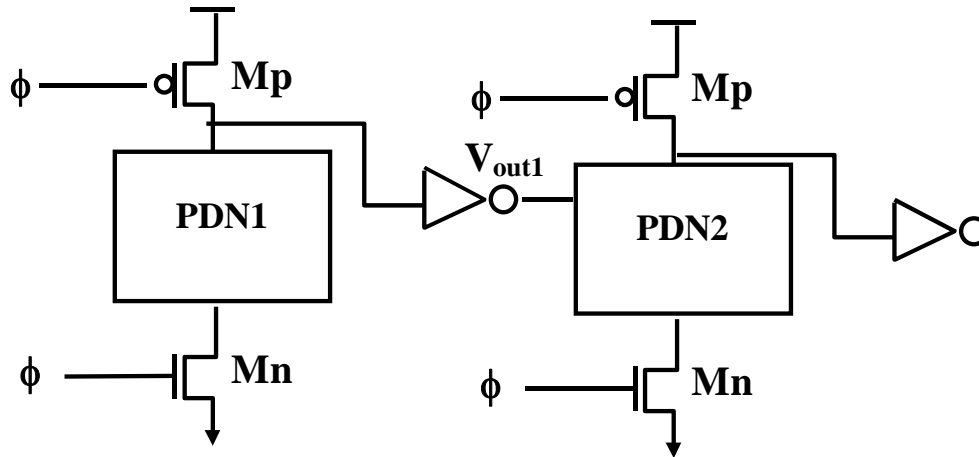


precarga evaluacion en cascada

2.3 LÓGICA DOMINO

La lógica dominó surgió como solución al problema de la descarga de la capacidad cuando se conectaba lógica de precarga y evaluación en cascada.

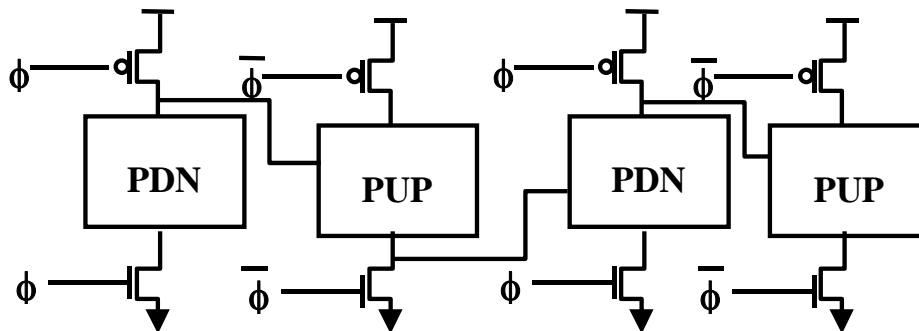
A) Sabiendo que el esquema de la lógica dominó es el de la figura implementala suponiendo que $PDN1 = (a+b)(c+d)$ y $PDN2 = (p-l+r-t)$ nota: En está lógica se debe tener cuidado con los diseños porque al ser puertas que implementan funciones lógicas positivas no es una lógica completa



2.4 LÓGICA ZIPPER

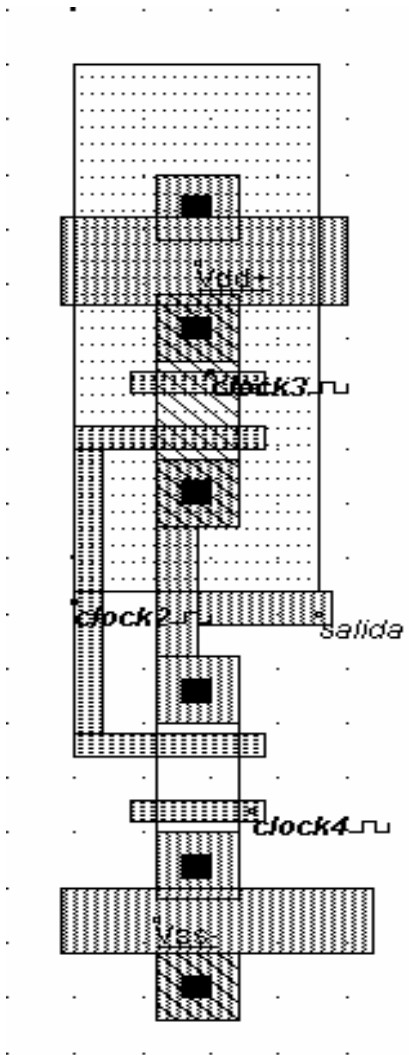
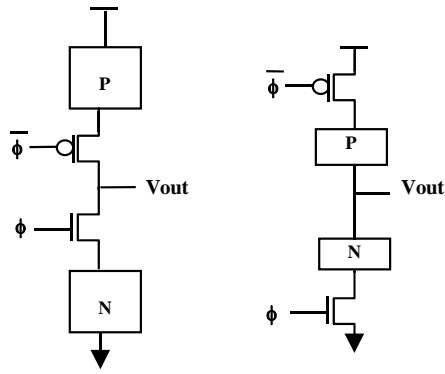
El problema que introducía la lógica dominó era que al tener que añadir inversores a la salida de las puertas estas solo implementaban lógica positiva y esto hacía los diseños complejos. Además se consumía mucho área debido al inversor. Como solución surgió la lógica Zipper cuyo esquema se puede ver en la figura X.

A) Implementa con lógica zipper la misma función implementada para la lógica dominó y compara los resultados.



2.5 C²MOS CLOCKED CMOS

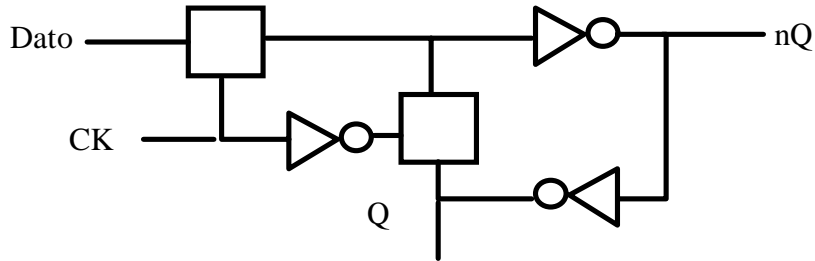
Implementar la función $F=(a+b\cdot c)\cdot d$ con los dos esquemas que aparecen en la figura y comenta los resultados.



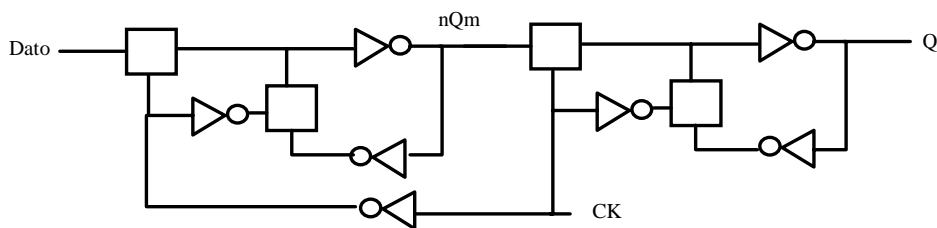
3. LÓGICA SECUENCIAL

3.1 ESTRUCTURAS DE MEMORIA DE UNA FASE DE RELOJ

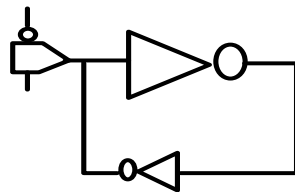
3.1.1 BIESTABLE D



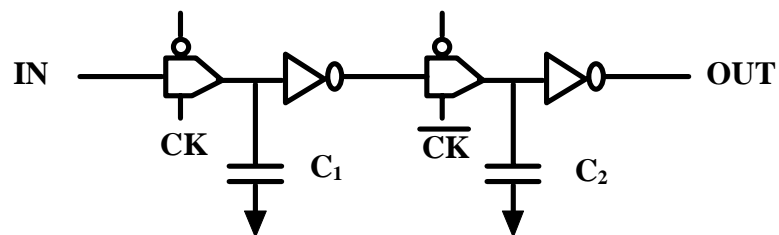
3.1.2 CARGA POR FLANCO



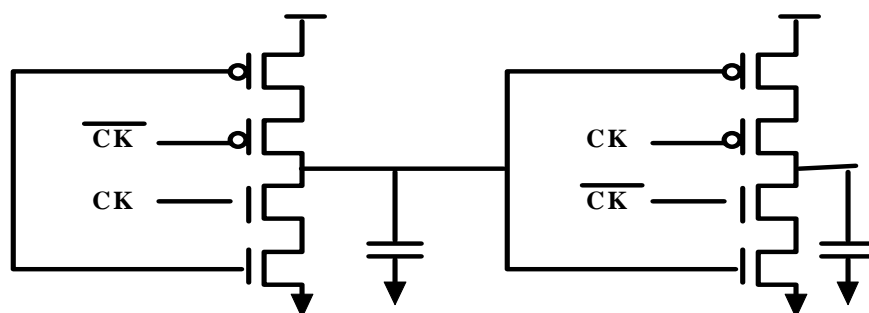
3.1.3 LATCH CON SOLO UNA PUERTA DE TRANSMISIÓN.



3.1.4 REGISTRO DE DESPLAZAMIENTO DINÁMICO

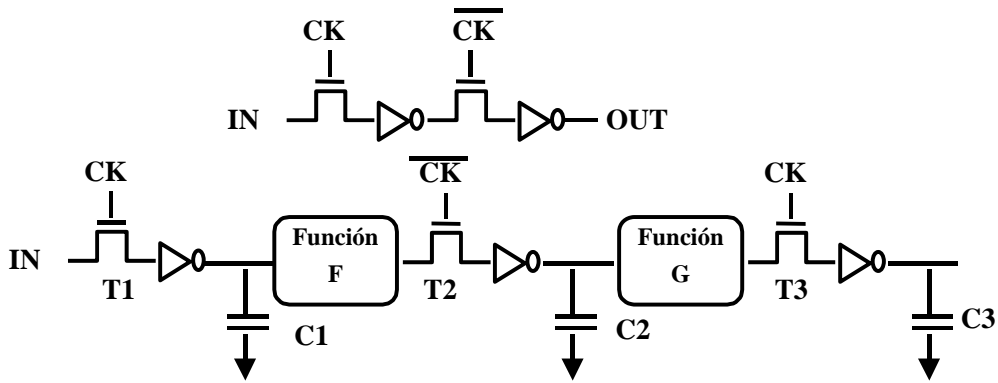


3.1.5 REGISTRO C²MOS



3.2 ESTRUCTURAS LÓGICAS DE UNA FASE DE RELOJ

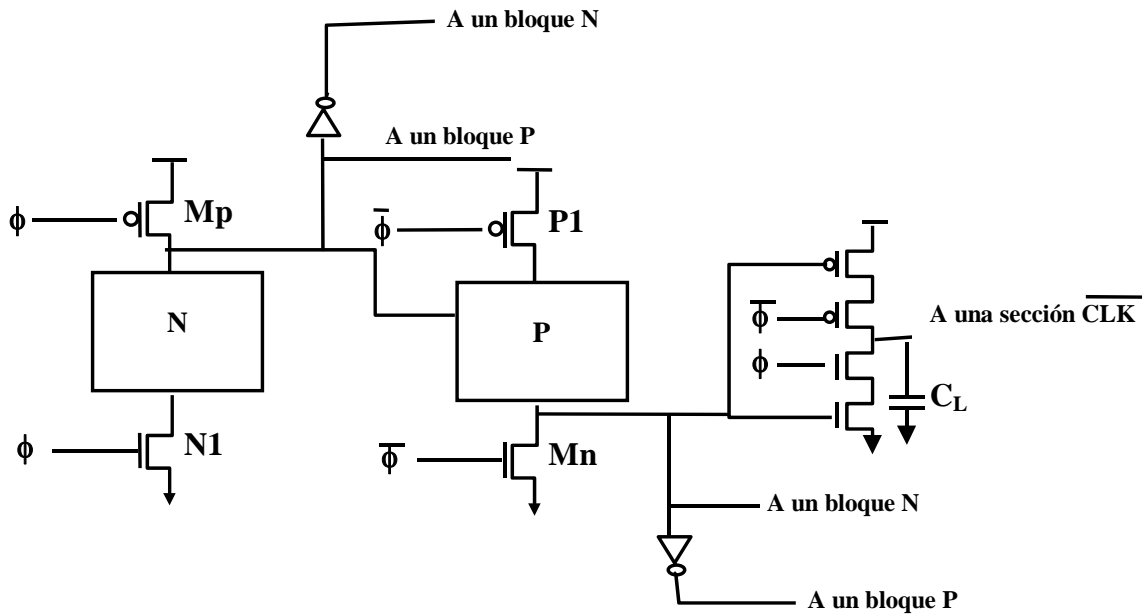
Partiendo de un registro de desplazamiento dinámico se puede realizar la siguiente implementación de un Pipe Line:



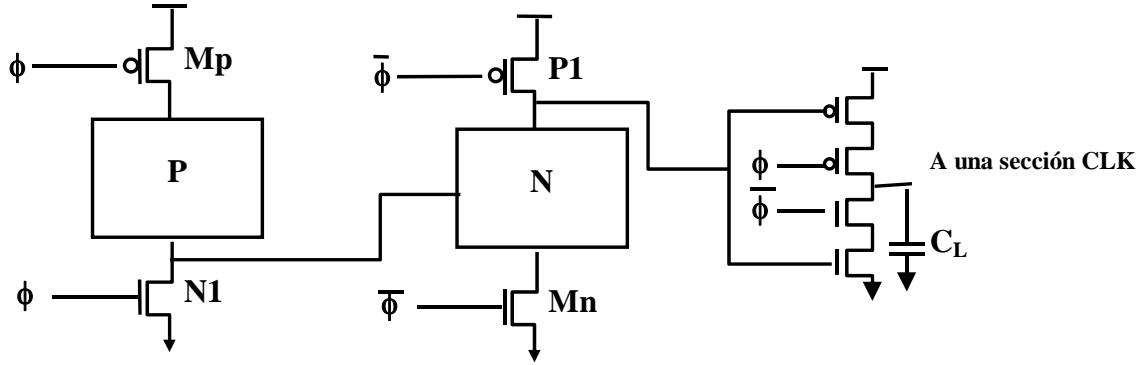
3.3 LÓGICA NORA

sabiendo que

V BLOQUES CK

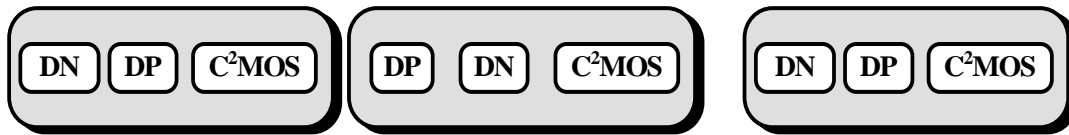


v BLOQUE NOTCLK



Para

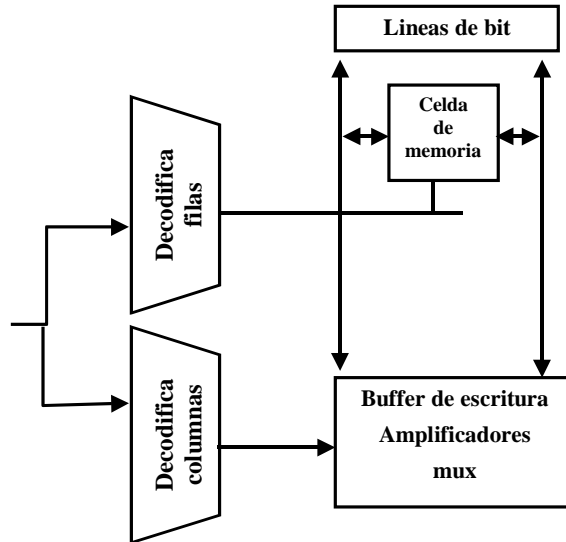
conseguir bloques en cascada se hace de la siguiente manera:



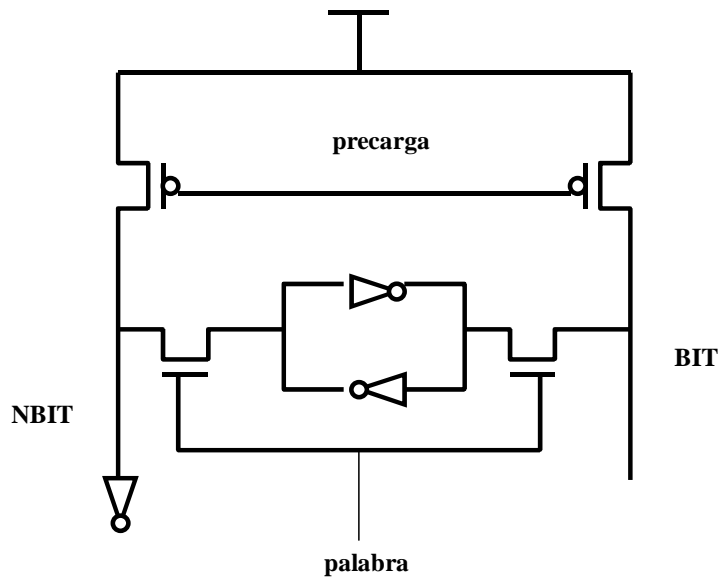
4. SUBSISTEMA DE MEMORIA

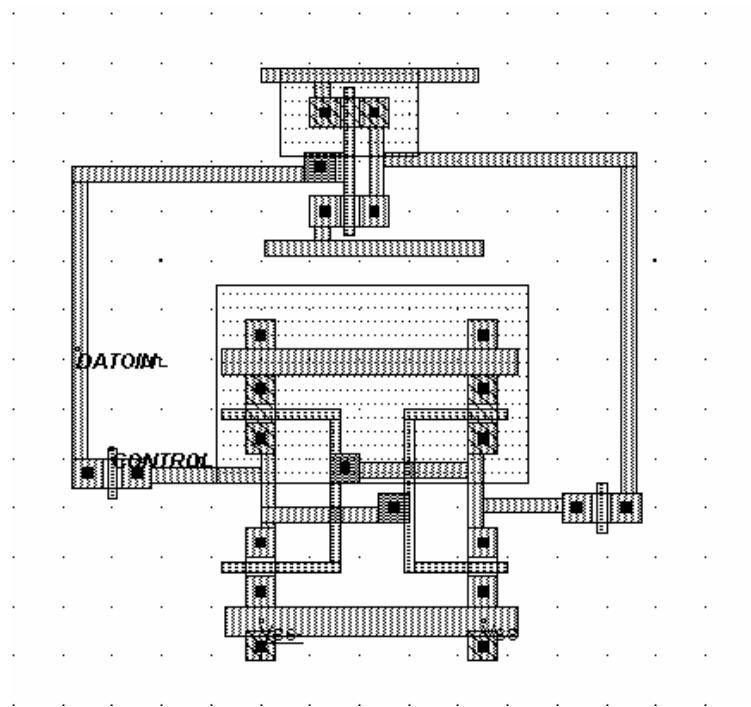
4.1 INTRODUCCIÓN

El esquema general de un bit de memoria aparece en la siguiente figura:

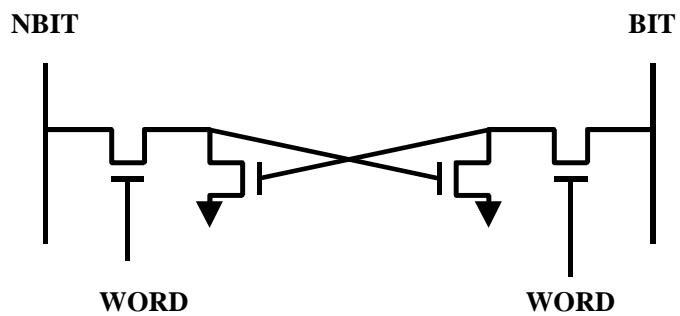


4.2 CELDA DE 6 TRANSISTORES CON PRECARGA





4.3 CELDA DE 4 TRANSISTORES (DINÁMICA)

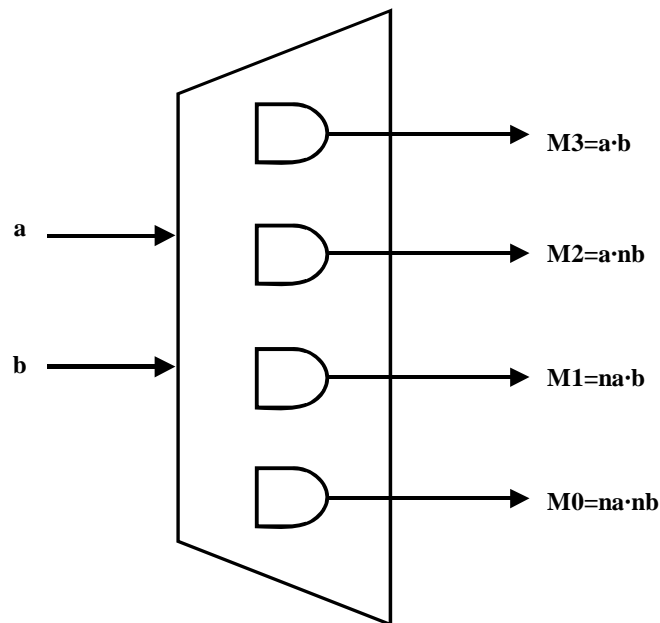


4.4 DECODIFICADORES

4.4.1 DECODIFICADORES DE FILA

Weste pg 574. Me gusta más lo que viene en el weste porque se complementa con lo que se dijo en clase de que se usan decodificadores implementados en lógica pseudo-nmos. También comenta aquí lo de la decodificación de dos niveles (Rabey pg 593)

Un decodificador de N entradas tiene 2^N salidas cada una de las cuales es un mintermino de la entrada, es decir cada línea de selección viene implementada por un puerta nand de tantas entradas como tiene el decodificador.



La primera manera que a uno se le puede ocurrir de implementar esto es colocar una puerta NAND complementaria seguida de un inversor. Pero esta implementación no es demasiado buena cuando se trata de memorias de gran tamaño, en la que es crítica tanto el área consumida como la velocidad de funcionamiento.

La implementación habitual es mediante puertas NOR de lógica pseudo NMOS de tamaño mínimo bufferadas con dos inversores convenientemente escalados.

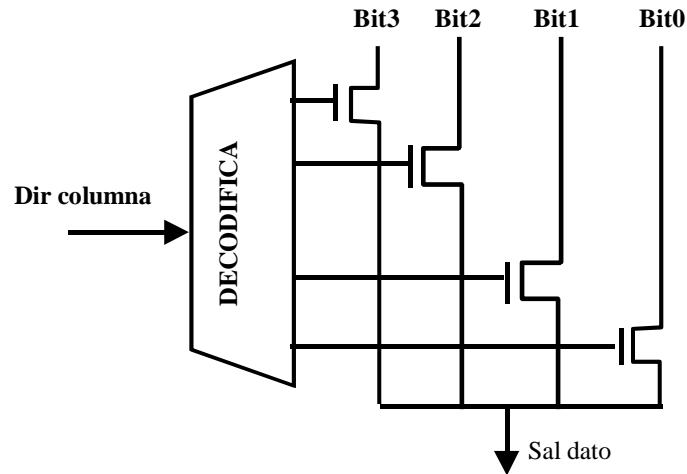
Por otro lado conviene saber que en ocasiones dividir una puerta lógica en dos o más layers lógicos a menudo produce layers más rápidos y económicos. Esto hace surgir el concepto de predecodificación. Se decodifica un segmento de la dirección en el primer layer y el segundo layer produce la selección de la palabra. Esto se puede traducir en la siguiente aproximación. Se sustituye una puerta AND por dos niveles de puertas (NAND+NOR).

4.4.2 DECODIFICADORES DE COLUMNAS

Rabey pg 594

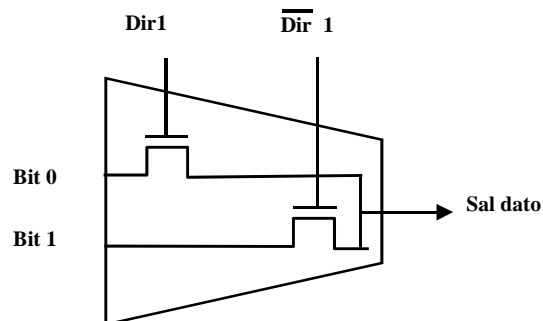
La diferencia con los decodificadores de fila es que estos suelen ser bastante mayores que los de columna. Por eso en este segundo caso se suele utilizar multiplexores para realizar la selección del dato de salida. En la actualidad existen dos tipos de implementación que se ven a continuación.

Implementación 1

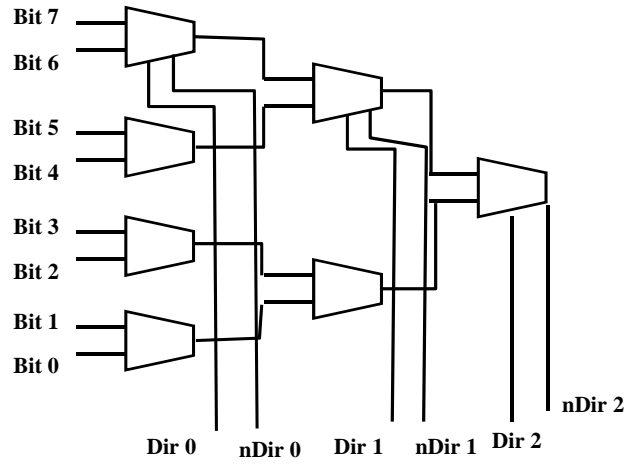


Implementación 2

Sabiendo que un multiplexor se puede implementar de la siguiente manera:



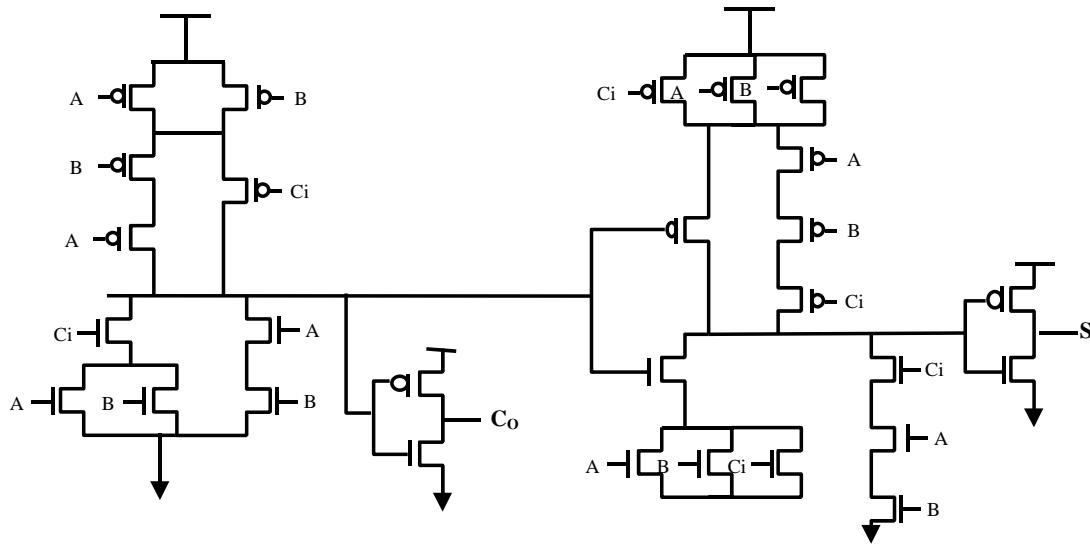
Para implementar un multiplexor que debe seleccionar entre 8 bits se utilizara un diseño multinivel de multiplexores:



Esta aproximación tiene la ventaja del enorme ahorro de área con respecto al primer decodificador, pero es mucho más lento debido a la gran cantidad de transistores que tiene que atravesar, además de la degeneración de la señal que se produce.

5. SUBSISTEMA ARITMETICO

5.1 SUMADOR



Las expresiones típicas del sumador se pueden manipular para conseguir compartir layout entre las dos funciones lógicas. El resultado de esta manipulación se puede ver en las expresiones que aparecen a continuación y en el esquemático de la figura.

$$C_0 = AB + BC_i + AC_i$$

$$S = ABC_i + C_0'(A + B + C_i)$$

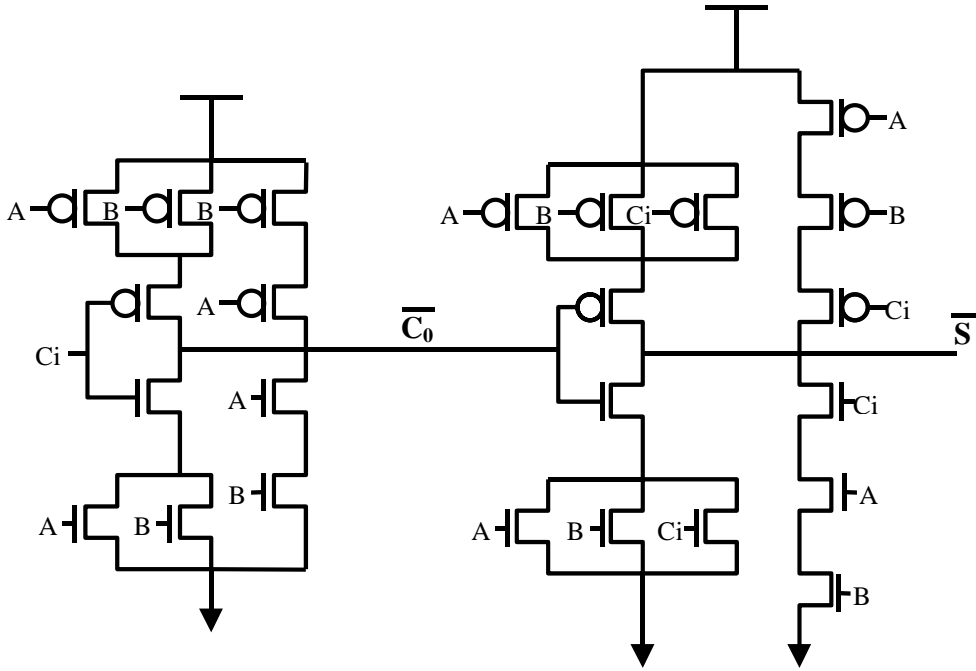
Aunque este sumador consume menos área que la implementación directa de las expresiones de suma tiene importantes inconvenientes como utilizar 28 transistores para su implementación, es decir sigue consumiendo mucho área y potencia. Y además es un circuito muy lento.:

Cadenas de transistores p largas tanto en el carry como en la suma

La señal c_0 tiene una capacidad intrínseca muy grande que consta de dos transistores y seis capacidades de puerta.

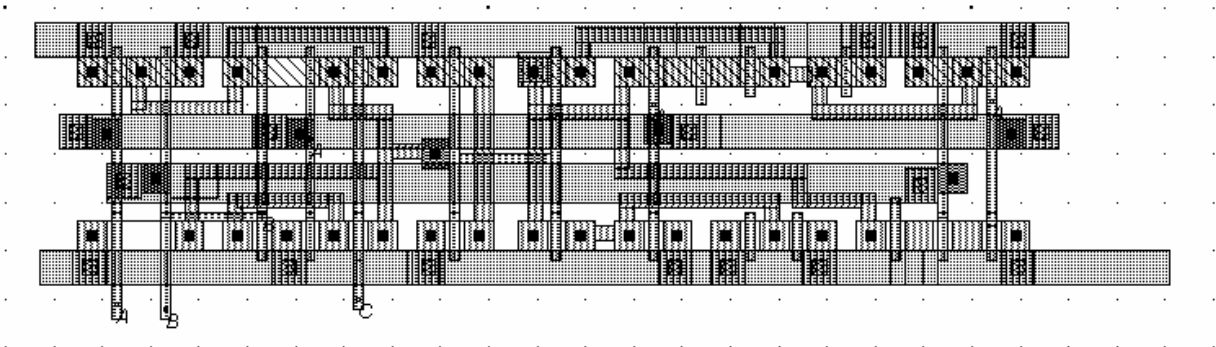
Práctica:

Implementar el sumador de la figura para cuatro bits utilizando los el primer estilo de layout que se explica en el tema 8.



Práctica:

Implementar el sumador de la figura para cuatro bits utilizando los el segundo estilo de layout que se explica en el tema 8.



6. REGLAS DE DISEÑO

Las reglas de diseño son el punto de conexión entre el diseñador de C.I. y el ingeniero de procesos durante la fase de fabricación. El principal objetivo de estas reglas de diseño es obtener un circuito con un rendimiento de producción óptimo (circuitos validos/circuitos no validos) en un área lo menor posible sin comprometer la fiabilidad del circuito. En definitiva representan el mejor compromiso entre la rentabilidad de la fabricación y el performance del circuito. Si las reglas son conservadoras dan lugar a circuitos que cuya rentabilidad de fabricación es mayor pero que son mas lentos y ocupan mayor área. Las reglas mas agresivas tienen mayor probabilidad de generar mejoras en el “performance”, pero estas mejoras pueden dañar la rentabilidad.

Las reglas de diseño especifican al diseñador ligaduras geométricas y topológicas que deben cumplir los patrones utilizados en el proceso de fabricación. Estas ligaduras no son leyes rígidas que se deban cumplir inexorablemente para que los circuitos funcionen correctamente, sino más bien son recomendaciones del fabricante que aseguran una alta probabilidad de una fabricación correcta. Se pueden encontrar diseños que violan las reglas y viceversa. Existen dos conjuntos bien diferenciados de reglas de diseño: las que especifican la anchura mínima de las líneas y las que especifican distancia entre layers. Una anchura demasiado pequeña lleva consigo una discontinuidad en las líneas lo que puede provocar cortocircuitos, y si los layers están demasiado cercanos se pueden fundir o interactuar el uno con el otro produciendo un cortocircuito entre dos nodos de circuitos diferentes

Hay dos aproximaciones para describir las reglas de diseño: las Reglas “micron” que dan las anchuras y distancias entre layers en micras m y son con las que se trabaja en en la industria y las reglas basadas en λ . Con estas normas no se trabaja directamente con la longitud en micras, sino con un parámetro al que se le da un valor determinado según la tecnología con la que estemos trabajando. En teoría permite trabajar con diseños independientemente del avance tecnológico. Es decir 0Un diseño que utilizase reglas λ en su descripción serviría para diferentes tecnologías. Esto fue cierto cuando se trabajaba con tecnologías sobremicra (de 4- a 5 μ .) porque el escalado era lineal pero no es valido para tecnologías inferiores.

6.1.1 REPRESENTACIÓN DE LAYERS

El proceso de fabricación de un dispositivo CMOS es muy complejo y necesita muchos niveles de mascarar. Si el diseño tuviera que describir todas las mascarar, este se haría engorroso y pesado. Generalmente se hace una abstracción del diseño y solo se utilizan las mascarar estrictamente necesarias. Los elementos que se utilizan para describir un diseño CMOS son:

- Polisilicio
- Difusión n
- Difusión p

Metal 1

Metal 2

Pozo

Contacto n

Contacto p

Contacto metal-poli

Contacto metal-metal

La forma de representar estos layers:

Diferentes colores.

Diferentes tipos de bordes.

Diferentes rellenos o fondos.

Rojo-polisilicio

Azul metal

Verde claro difusión P

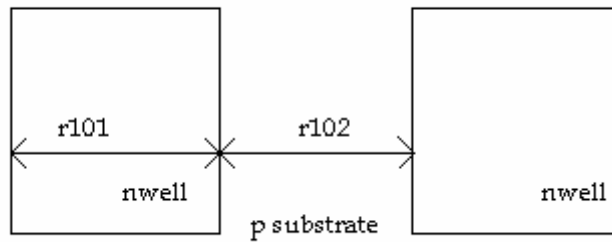
Verde oscuro difusión n

6.1.2 ENUMERACIÓN DE LAS REGLAS

v Pozo N

r101. Tamaño mínimo: 12λ

r102. Espacio entre dos pozos : 12λ



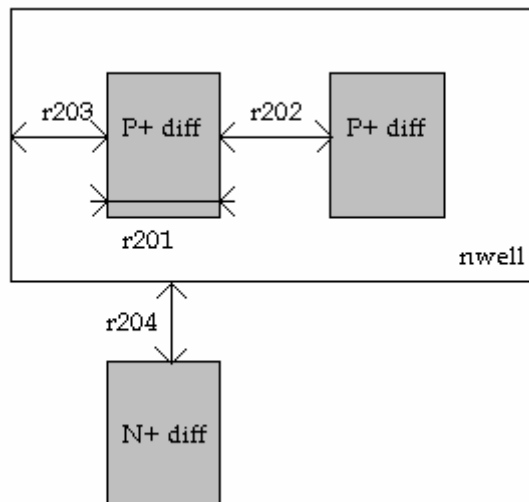
v DIFUSION

r201. Tamaño mínimo 4λ

r202. Espacio mínimo 4λ

r203. Pozo extra tras la difusión: 6λ

r204. Distancia de un pozo y la difusión: 6λ



v Polisilicio.

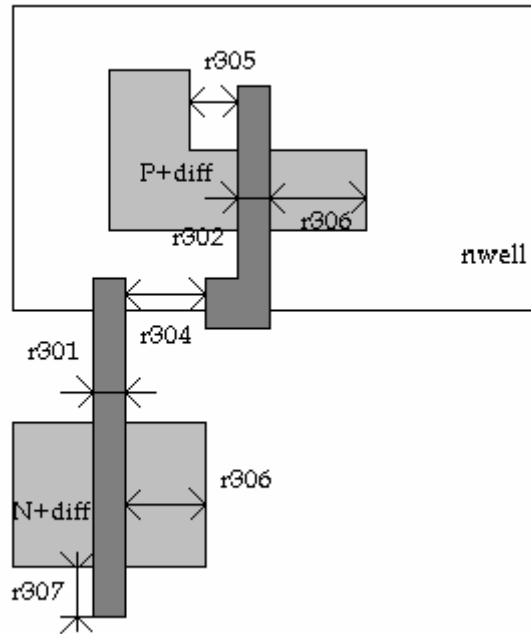
r301, r302, r303. Tamaño mínimo 2λ

r304. Distancia entre dos polisilicios 3λ

R305. Distancia entre poli y difusión 2λ

R306. difusión despues de poli 4λ

R307. Extensión depoli despues de difusión 2λ



v **CONTACTOS.**

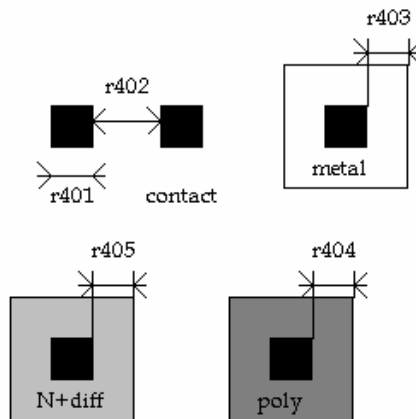
R401: anchura de contacto : 2λ

R404: entre dos contactos: 3λ

R403 extrametal sobre contacto 2λ

R404: extra poli sobre contacto 2λ

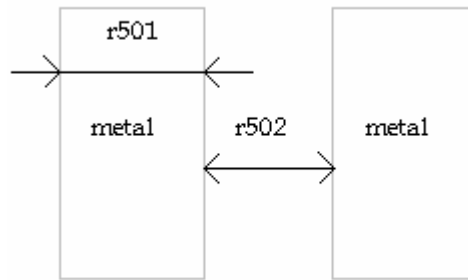
R405 extra diff sobre contacto 2λ



v **METAL 1**

R701: Tamaño minimo 3λ

2. Distancia minima 3λ



VIA ENTRE M1 Y M2

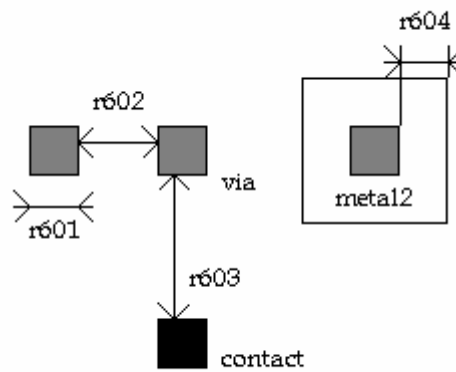
r601: anchura de la via: 3λ

r602 distancia entre dos vias 3λ

r603 distancia entre via y contacto 3λ

r604 extra metal over via: 2λ

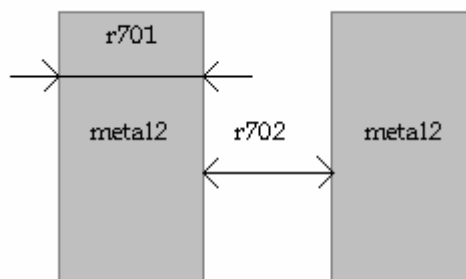
r605: extrametal 2over via



metal2 :

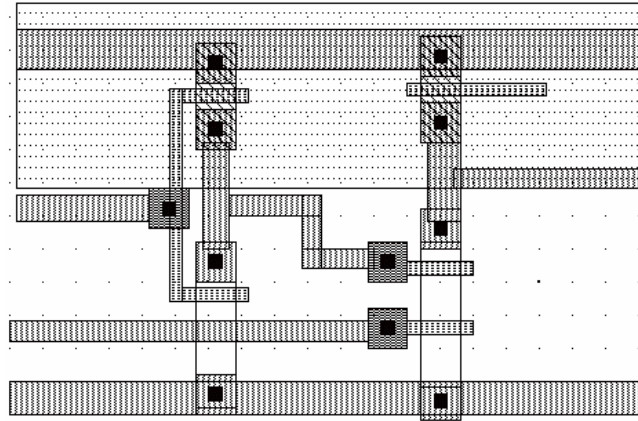
r701 anchura 5λ

r702 distancia entre dos metales $2:5\lambda$

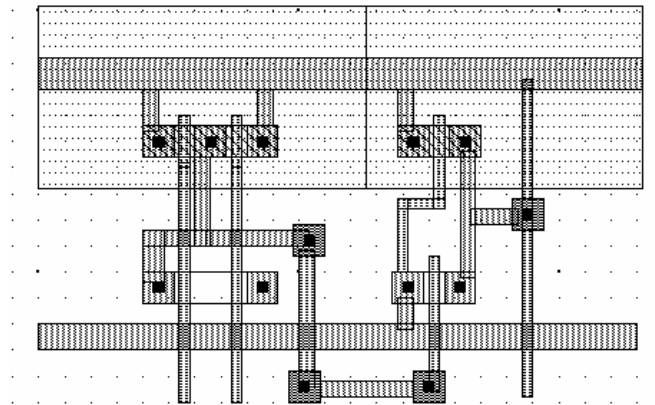


7. TÉCNICAS DE DISEÑO

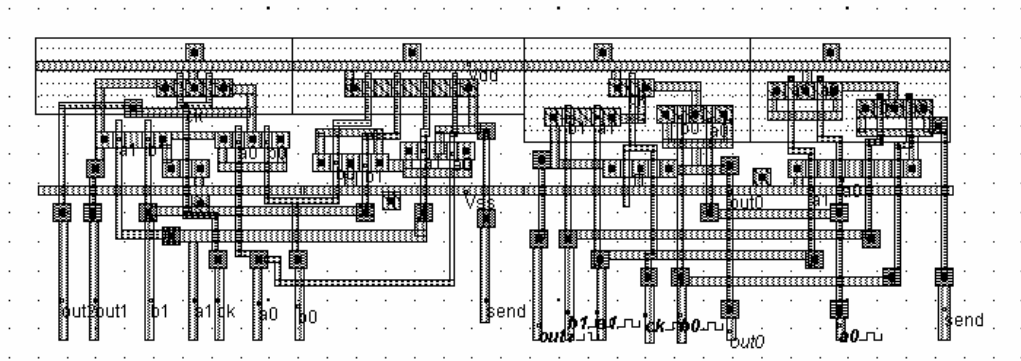
Para puertas complejas existen dos técnicas de layout: la de Weinberger y la de celdas estándar. En la técnica de Weinberger las líneas de entrada y salida de datos se rutan en metal y en paralelo con las líneas de alimentación y en perpendicular a las áreas de difusión. Esta técnica es particularmente efectiva para los caminos de datos bit-sliced.



En la técnica de celdas estándar. Las señales se rutan en polisilicio y perpendiculares a las líneas de alimentación. Esta aproximación da lugar a layout de gran densidad sobre todo en los orientados a lógica combinacional estática. Las interconexiones entre celdas se hacen a través de los llamados canales de rutado. Esta técnica es muy popular debido al alto grado de automatización que se puede obtener.



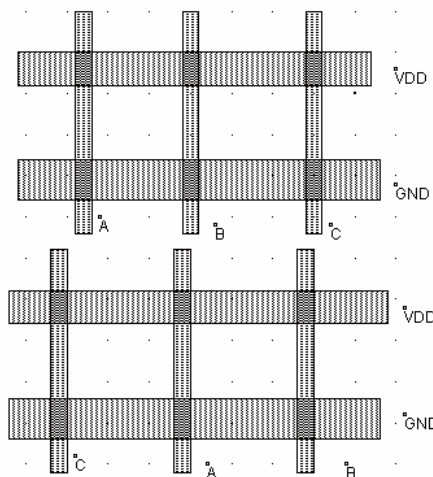
Por razones de densidad conviene que los transistores NMOS y PMOS se implementen como una fila continua de dispositivos con líneas de fuente y alimentación colindantes. Con esta estrategia se necesita una única banda de difusión. Este objetivo no siempre se puede alcanzar, pero se pueden obtener muy buenos resultados si se analizan el orden en que las señales de entrada llegan al layout. Para esto existe un análisis sistemático de las señales de entrada que sirve para encontrar el mejor resultado.

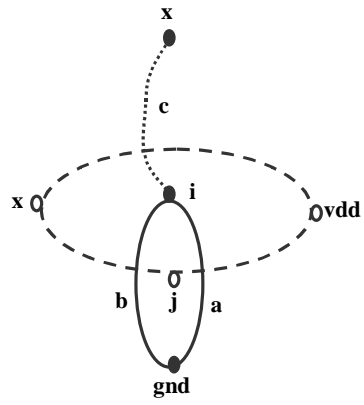
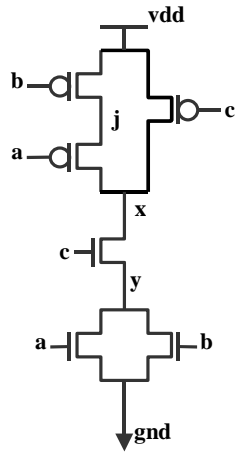


Las librerías de celdas estándar suelen proporcionar un amplio rango de puertas, tanto para diferentes fanin, como fanout. Las puertas más típicas son and/nand, or/nor, exor/xor y flip-flops, aunque también suelen contener celdas más complejas como por ejemplo and-or-inversor, mux, sumadores, contadores, decodificadores. Puesto que tanto el fan-out de las puertas como la capacidad de carga debido al hilo son desconocidas, para asegurar a las puertas son capaces de cargarlas correctamente en tiempos aceptables se suelen emplear transistores de salida muy grandes. Esto simplifica el diseño pero va en detrimento del área y la potencia. La opción alternativa es proporcionar para cada puerta lógica diferentes celdas encargadas de cargar diferentes capacidades.

Técnica del grafo lógico. El grafo lógico de una red de transistores es aquel en el que los vértices corresponden a los nodos del circuito (siendo los nodos las uniones entre transistores y la alimentación y la tierra). Las aristas del grafo son los transistores que unen los nodos. A estas aristas se les da el nombre de la señal que lo controla. El grafo lógico puede ser del árbol NMOS o del árbol PMOS.

Se llama camino de Euler al camino que recorre todas las aristas del grafo sin repetir ninguna de ellas. El orden en que se recorren las aristas es el orden en que deben llegar las entradas. Como existe un grafo para el árbol PMOS y otro para el árbol NMOS, puede que se obtengan dos ordenes de entradas diferentes. Cuando el orden de las entradas es el mismo para los dos grafos se dice que los caminos de Euler son consistentes. Este es el caso óptimo para optimizar el layout. En la figura que viene a continuación se ve como son los grafos lógicos para puerta de la izquierda. Se ve con claridad que {a,b,c} es un camino de datos consistente que fija el mejor orden de llegada de las señales al layout





8. DEL ESQUEMATICO DEL CAMINO DE DATOS AL AYOUT

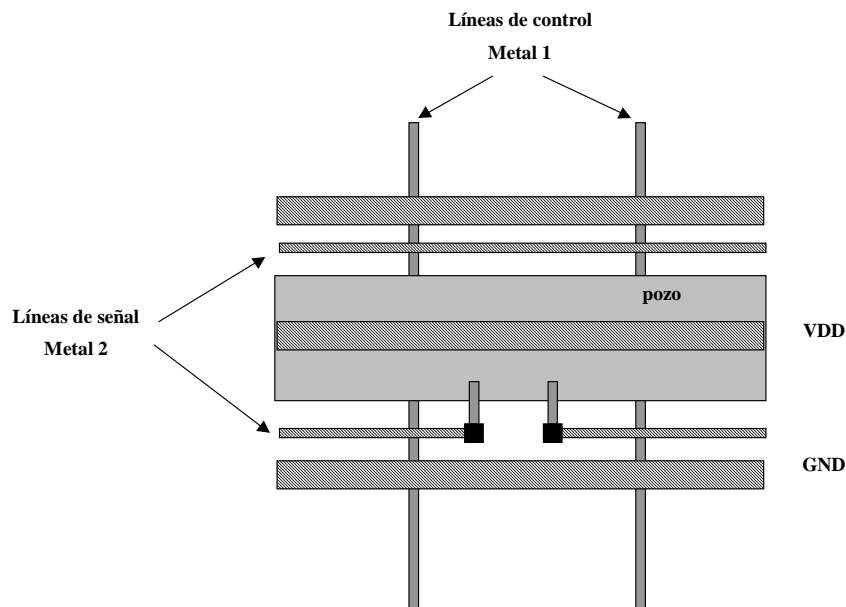
Las necesidades de rendimiento afectan a la manera final en que la estructura del circuito se vuelca sobre el silicio.

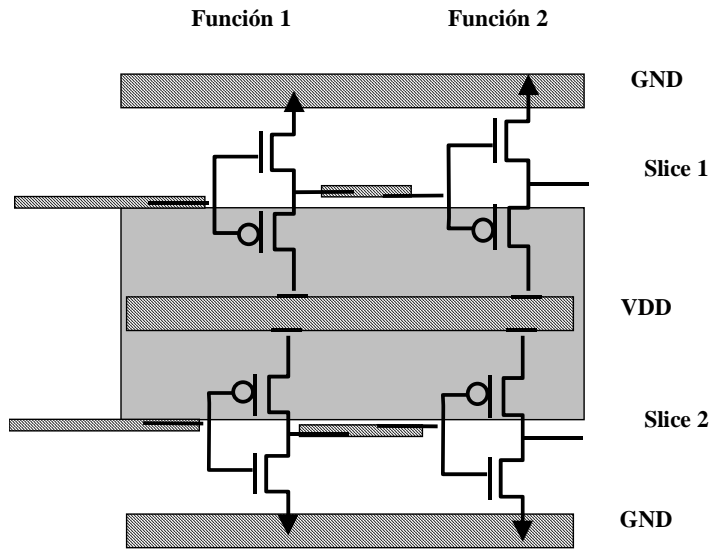
Hay que tener gran cuidado en minimizar las capacidades parásitas de los caminos críticos. El primer objetivo es conservar la estructura del diseño minimizando la longitud de los hilos. Esto explica el porque el diseño del camino de datos todavía es manual, sobre todo en microprocesadores de alto rendimiento, mientras el diseño de otras unidades como la de control es automática.

Afortunadamente la regularidad de las estructuras del camino de datos simplifica la tarea del diseñador, aplicando técnicas de bit-slice.

En un proceso de dos líneas de metal, existen un cierto número de estrategias para implementar los módulos bit-slice. Las topología de básica y la elección de los layer de rutado semuestran en la figura

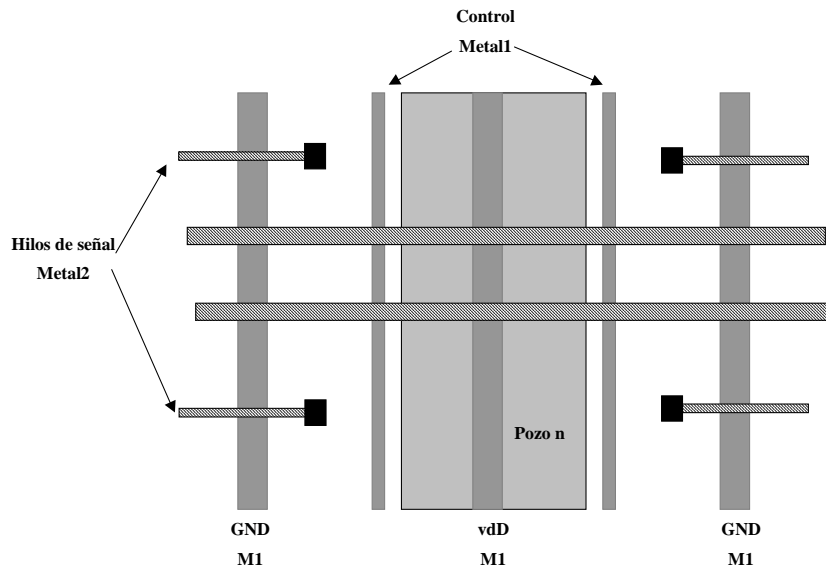
En la primera aproximación, los pozos se orientan horizontalmente y se comparten entre slices vecinos.. La línea de alimentación Vdd también se comparte entre slice vecinos. Esto hace colocar los slices pares e impares tomando como eje la línea de alimentación

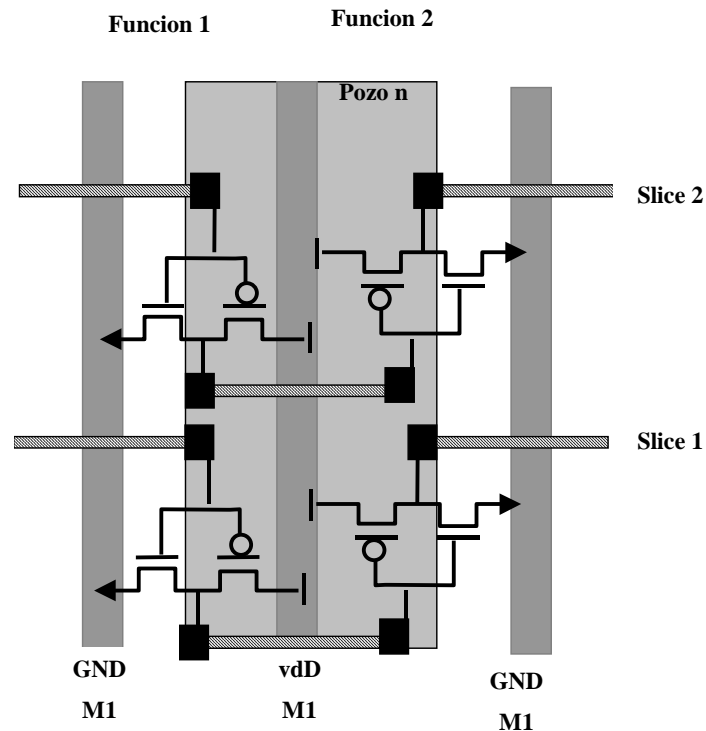




Slice 1 y slice 2 vecinos

En la segunda estrategia, el pozo y las líneas de alimentación corren de arriba abajo y pueden compartirse entre celdas vecinas del mismo slace, por ejemplo entre una celda sumadora y una multiplicadora. Esto implica que estas celdas puedan conectarse "by abutment" lo que excluye la inserción de canales de rutado verticales. Esta aproximación es mejor para el diseño manual.





Algunas características de este tipo de layout se pueden ver a continuación:

Las líneas globales de alimentación se rutan en la parte superior e inferior del diseño en metal 2 (horizontalmente) mientras que se distribuyen en vertical en metal 1

Los canales de rutado verticales se proporcionan para capacitar las conexiones entre celdas vecinas, esto impide que estas celdas compartan pozo. De hecho el layaout clásico explicado solo se encuentra en la actualidad en celdas de tipo sumador donde la profundidad de la lógica requiere puertas en cascada. En estos caso las líneas de alimentación y los pozos si se comparten entre celdas vecinas.

La aproximación de datapath bite-slice usa un paradigma de ubicación lineal. Esto es todas las celdas son ubicadas en un único eje. Debido a esto algunas conexiones deben atravesar entre celdas no vecinas. Para intentar que no crezca en exceso la longitud del hilo y por lo tanto su capacidad asociada, feedthroughs son usadas . Estas son areas entre celdas no usadas