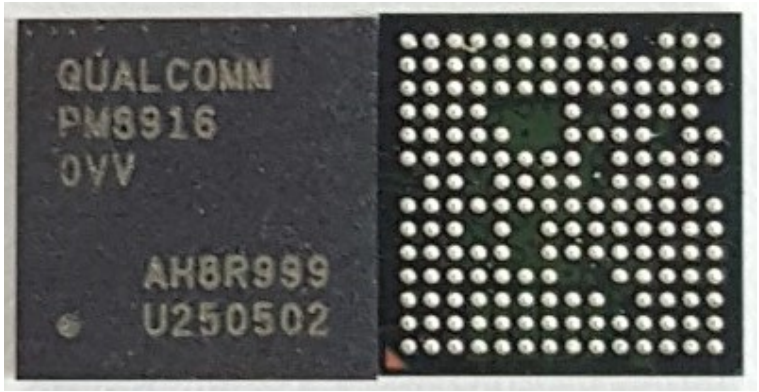


Subsistema de Memoria

Alejandro Furfaro

12 de octubre de 2020

Temario

Temario

It's the Memory, Stupid!

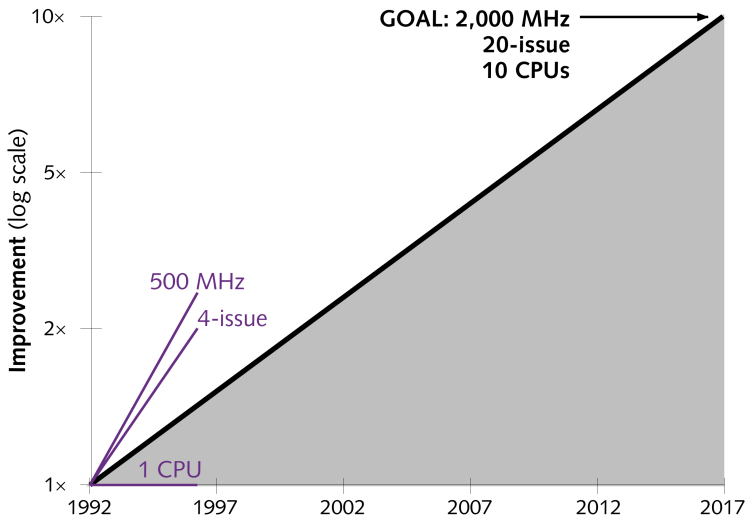
It's the Memory, Stupid!

Richard Sites, arquitecto senior de computadores en Digital Equipment Corporation (DEC), uno de los líderes del desarrollo del procesador Alfa, publicó en 1996 un reporte con este título provocador, en el prestigioso "Microprocessor Report ".

It's the Memory, Stupid!

Richard Sites, arquitecto senior de computadores en Digital Equipment Corporation (DEC), uno de los líderes del desarrollo del procesador Alfa, publicó en 1996 un reporte con este título provocador, en el prestigioso "Microprocessor Report ".

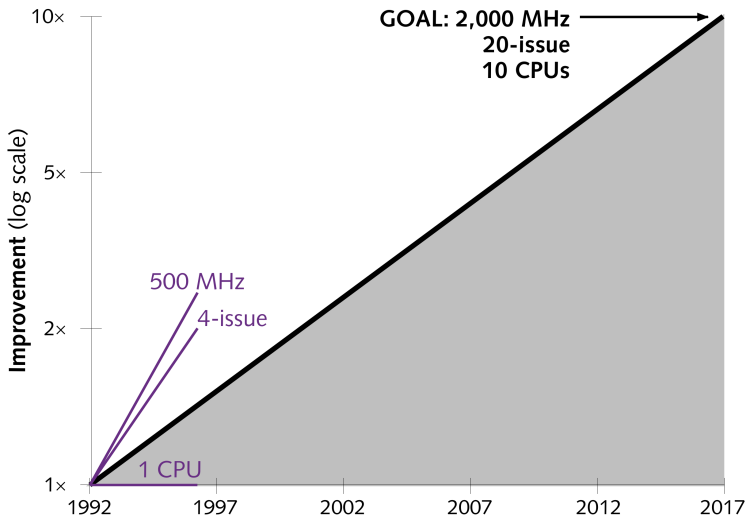
- Explica la estimación de su equipo en DEC al inicio del proyecto Alfa



It's the Memory, Stupid!

Richard Sites, arquitecto senior de computadores en Digital Equipment Corporation (DEC), uno de los líderes del desarrollo del procesador Alfa, publicó en 1996 un reporte con este título provocador, en el prestigioso "Microprocessor Report ".

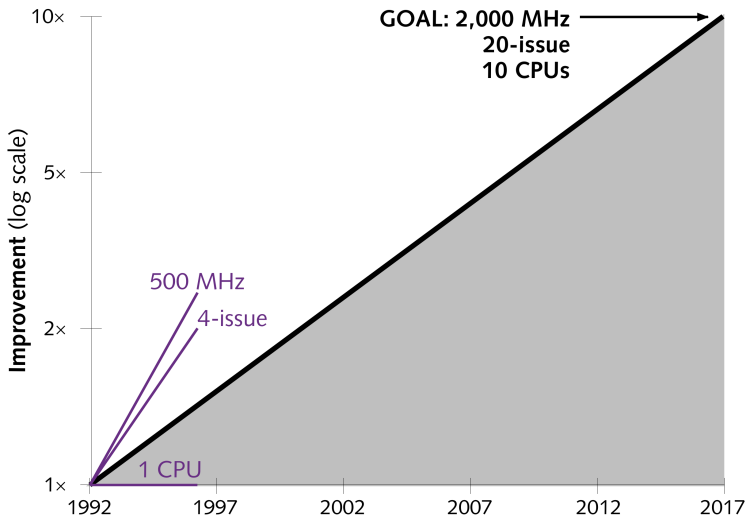
- Explica la estimación de su equipo en DEC al inicio del proyecto Alfa
- Prospección a 25 años (es decir a 2017): performance x1000.



It's the Memory, Stupid!

Richard Sites, arquitecto senior de computadores en Digital Equipment Corporation (DEC), uno de los líderes del desarrollo del procesador Alfa, publicó en 1996 un reporte con este título provocador, en el prestigioso "Microprocessor Report ".

- Explica la estimación de su equipo en DEC al inicio del proyecto Alfa
- Prospección a 25 años (es decir a 2017): performance x1000.
- Frecuencia de clock x10, # de instrucciones enviadas a ejecución por ciclo de clock x10, y # de cores x10



It's the Memory, Stupid!

- En 1992 se lanza el procesador Alfa 21064, con una CPU@200Mhz, y superescalar de dos vías (envía dos instrucciones a ejecutar por ciclo de clock).

It's the Memory, Stupid!

- En 1992 se lanza el procesador Alfa 21064, con una CPU@200Mhz, y superescalar de dos vías (envía dos instrucciones a ejecutar por ciclo de clock).
- Según la prospección, estas tres magnitudes cuatro años después deberían estar en promedio en x1.45.

It's the Memory, Stupid!

- En 1992 se lanza el procesador Alfa 21064, con una CPU@200Mhz, y superescalar de dos vías (envía dos instrucciones a ejecutar por ciclo de clock).
- Según la prospección, estas tres magnitudes cuatro años después deberían estar en promedio en x1.45.
- En 1996 año del artículo, Sites mostró que los pronósticos de estas tres magnitudes fueron conservadores.

It's the Memory, Stupid!

- En 1992 se lanza el procesador Alfa 21064, con una CPU@200Mhz, y superescalar de dos vías (envía dos instrucciones a ejecutar por ciclo de clock).
- Según la prospección, estas tres magnitudes cuatro años después deberían estar en promedio en x1.45.
- En 1996 año del artículo, Sites mostró que los pronósticos de estas tres magnitudes fueron conservadores.
- El Alfa 21164 tenía 1 CPU@500Mhz y enviaba 4 instrucciones a ejecución con su superescalar de 4 vías.

It's the Memory, Stupid!

- En 1992 se lanza el procesador Alfa 21064, con una CPU@200Mhz, y superescalar de dos vías (envía dos instrucciones a ejecutar por ciclo de clock).
- Según la prospección, estas tres magnitudes cuatro años después deberían estar en promedio en x1.45.
- En 1996 año del artículo, Sites mostró que los pronósticos de estas tres magnitudes fueron conservadores.
- El Alfa 21164 tenía 1 CPU@500Mhz y enviaba 4 instrucciones a ejecución con su superescalar de 4 vías.
- La frecuencia incrementó x2.5, la cantidad de instrucciones enviadas x2 y solo la cantidad de CPUs quedó atrás.

It's the Memory, Stupid!

- En 1992 se lanza el procesador Alfa 21064, con una CPU@200Mhz, y superescalar de dos vías (envía dos instrucciones a ejecutar por ciclo de clock).
- Según la prospección, estas tres magnitudes cuatro años después deberían estar en promedio en x1.45.
- En 1996 año del artículo, Sites mostró que los pronósticos de estas tres magnitudes fueron conservadores.
- El Alfa 21164 tenía 1 CPU@500Mhz y enviaba 4 instrucciones a ejecución con su superescalar de 4 vías.
- La frecuencia incrementó x2.5, la cantidad de instrucciones enviadas x2 y solo la cantidad de CPUs quedó atrás.
- Si miramos la prospección a 2017, de las tres magnitudes por separado el pronóstico tenía mucho sentido.

Perdón...¿Y la performance?

Perdón...¿Y la performance?

- Esa es otra historia...

Perdón...¿Y la performance?

- Esa es otra historia...
- Un benchmark con bases de datos utilizando TPC-C (Transaction Processing Performance Council.), ejecutados en un procesador Alfa 21164@400Mhz, y un Pentium-Pro@200Mhz, indicó que entregaban un resultado cada 4.2, y 4.5 ciclos de clock respectivamente.

Perdón...¿Y la performance?

- Esa es otra historia...
- Un benchmark con bases de datos utilizando TPC-C (Transaction Processing Performance Council.), ejecutados en un procesador Alfa 21164@400Mhz, y un Pentium-Pro@200Mhz, indicó que entregaban un resultado cada 4.2, y 4.5 ciclos de clock respectivamente.
- Dicho de otro modo, están tres ciclos de clock esperando a la memoria.

¿Conclusión?

¿Conclusión?

It's the Memory, Stupid!

Temario

Memorias No volátiles

Memorias No volátiles

- Capaces de retener la información almacenada cuando se les desconecta la alimentación.

Memorias No volátiles

- Capaces de retener la información almacenada cuando se les desconecta la alimentación.

- Han evolucionado tecnológicamente pasando por múltiples modelos, partiendo de la grabación en fábrica, paulatinamente permitieron su modificación offline, y actualmente, son modificables en tiempo real.

Memorias No volátiles

- Las viejas memorias denominadas ***ROM*** (por Read Only Memory), son las primeras implementaciones. Debían ser grabadas por el fabricante del chip, y no eran modificables.

Memorias No volátiles

- Las viejas memorias denominadas ***ROM*** (por Read Only Memory), son las primeras implementaciones. Debían ser grabadas por el fabricante del chip, y no eran modificables.

- Componentes programables solo una vez (***OTPROM***), programables y borrables con luz ultravioleta de una determinada longitud de onda (***EPROM***), etc.

Memorias No volátiles

- Las viejas memorias denominadas ***ROM*** (por Read Only Memory), son las primeras implementaciones. Debían ser grabadas por el fabricante del chip, y no eran modificables.

- Componentes programables solo una vez (***OTPROM***), programables y borrables con luz ultravioleta de una determinada longitud de onda (***EPROM***), etc.

- Las actuales memorias No volátiles denominadas flash por su tecnología de origen, pueden ser grabadas “on the fly”por algoritmos de escritura. Su ejemplo mas habitual son los discos de estado sólido de los equipos portátiles modernos, o las tarjetas SD en los sistemas embedded.

Volátiles

Volátiles

- Conocidas como ***RAM*** (Random Access Memory), se caracterizan por que una vez interrumpida la alimentación eléctrica, la información que almacenaban se pierde.

Volátiles

- Conocidas como ***RAM*** (Random Access Memory), se caracterizan por que una vez interrumpida la alimentación eléctrica, la información que almacenaban se pierde.

- Sin embargo estas memorias pueden almacenar mayores cantidades de información y modificarla en tiempo real a gran velocidad en comparación con las No Volátiles.

Volátiles

- Conocidas como ***RAM*** (Random Access Memory), se caracterizan por que una vez interrumpida la alimentación eléctrica, la información que almacenaban se pierde.

- Sin embargo estas memorias pueden almacenar mayores cantidades de información y modificarla en tiempo real a gran velocidad en comparación con las No Volátiles.

- Se clasifican de acuerdo con la tecnología y su diseño interno en dinámicas (***DRAM***)y estáticas (***SRAM***).

Uso en un computador

Uso en un computador

- La memoria no volátil se usa fundamentalmente para almacenar el programa de arranque de cualquier sistema. Le seguimos diciendo ***ROM***, mas por costumbre que por corrección técnica.

Uso en un computador

- La memoria no volátil se usa fundamentalmente para almacenar el programa de arranque de cualquier sistema. Le seguimos diciendo ***ROM***, mas por costumbre que por corrección técnica.
- Se conectan en un espacio de direcciones determinado por el propio microprocesador de acuerdo a la dirección en la que éste irá a buscar la primer instrucción luego de encender el equipo.

Uso en un computador

- La memoria no volátil se usa fundamentalmente para almacenar el programa de arranque de cualquier sistema. Le seguimos diciendo ***ROM***, mas por costumbre que por corrección técnica.
- Se conectan en un espacio de direcciones determinado por el propio microprocesador de acuerdo a la dirección en la que éste irá a buscar la primer instrucción luego de encender el equipo.
- El resto es ***RAM*** y allí el sistema copia incluso buena parte del código de arranque para que se ejecute mas rápido ya que las memorias volátiles tienen menor tiempo de acceso, (menos ciclos de clock para acceder al contenido de una dirección).

Temario

Memoria: Mas que un componente, un subsistema

Memoria: Mas que un componente, un subsistema

- El sueño: tener un sistema de Memoria de una sola tecnología y sin jerarquía. Diseño simple, costo moderado. Fin.

Memoria: Mas que un componente, un subsistema

- El sueño: tener un sistema de Memoria de una sola tecnología y sin jerarquía. Diseño simple, costo moderado. Fin.
- La realidad: Solo en sistemas triviales basados en Microcontroladores sencillos para actividades de control rudimentarias.

Memoria: Mas que un componente, un subsistema

- El sueño: tener un sistema de Memoria de una sola tecnología y sin jerarquía. Diseño simple, costo moderado. Fin.
- La realidad: Solo en sistemas triviales basados en Microcontroladores sencillos para actividades de control rudimentarias.
- En un sistema de cómputo un poco mas sofisticado no hay alternativas. Necesitamos mirar la memoria como un subsistema.

Memoria: Mas que un componente, un subsistema

- El sueño: tener un sistema de Memoria de una sola tecnología y sin jerarquía. Diseño simple, costo moderado. Fin.
- La realidad: Solo en sistemas triviales basados en Microcontroladores sencillos para actividades de control rudimentarias.
- En un sistema de cómputo un poco mas sofisticado no hay alternativas. Necesitamos mirar la memoria como un subsistema.
- A punto tal que es mas importante el diseño de este subsistema que el de la propia CPU.

Memoria: Mas que un componente, un subsistema

- El sueño: tener un sistema de Memoria de una sola tecnología y sin jerarquía. Diseño simple, costo moderado. Fin.
- La realidad: Solo en sistemas triviales basados en Microcontroladores sencillos para actividades de control rudimentarias.
- En un sistema de cómputo un poco mas sofisticado no hay alternativas. Necesitamos mirar la memoria como un subsistema.
- A punto tal que es mas importante el diseño de este subsistema que el de la propia CPU.
- Con los avances de las últimas dos décadas en materia de miniaturización, la interacción entre las diferentes jerarquías no es trivial y afloraron como parte de la problemática del diseño de memoria problemas relacionados con ésta interacción que hasta este siglo tenían mínima influencia. Ejemplo: integridad de señal entre diferentes tipos de memoria y la CPU por ejemplo.

Requerimientos mutuamente excluyentes

Requerimientos mutuamente excluyentes

- Gran capacidad de almacenamiento

Requerimientos mutuamente excluyentes

- Gran capacidad de almacenamiento
- Tiempo de acceso mínimo

Requerimientos mutuamente excluyentes

- Gran capacidad de almacenamiento
- Tiempo de acceso mínimo
- No volatilidad para recuperar los datos un cuando se apague el equipo

Requerimientos mutuamente excluyentes

- Gran capacidad de almacenamiento
- Tiempo de acceso mínimo
- No volatilidad para recuperar los datos un cuando se apague el equipo
- Y por supuesto no puede faltar en ninguna lista aspiracional: Bajo costo

Requerimientos mutuamente excluyentes

- Gran capacidad de almacenamiento
- Tiempo de acceso mínimo
- No volatilidad para recuperar los datos un cuando se apague el equipo
- Y por supuesto no puede faltar en ninguna lista aspiracional: Bajo costo

Es una frazada demasiado corta!. Imposible de conseguir con una sola tecnología. De esto deriva la necesidad de construir una jerarquía de memoria.

¿Como construir una jerarquía?

¿Como construir una jerarquía?

- Los procesadores mas modernos construidos con tecnología de 14 nm logran leer datos e instrucciones a casi 1.2 Tbps.

¿Como construir una jerarquía?

- Los procesadores mas modernos construidos con tecnología de 14 nm logran leer datos e instrucciones a casi 1.2 Tbps.
- El costo por bit de la memoria de un equipo de 16GiB de capacidad es en todo concepto inferior a los u$s 100.

¿Como construir una jerarquía?

- Los procesadores mas modernos construidos con tecnología de 14 nm logran leer datos e instrucciones a casi 1.2 Tbps.
- El costo por bit de la memoria de un equipo de 16GiB de capacidad es en todo concepto inferior a los u$s 100.
- El principio rector de la jerarquía de memoria se conoce como localidad de referencia.

¿Como construir una jerarquía?

- Los procesadores mas modernos construidos con tecnología de 14 nm logran leer datos e instrucciones a casi 1.2 Tbps.
- El costo por bit de la memoria de un equipo de 16GiB de capacidad es en todo concepto inferior a los u$s 100.
- El principio rector de la jerarquía de memoria se conoce como localidad de referencia.
- Se basa en dos trabajos de investigación de intercambio de bloques de memoria y storage cuando se desarrollaron los algoritmos para manejo de memoria virtual.

¿Como construir una jerarquía?

- Los procesadores mas modernos construidos con tecnología de 14 nm logran leer datos e instrucciones a casi 1.2 Tbps.
- El costo por bit de la memoria de un equipo de 16GiB de capacidad es en todo concepto inferior a los u$s 100.
- El principio rector de la jerarquía de memoria se conoce como localidad de referencia.
- Se basa en dos trabajos de investigación de intercambio de bloques de memoria y storage cuando se desarrollaron los algoritmos para manejo de memoria virtual.
- En 1966 L. A. Belady, publicó en IBM System Journal un estudio muy comprensivo y esclarecedor acerca del funcionamiento real del acceso a memoria de un computador.

¿Como construir una jerarquía?

- Los procesadores mas modernos construidos con tecnología de 14 nm logran leer datos e instrucciones a casi 1.2 Tbps.
- El costo por bit de la memoria de un equipo de 16GiB de capacidad es en todo concepto inferior a los u$s 100.
- El principio rector de la jerarquía de memoria se conoce como localidad de referencia.
- Se basa en dos trabajos de investigación de intercambio de bloques de memoria y storage cuando se desarrollaron los algoritmos para manejo de memoria virtual.
- En 1966 L. A. Belady, publicó en IBM System Journal un estudio muy comprensivo y esclarecedor acerca del funcionamiento real del acceso a memoria de un computador.
- En pocas palabras, sus conclusiones fueron: el acceso a memoria no es tan aleatorio como el “Random Access Memory”lo sugiere, sino que por el contrario es bastante predecible.

Principio de localidad

Principio de localidad

Observación empírica

Si tomamos cualquier programa medianamente bien escrito, que respete nimiamente las buenas prácticas de programación, veremos que los acceso a memoria se restringen a un intervalo continuo de direcciones de memoria durante una cantidad de tiempo significativa.

Principio de localidad

Localidad temporal

Se refiere a un patrón de acceso a las mismas direcciones de memoria en un intervalo temporal finito y acotado.
Las direcciones de memoria que la CPU está utilizando actualmente tienen una probabilidad muy alta de seguir siendo accedidas en los próximos ciclos de memoria.
La probabilidad de iniciar un ciclo de memoria hacia direcciones que no están siendo utilizadas, en cambio, es muy baja.

Principio de localidad

Localidad espacial

Se refiere al espacio de direcciones que utiliza una CPU en un intervalo temporal finito y acotado.
La probabilidad que la CPU utilice direcciones vecinas a la del objeto que está direccionando actualmente es muy alta.
Contrariamente es muy poco probable que utilice direcciones no vecinas.

Ejemplo: Algoritmo de Convolución

```
1     for (i = 0 ; i < 256 ; i++ )
2     {
3         suma = 0.0f;
4         for (j = 0 ; (j <= i && j < 256) ; j++)
5             suma += v0[i-j] * v1[j];
6         fAux[i] = suma;
7     }
```

Ejemplo: Algoritmo de Convolución

```
1     for (i = 0 ; i < 256 ; i++ )
2     {
3         suma = 0.0f;
4         for (j = 0 ; (j <= i && j < 256) ; j++)
5             suma += v0[i-j] * v1[j];
6         fAux[i] = suma;
7     }
```

- Las variables ***i***, ***j***, ***suma***, se utilizan muy a menudo. Son claro ejemplo del principio de vecindad temporal. Las mismas direcciones se emplean con una frecuencia de uso alta.

Ejemplo: Algoritmo de Convolución

```
1     for (i = 0 ; i < 256 ; i++ )
2     {
3         suma = 0.0f;
4         for (j = 0 ; (j <= i && j < 256) ; j++)
5             suma += v0[i-j] * v1[j];
6         fAux[i] = suma;
7     }
```

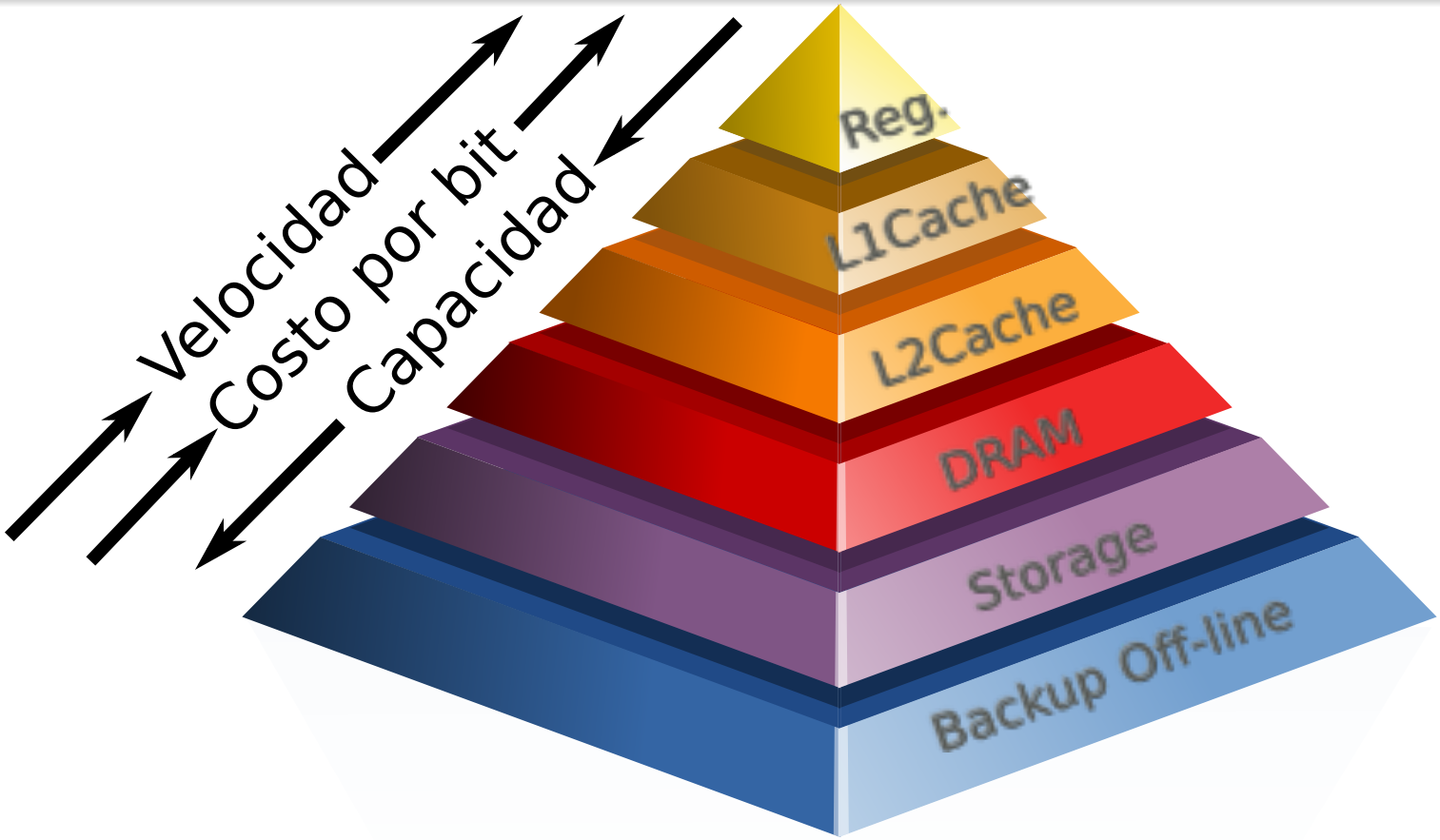
- Las variables ***i***, ***j***, ***suma***, se utilizan muy a menudo. Son claro ejemplo del principio de vecindad temporal. Las mismas direcciones se emplean con una frecuencia de uso alta.
- El espacio de direcciones del código verifica el principio de localidad espacial. La probabilidad de acceso a direcciones de ese espacio es alta mientras estemos dentro de ese loop y no haya conmutación de tareas.

Ejemplo: Algoritmo de Convolución

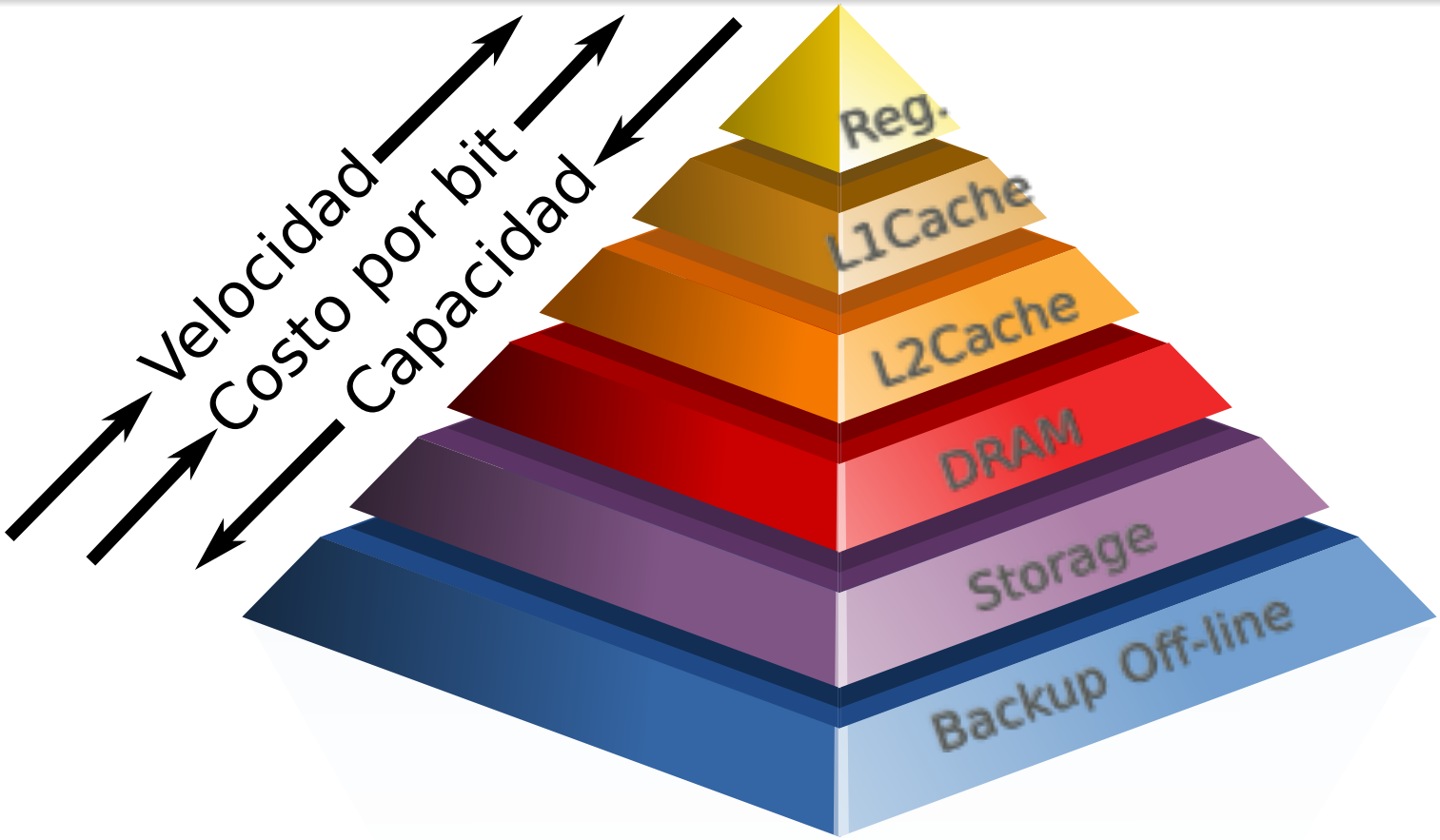
```
1    for (i = 0 ; i < 256 ; i++ )
2    {
3        suma = 0.0f;
4        for (j = 0 ; (j <= i && j < 256) ; j++)
5            suma += v0[i-j] * v1[j];
6        fAux[i] = suma;
7    }
```

- Las variables ***i***, ***j***, ***suma***, se utilizan muy a menudo. Son claro ejemplo del principio de vecindad temporal. Las mismas direcciones se emplean con una frecuencia de uso alta.
- El espacio de direcciones del código verifica el principio de localidad espacial. La probabilidad de acceso a direcciones de ese espacio es alta mientras estemos dentro de ese loop y no haya conmutación de tareas.
- Idea: Si logramos que este código y las variables estén en una memoria rápida cuando el procesador los necesite, su tiempo de acceso será mínimo (óptimo) para ejecutar la convolución.

Memoria: jerarquía piramidal



Memoria: jerarquía piramidal



La jerarquía piramidal plantea que si los datos y el código que el procesador utiliza están (casi) siempre cerca del vértice (es decir, en las memorias mas rápidas), el sistema se comporta de manera óptima.

Temario

Performance

Performance

$$CPI = \frac{TEC}{TIC} \tag{1}$$

CPI = Cycles per Instruction; ***TEC*** = Total execution cycles; ***TIC*** = Total user-level Instructions Committed

Performance

$$CPI = \frac{TEC}{TIC} \tag{1}$$

CPI = Cycles per Instruction; ***TEC*** = Total execution cycles; ***TIC*** = Total user-level Instructions Committed

$$MemSys\_CPI\_Ovrh = RealCPI - TeoricCPI \tag{2}$$

MemSys_CPI_Ovrh = Memory System CPI Overhead; ***Real CPI*** = CPI Real (el que se mide); ***TeoricCPI*** = CPI considerando Memoria Ideal.

Performance

$$CPI = \frac{TEC}{TIC} \tag{1}$$

CPI = Cycles per Instruction; ***TEC*** = Total execution cycles; ***TIC*** = Total user-level Instructions Committed

$$MemSys\_CPI\_Ovrh = RealCPI - TeoricCPI \tag{2}$$

MemSys_CPI_Ovrh = Memory System CPI Overhead; ***Real CPI*** = CPI Real (el que se mide); ***TeoricCPI*** = CPI considerando Memoria Ideal.

$$MCPI = \frac{TMC}{TIC} \tag{3}$$

MCPI = Memory Cycles Per Instruction; ***RealTMC*** = Ciclos de clock totales insumidos por la memoria; ***TIC*** = Total user-level Instructions Committed.

Energía y Potencia

Energía y Potencia

- En Física, Energía consumida es el trabajo efectuado para una tarea.

Energía y Potencia

- En Física, Energía consumida es el trabajo efectuado para una tarea.
- Aplicado a un sistema de cómputo, la Energía consumida es la medida de cuanta carga de batería consume un algoritmo.

Energía y Potencia

- En Física, Energía consumida es el trabajo efectuado para una tarea.
- Aplicado a un sistema de cómputo, la Energía consumida es la medida de cuanta carga de batería consume un algoritmo.
- Sin embargo en general los Microprocesadores y las memorias, especifican Potencia instantánea. (Watts)

Energía y Potencia

- En Física, Energía consumida es el trabajo efectuado para una tarea.
- Aplicado a un sistema de cómputo, la Energía consumida es la medida de cuanta carga de batería consume un algoritmo.
- Sin embargo en general los Microprocesadores y las memorias, especifican Potencia instantánea. (Watts)
- En los dispositivos CMOS se especifica el promedio de los valores de Potencia instantánea en el período de conmutación del transistor CMOS (corte a saturación o visceversa).

Energía y Potencia

- En Física, Energía consumida es el trabajo efectuado para una tarea.
- Aplicado a un sistema de cómputo, la Energía consumida es la medida de cuanta carga de batería consume un algoritmo.
- Sin embargo en general los Microprocesadores y las memorias, especifican Potencia instantánea. (Watts)
- En los dispositivos CMOS se especifica el promedio de los valores de Potencia instantánea en el período de conmutación del transistor CMOS (corte a saturación o visceversa).

$$P_{avg} = (P_{dynamic} + P_{static}) \approxeq C_{tot} V_{dd}^2 f + I_{leak} V_{dd} \quad (4)$$

$\boldsymbol{P_{avg}}$ = Potencia promedio disipada; $\boldsymbol{C_{tot}}$ = Capacidad total de carga del CMOS; $\boldsymbol{V_{dd}}$ = Tensión de alimentación; $\boldsymbol{f}$ = Frecuencia de conmutación (clock); e $\boldsymbol{I_{leak}}$ = Corriente de pérdida (No me digas que te creíste que un CMOS al corte no conduce corriente en la malla de salida... ¡Dale!)

Energía y Potencia

- La corriente de pérdida I_{leak}, es muy baja pero a partir de las tecnologías de 45 a 30 nm dejó de disminuir conforme se miniaturiza un transistor. Otra mala noticia para el tramo final de la Ley de Moore: leakage pasó a ser un “big issue”.

Energía y Potencia

- La corriente de pérdida I_{leak}, es muy baja pero a partir de las tecnologías de 45 a 30 nm dejó de disminuir conforme se miniaturiza un transistor. Otra mala noticia para el tramo final de la Ley de Moore: leakage pasó a ser un “big issue”.
- La Energía está directamente relacionada con la potencia a través del tiempo. De hecho, la potencia media en un intervalo de tiempo es la relación entre la Energía y la duración del intervalo.

Energía y Potencia

- La corriente de pérdida I_{leak}, es muy baja pero a partir de las tecnologías de 45 a 30 nm dejó de disminuir conforme se miniaturiza un transistor. Otra mala noticia para el tramo final de la Ley de Moore: leakage pasó a ser un “big issue”.
- La Energía está directamente relacionada con la potencia a través del tiempo. De hecho, la potencia media en un intervalo de tiempo es la relación entre la Energía y la duración del intervalo.
- En un intervalo T, la cantidad de Joules consumidos es:

Energía y Potencia

- La corriente de pérdida I_{leak}, es muy baja pero a partir de las tecnologías de 45 a 30 nm dejó de disminuir conforme se miniaturiza un transistor. Otra mala noticia para el tramo final de la Ley de Moore: leakage pasó a ser un “big issue”.
- La Energía está directamente relacionada con la potencia a través del tiempo. De hecho, la potencia media en un intervalo de tiempo es la relación entre la Energía y la duración del intervalo.
- En un intervalo T, la cantidad de Joules consumidos es:

$$E = P_{avg}.T \approxeq C_{tot} V_{dd}^2 N + I_{leak} V_{dd}.T \quad (5)$$

N = Cantidad de eventos de switching ocurridos en el intervalo T.

Energía y Potencia

- La corriente de pérdida I_{leak}, es muy baja pero a partir de las tecnologías de 45 a 30 nm dejó de disminuir conforme se miniaturiza un transistor. Otra mala noticia para el tramo final de la Ley de Moore: leakage pasó a ser un “big issue”.
- La Energía está directamente relacionada con la potencia a través del tiempo. De hecho, la potencia media en un intervalo de tiempo es la relación entre la Energía y la duración del intervalo.
- En un intervalo T, la cantidad de Joules consumidos es:

$$E = P_{avg}.T \approxeq C_{tot} V_{dd}^2 N + I_{leak} V_{dd}.T \quad (5)$$

N = Cantidad de eventos de switching ocurridos en el intervalo T.

- Una primer conclusión es que la energía consumida por un algoritmo (la medida del trabajo que realiza el computador) *no depende de la frecuencia de clock*.

Energía y Potencia

- Cuando nos referimos al consumo, lo correcto es hablar de Joules, es decir, Energía. Esto es duración de batería, o KWh en la factura del servicio de electricidad. Sin embargo coloquialmente se suele hablar de energía cuando se miden los Watts instantáneos de un chip. Esto no es correcto.

Energía y Potencia

- Cuando nos referimos al consumo, lo correcto es hablar de Joules, es decir, Energía. Esto es duración de batería, o KWh en la factura del servicio de electricidad. Sin embargo coloquialmente se suele hablar de energía cuando se miden los Watts instantáneos de un chip. Esto no es correcto.
- Típico anuncio de marketing: “Nuestra nueva familia de procesadores “low power”mejora a la línea anterior en un factor de 2”.

Energía y Potencia

- Cuando nos referimos al consumo, lo correcto es hablar de Joules, es decir, Energía. Esto es duración de batería, o KWh en la factura del servicio de electricidad. Sin embargo coloquialmente se suele hablar de energía cuando se miden los Watts instantáneos de un chip. Esto no es correcto.
- Típico anuncio de marketing: “Nuestra nueva familia de procesadores “low power”mejora a la línea anterior en un factor de 2”.
- Bien. Esto puede lograrse simplemente subclockeando a la mitad de frecuencia, reduciendo así a la mitad la potencia disipada.

Energía y Potencia

- Cuando nos referimos al consumo, lo correcto es hablar de Joules, es decir, Energía. Esto es duración de batería, o KWh en la factura del servicio de electricidad. Sin embargo coloquialmente se suele hablar de energía cuando se miden los Watts instantáneos de un chip. Esto no es correcto.
- Típico anuncio de marketing: “Nuestra nueva familia de procesadores “low power”mejora a la línea anterior en un factor de 2”.
- Bien. Esto puede lograrse simplemente subclockeando a la mitad de frecuencia, reduciendo así a la mitad la potencia disipada.
- Sin embargo, la potencia puede perfectamente definirse como la velocidad a la que se consume la energía.

Energía y Potencia

- Cuando nos referimos al consumo, lo correcto es hablar de Joules, es decir, Energía. Esto es duración de batería, o KWh en la factura del servicio de electricidad. Sin embargo coloquialmente se suele hablar de energía cuando se miden los Watts instantáneos de un chip. Esto no es correcto.
- Típico anuncio de marketing: “Nuestra nueva familia de procesadores “low power”mejora a la línea anterior en un factor de 2”.
- Bien. Esto puede lograrse simplemente subclockeando a la mitad de frecuencia, reduciendo así a la mitad la potencia disipada.
- Sin embargo, la potencia puede perfectamente definirse como la velocidad a la que se consume la energía.
- Subclockear solo disminuye la velocidad con que se descarga la batería. El trabajo realizado en Joules para un dado algoritmo no cambia. Tardará el doble en hacerlo.

Energía y Potencia: Métricas de Interés

- Considerando lo dicho en el slide anterior, las métricas que valen la pena y suelen encontrarse son las siguientes:

Energía y Potencia: Métricas de Interés

- Considerando lo dicho en el slide anterior, las métricas que valen la pena y suelen encontrarse son las siguientes:
- **`Energy-Delay Product`**

$$Energy-DelayProduct = EnergiaRequeridaPorLaTarea.TiempoRequeridoPorLaTarea \tag{6}$$

Energía y Potencia: Métricas de Interés

- Considerando lo dicho en el slide anterior, las métricas que valen la pena y suelen encontrarse son las siguientes:
- **Energy-Delay Product**

$$Energy-DelayProduct = EnergiaRequeridaPorLaTarea.TiempoRequeridoPorLaTarea \tag{6}$$

- **Power-Delay Product**

$$Power-DelayProduct = PotenciaConsumidaPorLaTarea.TiempoRequeridoPorLaTarea \tag{7}$$

Energía y Potencia: Métricas de Interés

- Considerando lo dicho en el slide anterior, las métricas que valen la pena y suelen encontrarse son las siguientes:
- **Energy-Delay Product**

$$Energy-DelayProduct = EnergiaRequeridaPorLaTarea.TiempoRequeridoPorLaTarea \tag{6}$$

- **Power-Delay Product**

$$Power-DelayProduct = PotenciaConsumidaPorLaTarea.TiempoRequeridoPorLaTarea \tag{7}$$

- **MIPS per watt**

$$MIPSperWatt = \frac{PerformanceBenchmarckEnMIPS}{PotenciaPromedioDisipadaPorLaTarea} \tag{8}$$

Costos de Tecnología

Costos de Tecnología

- El costo de producir una memoria involucra una serie de factores:

Costos de Tecnología

- El costo de producir una memoria involucra una serie de factores:
- El costo de producir un chip VLSI es proporcional al tamaño del die de Si.

Costos de Tecnología

- El costo de producir una memoria involucra una serie de factores:
- El costo de producir un chip VLSI es proporcional al tamaño del die de Si.
- El packaging también es una componente. El pinout y la tecnología de montaje seleccionados inciden en el costo de la memoria

Costos de Tecnología

- El costo de producir una memoria involucra una serie de factores:
- El costo de producir un chip VLSI es proporcional al tamaño del die de Si.
- El packaging también es una componente. El pinout y la tecnología de montaje seleccionados inciden en el costo de la memoria
- Complejidad de Diseño. Involucra, la cantidad de líneas de código HDL, a mayor cantidad mas complejo es el diseño síntesis y testing. Además requiere mayor cantidad de transistores.

Costos de Tecnología

- El costo de producir una memoria involucra una serie de factores:
- El costo de producir un chip VLSI es proporcional al tamaño del die de Si.
- El packaging también es una componente. El pinout y la tecnología de montaje seleccionados inciden en el costo de la memoria
- Complejidad de Diseño. Involucra, la cantidad de líneas de código HDL, a mayor cantidad mas complejo es el diseño síntesis y testing. Además requiere mayor cantidad de transistores.
- El clock de trabajo claramente incide en la complejidad de diseño de los montajes

Costos de Tecnología

- El costo de producir una memoria involucra una serie de factores:
- El costo de producir un chip VLSI es proporcional al tamaño del die de Si.
- El packaging también es una componente. El pinout y la tecnología de montaje seleccionados inciden en el costo de la memoria
- Complejidad de Diseño. Involucra, la cantidad de líneas de código HDL, a mayor cantidad mas complejo es el diseño síntesis y testing. Además requiere mayor cantidad de transistores.
- El clock de trabajo claramente incide en la complejidad de diseño de los montajes
- Las métricas habituales son: Costo por Bit, por byte, por KioBibyte, etc., Área de Die por bit de almacenamiento (dá una idea de la densidad o capacidad del dispositivo), ancho de banda por pin de encapsulado, entre otros.

Temario

Costos de Tecnología

Costos de Tecnología

- Un sistema jerárquico de memoria, tiene por objetivo que el comportamiento el sistema se aproxime al del componente mas rápido, logrando el costo por bit mas bajo posible, y la mayor eficiencia de energía posible.

Costos de Tecnología

- Un sistema jerárquico de memoria, tiene por objetivo que el comportamiento el sistema se aproxime al del componente mas rápido, logrando el costo por bit mas bajo posible, y la mayor eficiencia de energía posible.
- Independientemente de la escala del sistema bajo análisis, siempre vamos a poder identificar la presencia de diferentes niveles jerárquicos.

Costos de Tecnología

- Un sistema jerárquico de memoria, tiene por objetivo que el comportamiento el sistema se aproxime al del componente mas rápido, logrando el costo por bit mas bajo posible, y la mayor eficiencia de energía posible.
- Independientemente de la escala del sistema bajo análisis, siempre vamos a poder identificar la presencia de diferentes niveles jerárquicos.
- Al menos dos niveles en los microcontroladores mas modestos.

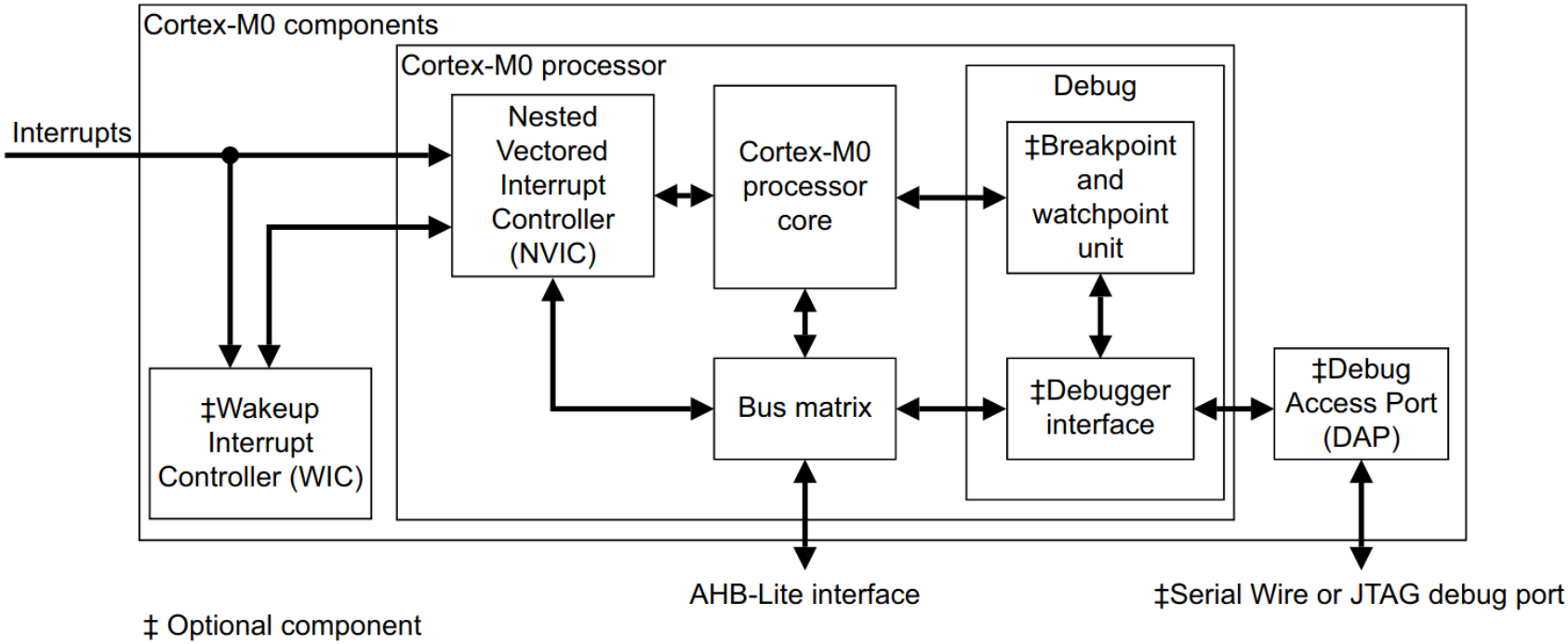
Costos de Tecnología

- Un sistema jerárquico de memoria, tiene por objetivo que el comportamiento el sistema se aproxime al del componente mas rápido, logrando el costo por bit mas bajo posible, y la mayor eficiencia de energía posible.
- Independientemente de la escala del sistema bajo análisis, siempre vamos a poder identificar la presencia de diferentes niveles jerárquicos.
- Al menos dos niveles en los microcontroladores mas modestos.
- A medida que aumentamos la escala del procesador base, la cantidad de niveles aumenta consistentemente

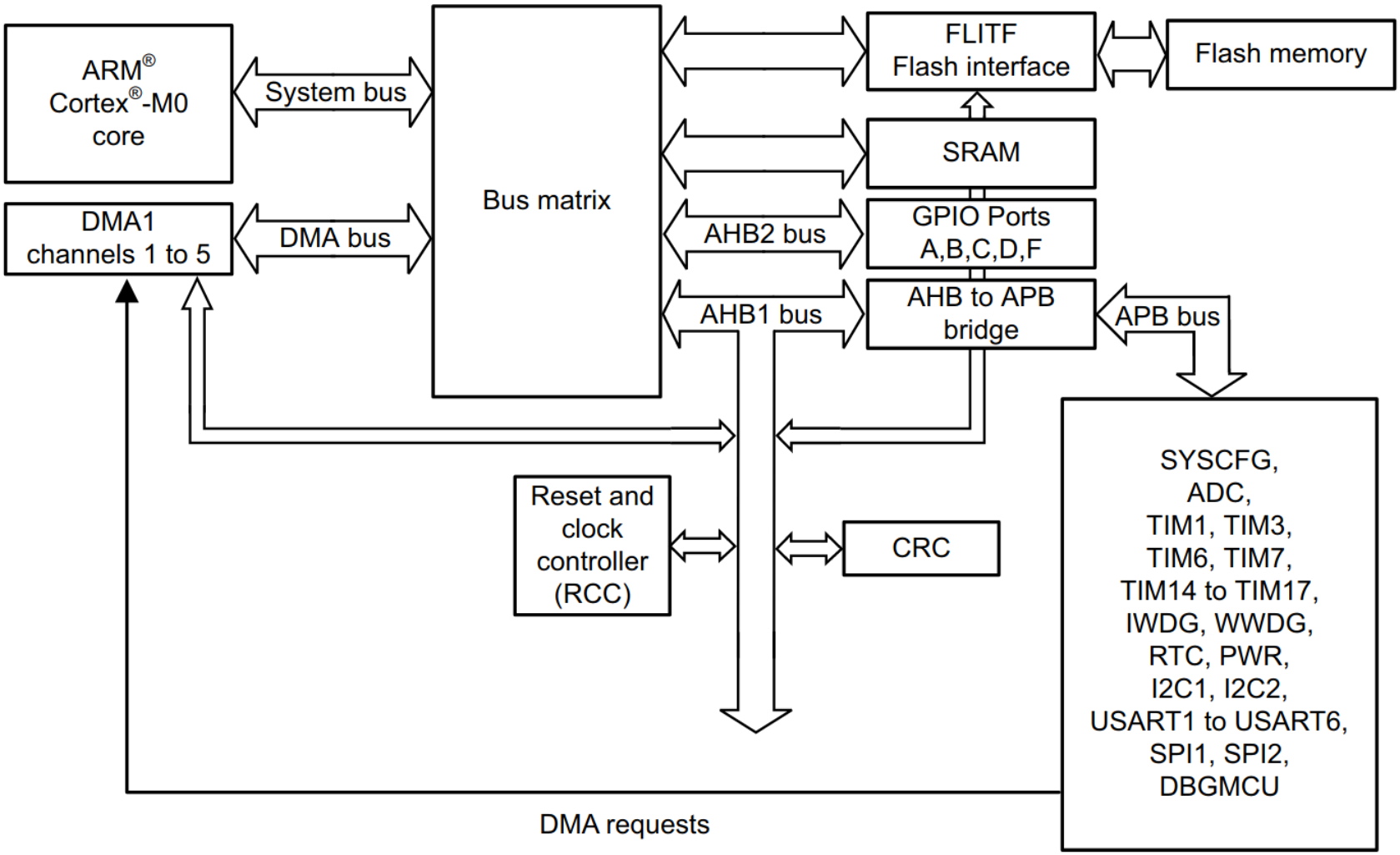
Costos de Tecnología

- Un sistema jerárquico de memoria, tiene por objetivo que el comportamiento el sistema se aproxime al del componente mas rápido, logrando el costo por bit mas bajo posible, y la mayor eficiencia de energía posible.
- Independientemente de la escala del sistema bajo análisis, siempre vamos a poder identificar la presencia de diferentes niveles jerárquicos.
- Al menos dos niveles en los microcontroladores mas modestos.
- A medida que aumentamos la escala del procesador base, la cantidad de niveles aumenta consistentemente
- Veamos:

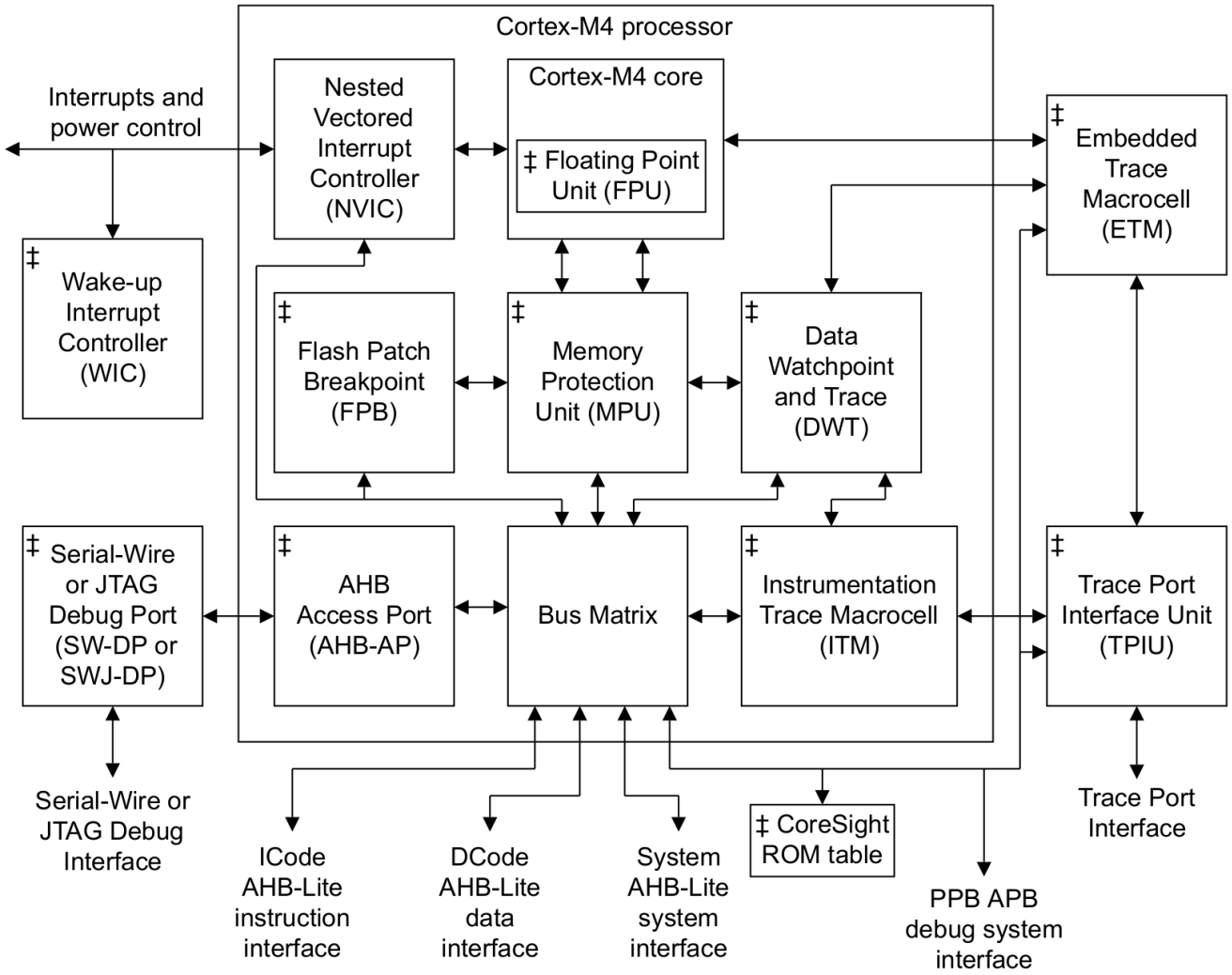
Caso 1: Cortex M0



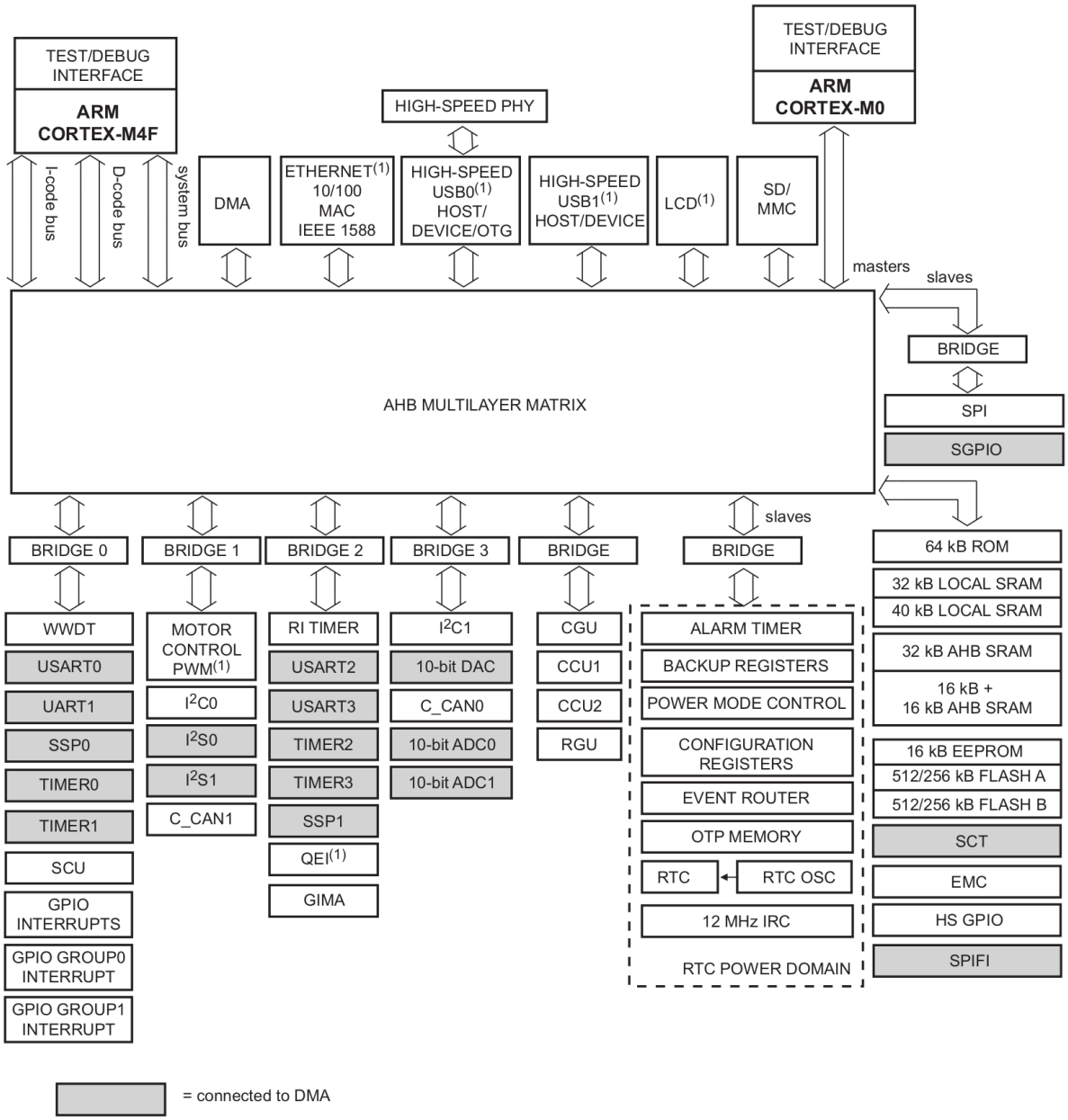
Caso 1: STM32F030x



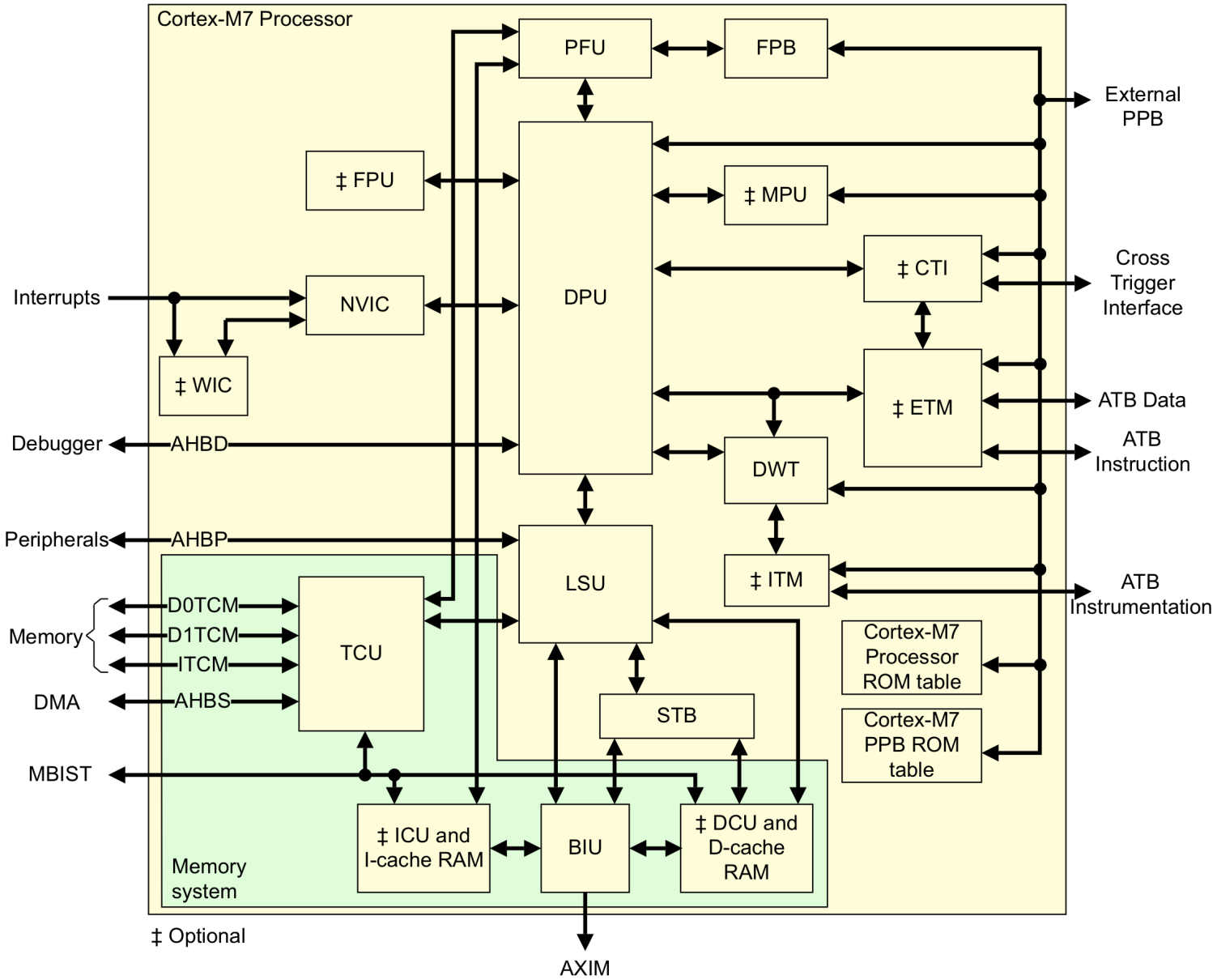
Caso 2: Cortex M4



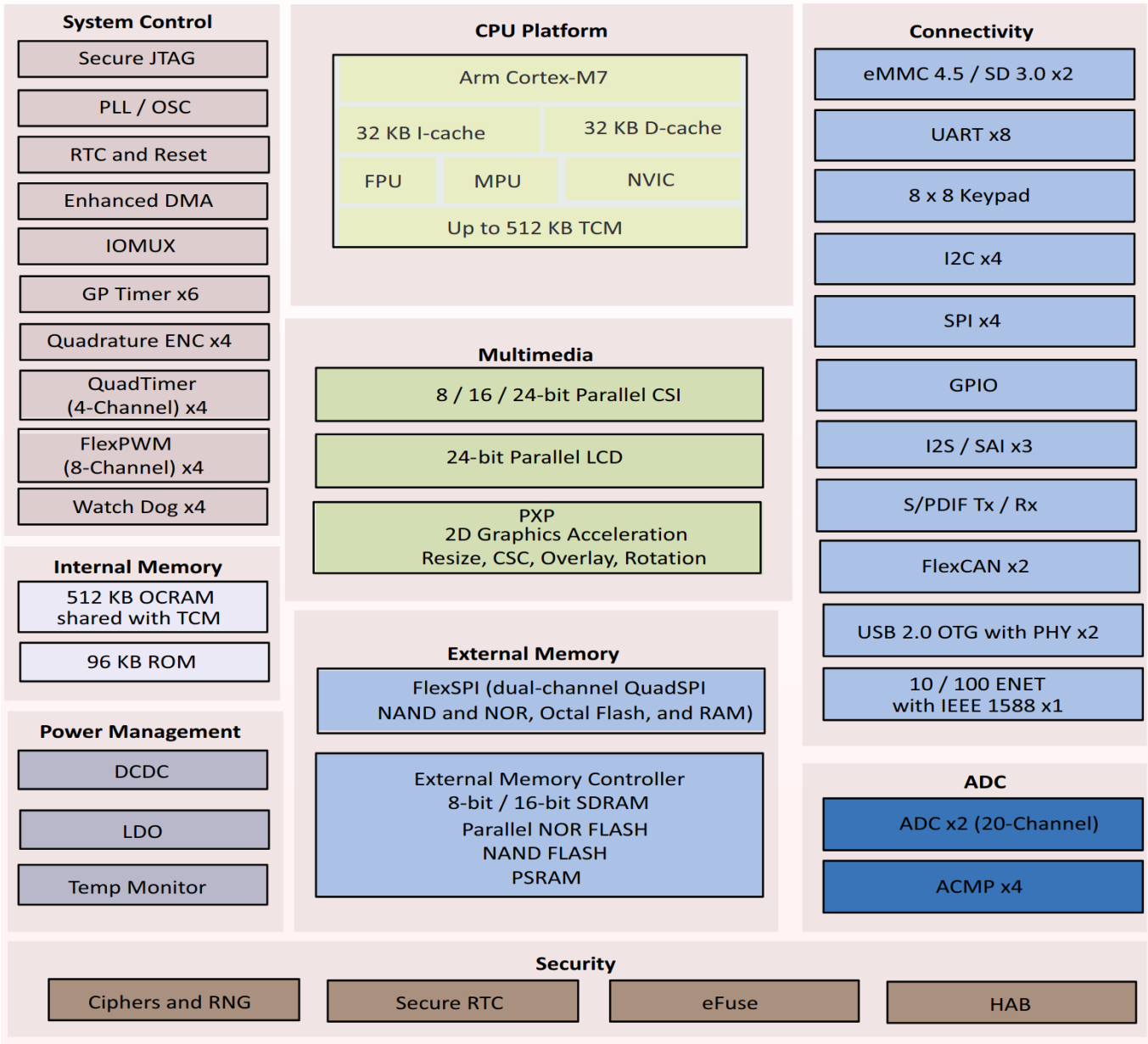
Caso 2: LPC433x



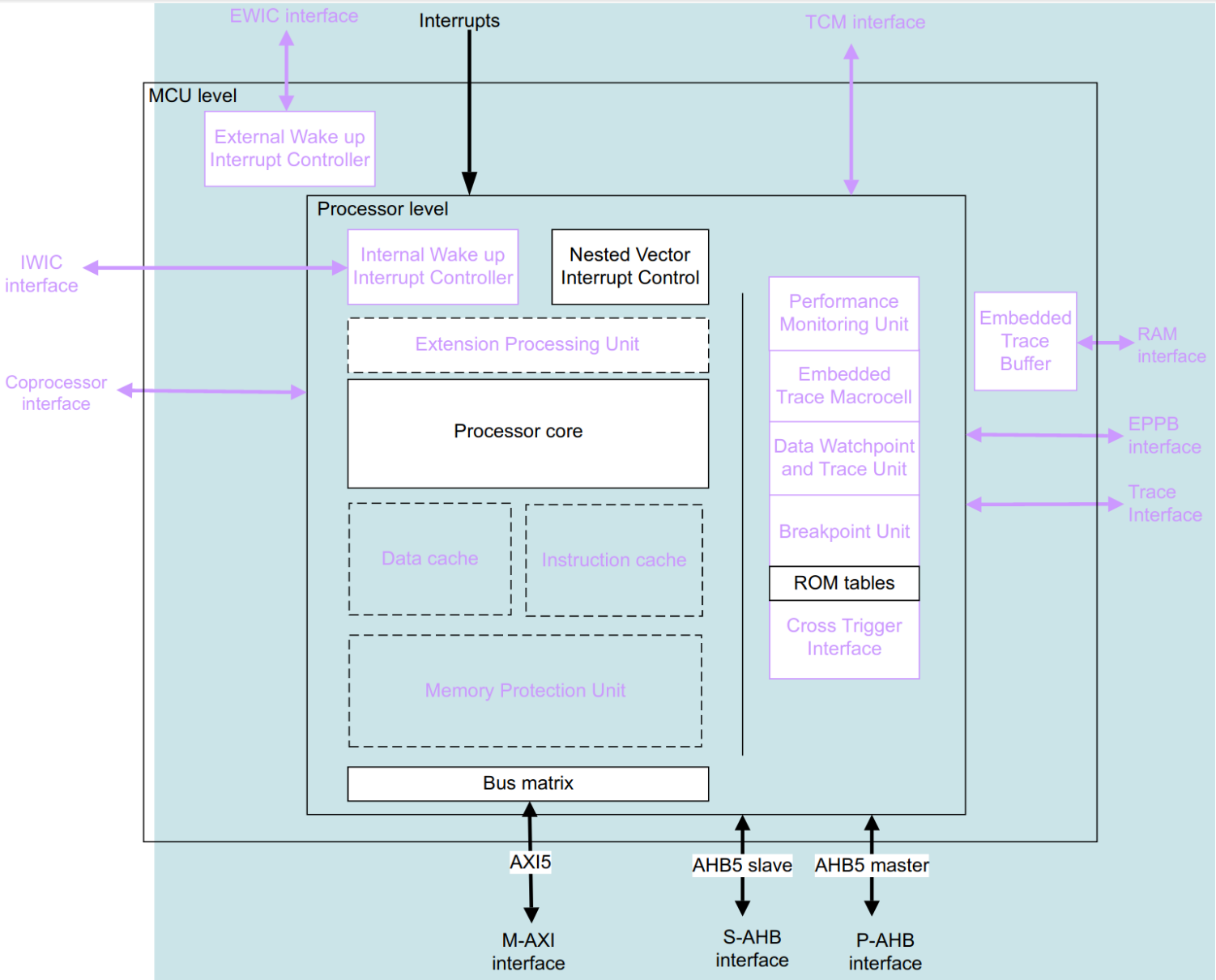
Caso 3: CortexM7



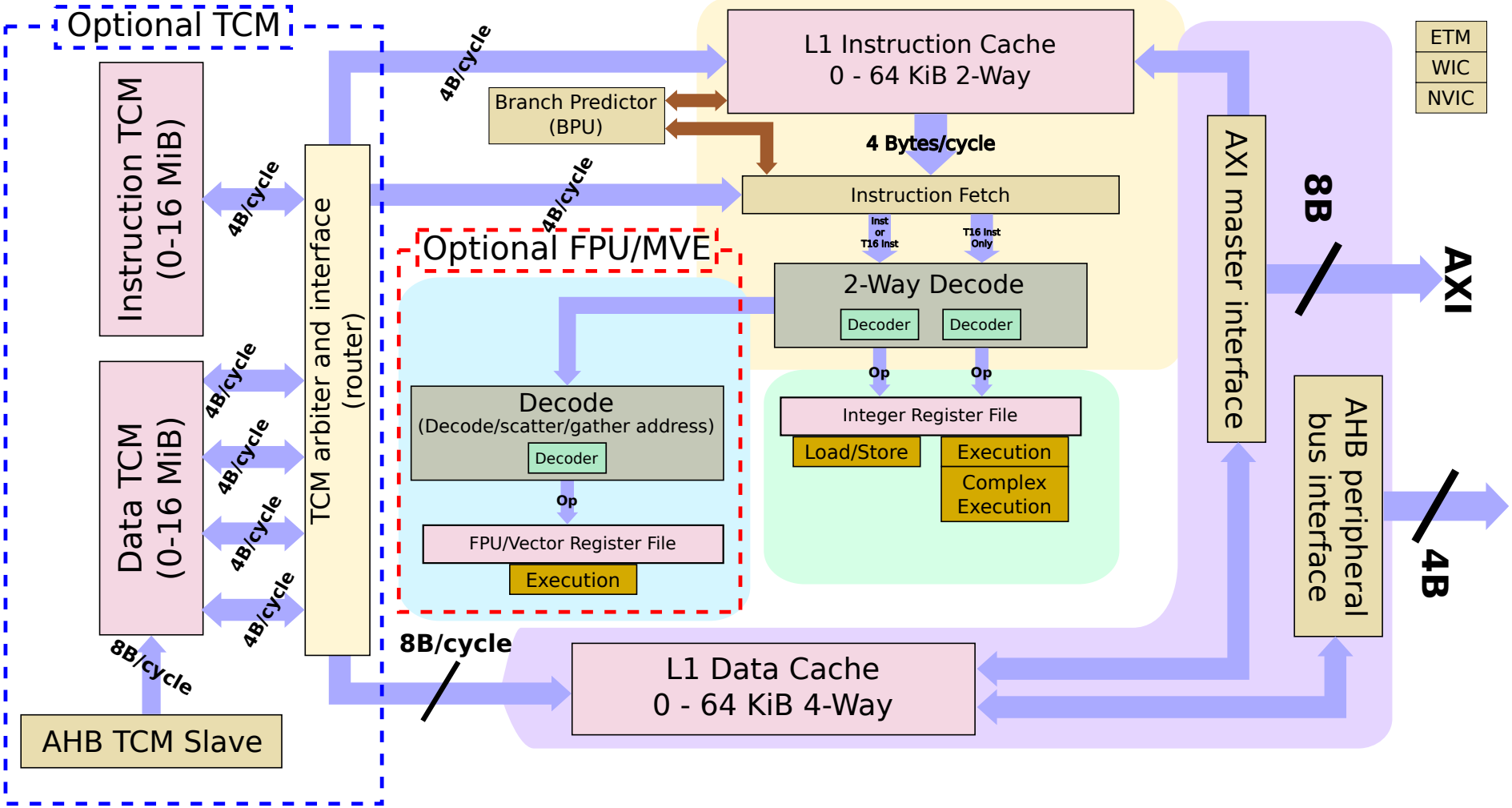
Caso 3: iMX RT 1050



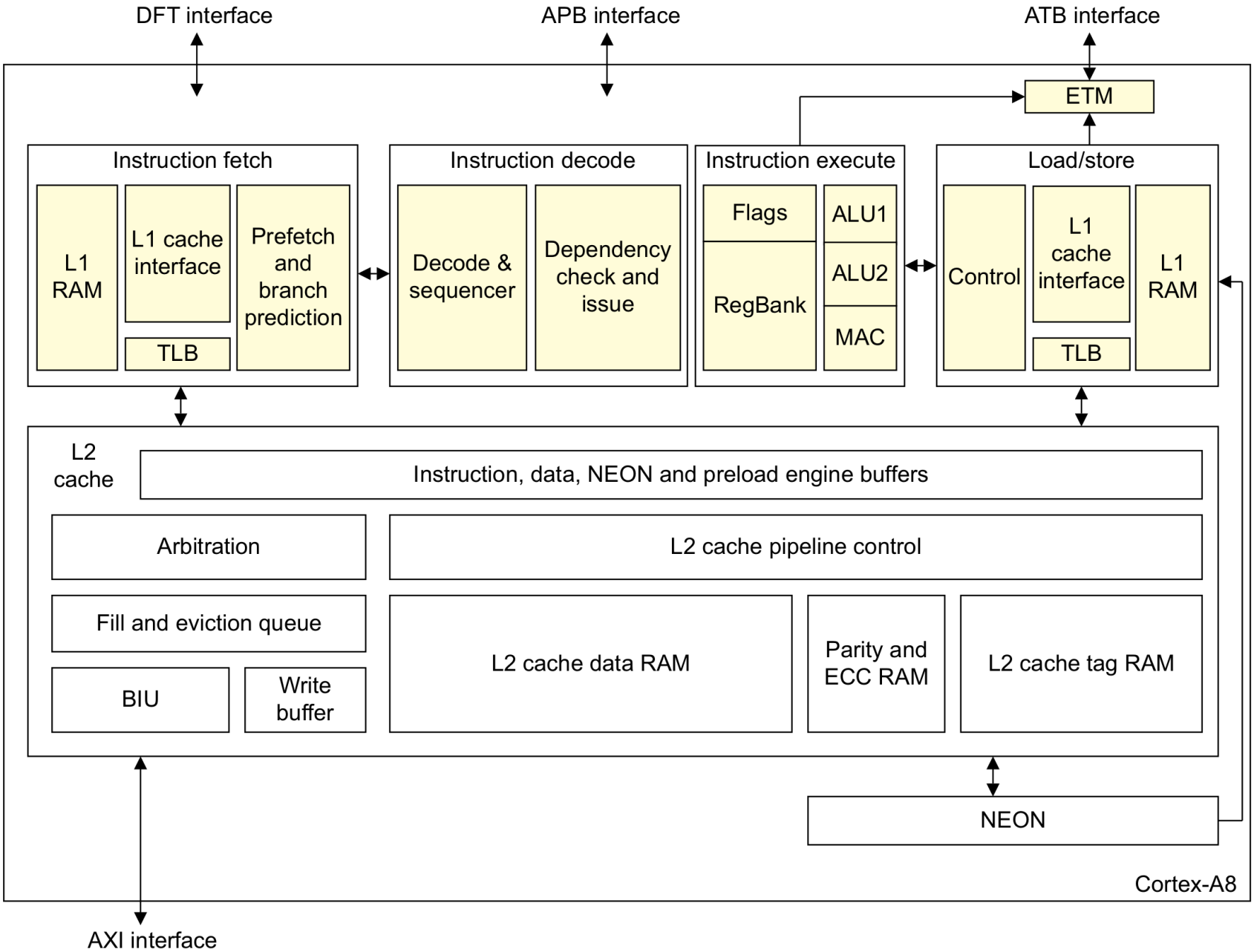
Caso 4: Cortex M55



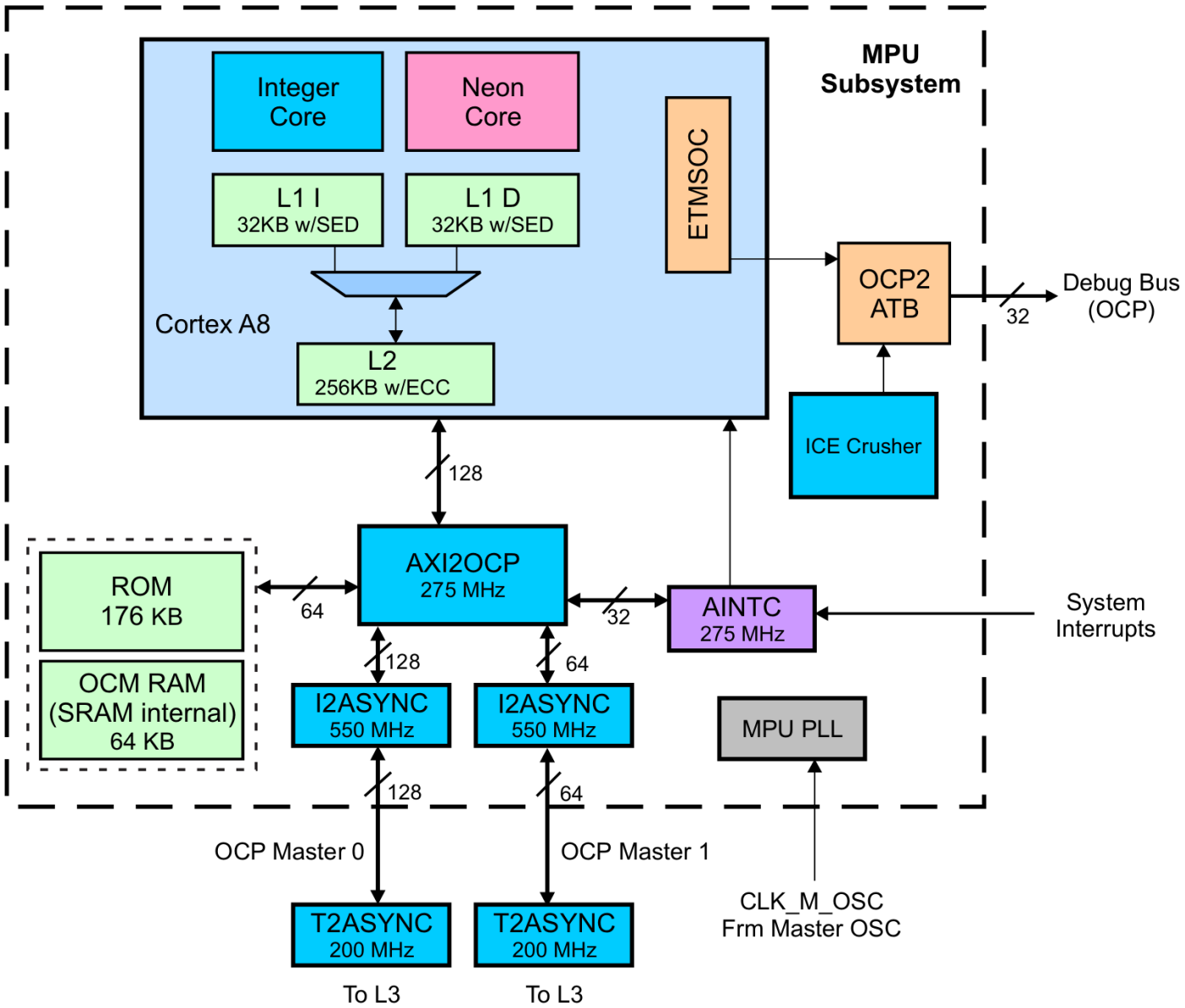
Caso 4: Cortex M55



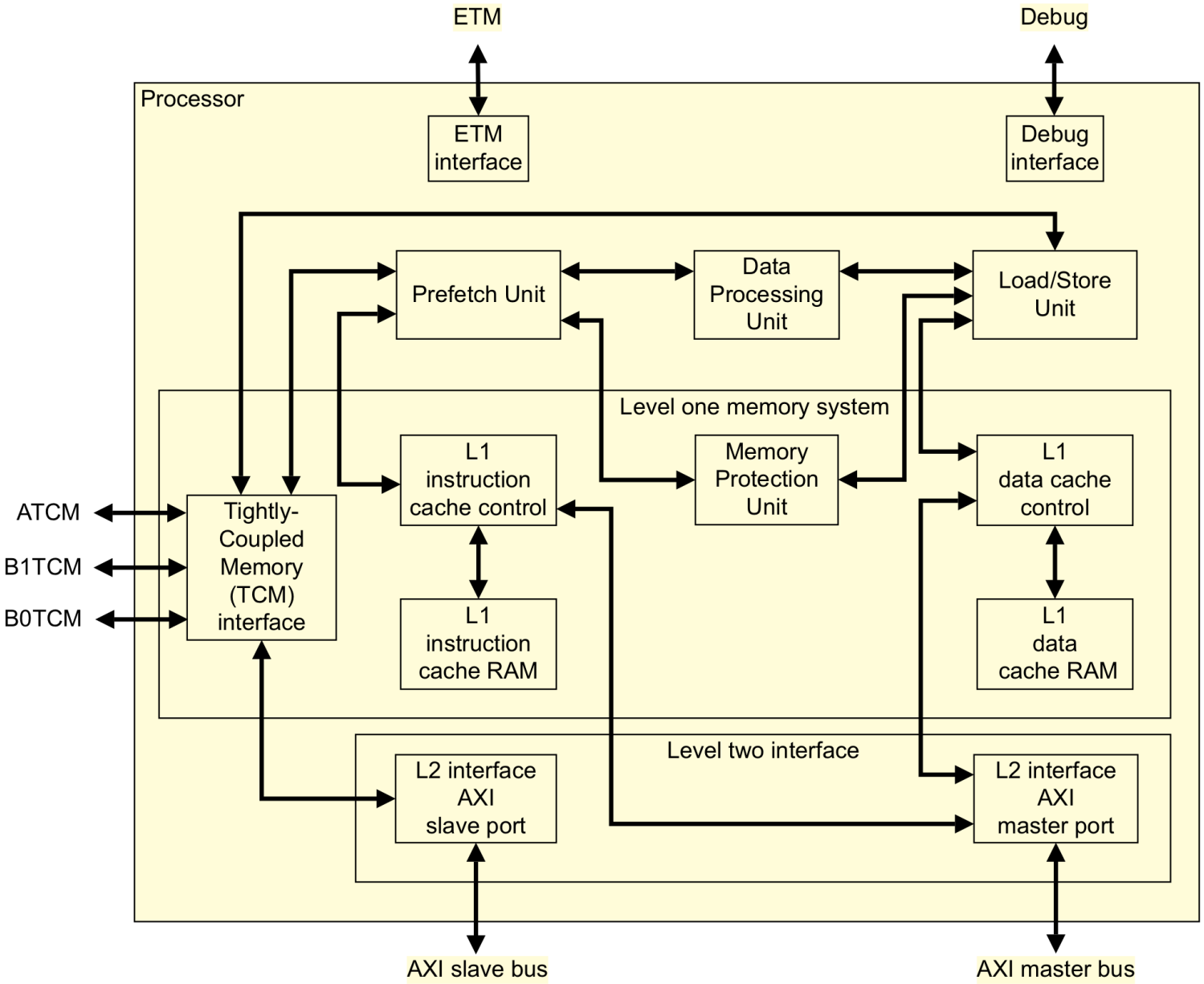
Caso 5: Cortex A8



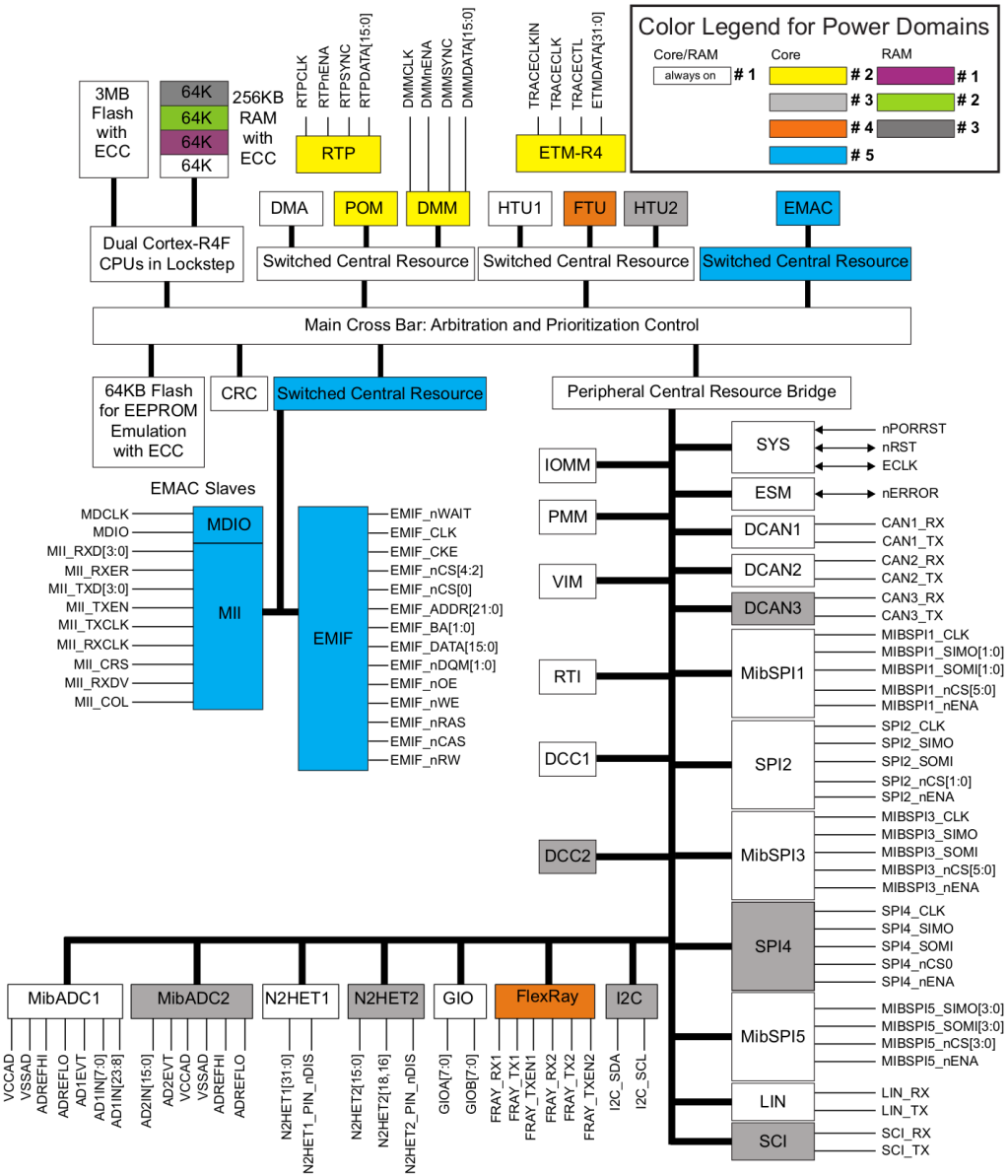
Caso 5: Sitara 335x



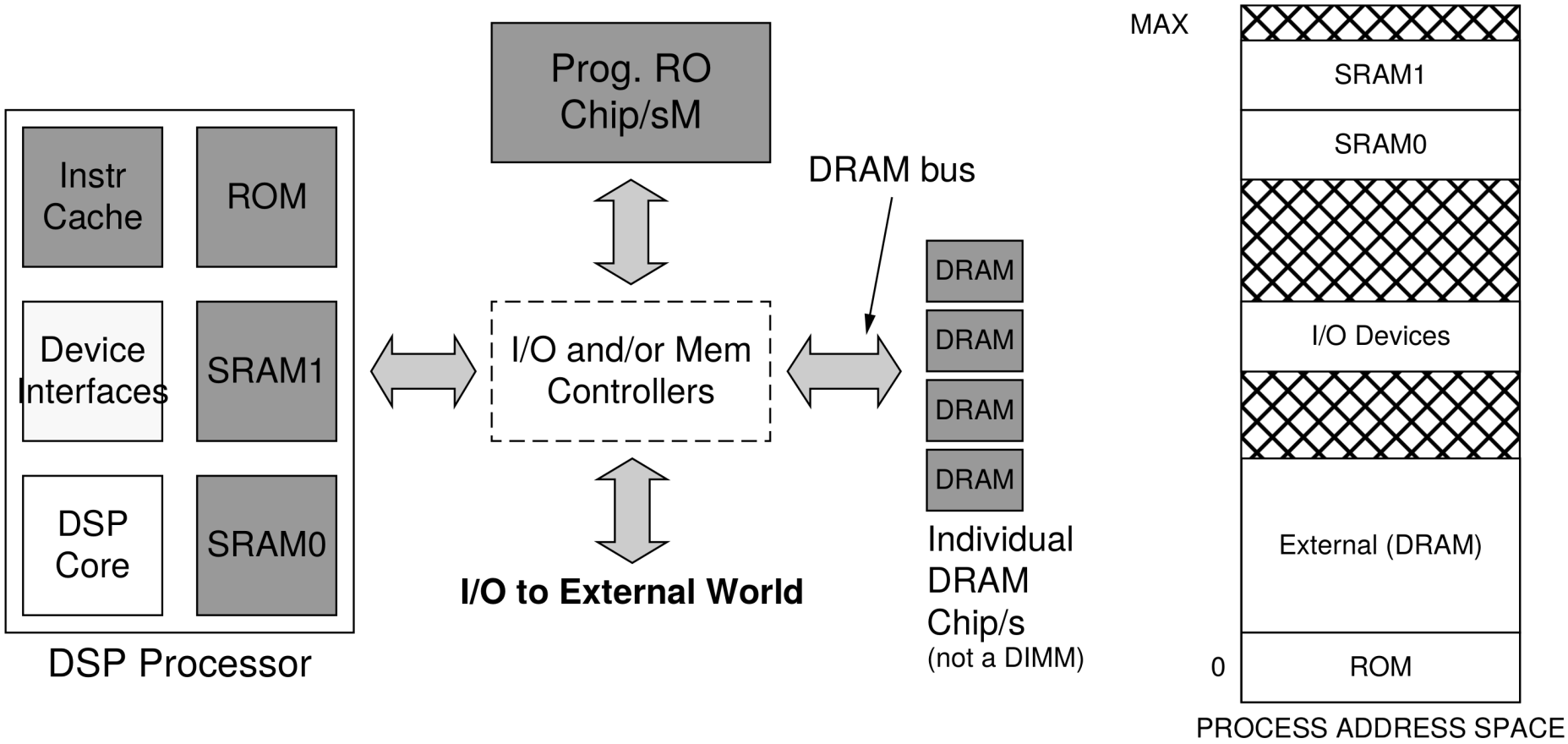
Caso 6: Cortex R4



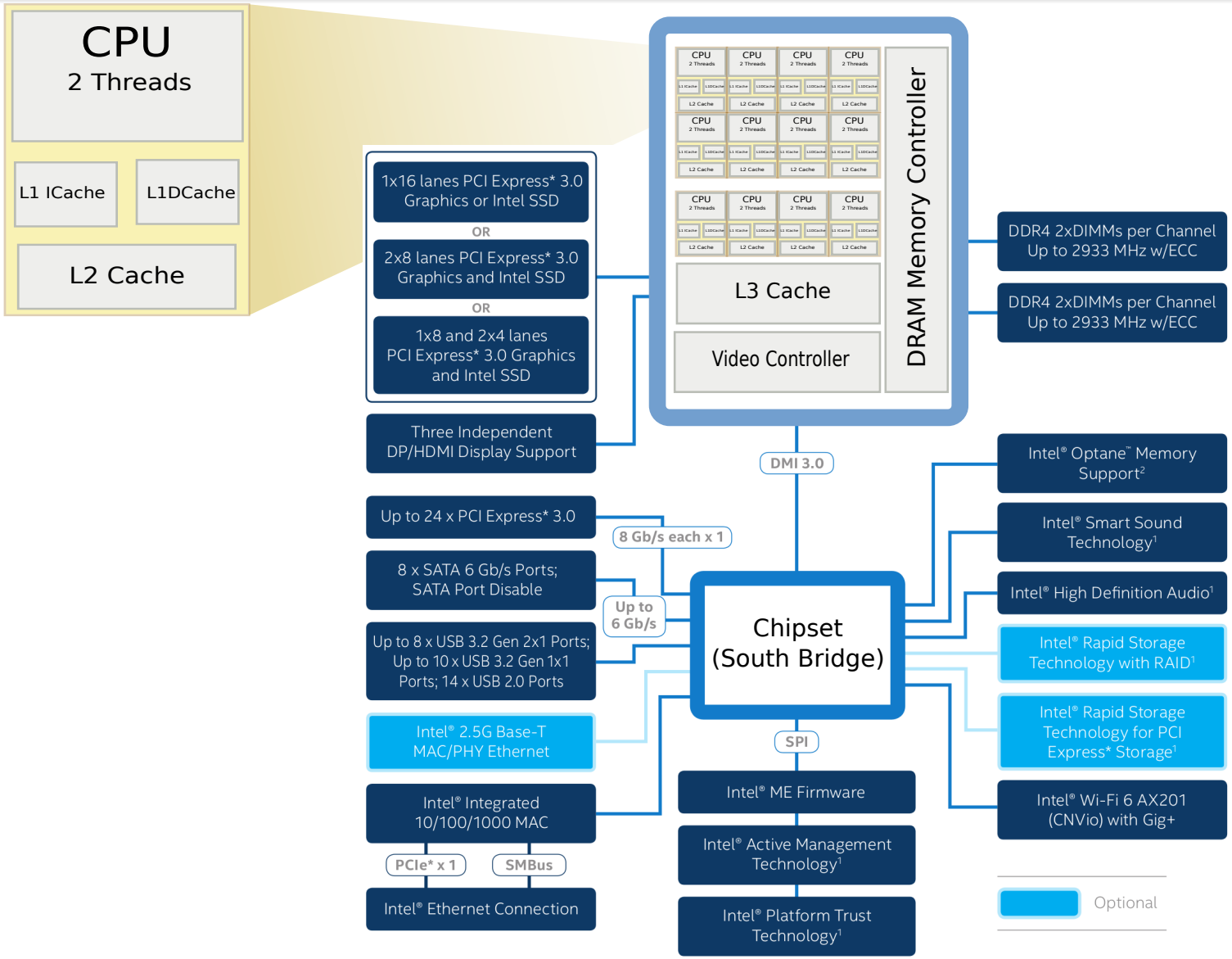
Caso 6: TMS570



Caso 7: Un DSP embedded no ARM



Caso 8: Fuerza bruta



Temario

Potencia Disipada en un sistema de cómputo

Potencia Disipada en un sistema de cómputo

- La potencia dispada en circuitos basados en transistores CMOS proviene de dos factores

Potencia Disipada en un sistema de cómputo

- La potencia dispada en circuitos basados en transistores CMOS proviene de dos factores

✓ ***Potencia estática (leakage power)***: Proviene del hecho que un transistor CMOS cuando está en estado de corte no está completamente “apagado”.

Potencia Disipada en un sistema de cómputo

- La potencia dispada en circuitos basados en transistores CMOS proviene de dos factores

✓ ***Potencia estática (leakage power)***: Proviene del hecho que un transistor CMOS cuando está en estado de corte no está completamente "apagado".

✓ ***Potencia dinámica***: Resultado de conmutar a una carga capacitiva en la malla de salida entre dos estados de tensión.

Potencia Disipada en un sistema de cómputo

- La potencia dispada en circuitos basados en transistores CMOS proviene de dos factores

✓ ***Potencia estática (leakage power)***: Proviene del hecho que un transistor CMOS cuando está en estado de corte no está completamente "apagado".

✓ ***Potencia dinámica***: Resultado de conmutar a una carga capacitiva en la malla de salida entre dos estados de tensión.

- La ***Potencia dinámica*** depende de la actividad de conmutación del circuito. Es decir de la frecuencia con que conmute. Si no cambia el valor de tensión en al salida del CMOS, no hay conmutación, y no se disipa potencia.

Potencia Disipada en un sistema de cómputo

- La potencia dispada en circuitos basados en transistores CMOS proviene de dos factores

✓ ***Potencia estática (leakage power)***: Proviene del hecho que un transistor CMOS cuando está en estado de corte no está completamente "apagado".

✓ ***Potencia dinámica***: Resultado de conmutar a una carga capacitiva en la malla de salida entre dos estados de tensión.

- La ***Potencia dinámica*** depende de la actividad de conmutación del circuito. Es decir de la frecuencia con que conmute. Si no cambia el valor de tensión en al salida del CMOS, no hay conmutación, y no se disipa potencia.
- Por su parte la ***Potencia estática*** es independiente de la frecuencia, y existe simplemente porque el chip está alimentado.

leakage

leakage

- La tecnología CMOS se destacó por su potencia de leakage prácticamente despreciable. Esa fue la razón de su adopción para transistores de circuitos lógicos.

leakage

- La tecnología CMOS se destacó por su potencia de leakage prácticamente despreciable. Esa fue la razón de su adopción para transistores de circuitos lógicos.
- Conforme avanzó la tecnología de integración, el tamaño de los transistores se redujo notablemente (scaling)

leakage

- La tecnología CMOS se destacó por su potencia de leakage prácticamente despreciable. Esa fue la razón de su adopción para transistores de circuitos lógicos.
- Conforme avanzó la tecnología de integración, el tamaño de los transistores se redujo notablemente (scaling)
- La Potencia dinámica se reduce linealmente con el tamaño del gate del transistor CMOS.

leakage

- La tecnología CMOS se destacó por su potencia de leakage prácticamente despreciable. Esa fue la razón de su adopción para transistores de circuitos lógicos.
- Conforme avanzó la tecnología de integración, el tamaño de los transistores se redujo notablemente (scaling)
- La Potencia dinámica se reduce linealmente con el tamaño del gate del transistor CMOS.
- La potencia de leakage no.

leakage

- La tecnología CMOS se destacó por su potencia de leakage prácticamente despreciable. Esa fue la razón de su adopción para transistores de circuitos lógicos.
- Conforme avanzó la tecnología de integración, el tamaño de los transistores se redujo notablemente (scaling)
- La Potencia dinámica se reduce linealmente con el tamaño del gate del transistor CMOS.
- La potencia de leakage no.
- La consecuencia de esta situación es que con el tiempo pasó a ser mas significativa que la potencia dinámica.

leakage

- La tecnología CMOS se destacó por su potencia de leakage prácticamente despreciable. Esa fue la razón de su adopción para transistores de circuitos lógicos.
- Conforme avanzó la tecnología de integración, el tamaño de los transistores se redujo notablemente (scaling)
- La Potencia dinámica se reduce linealmente con el tamaño del gate del transistor CMOS.
- La potencia de leakage no.
- La consecuencia de esta situación es que con el tiempo pasó a ser mas significativa que la potencia dinámica.
- En 2005 aproximadamente los diseñadores tenían como objetivo de diseño mantener la potencia de leakage en no mas del 25 % de la potencia disipada total.

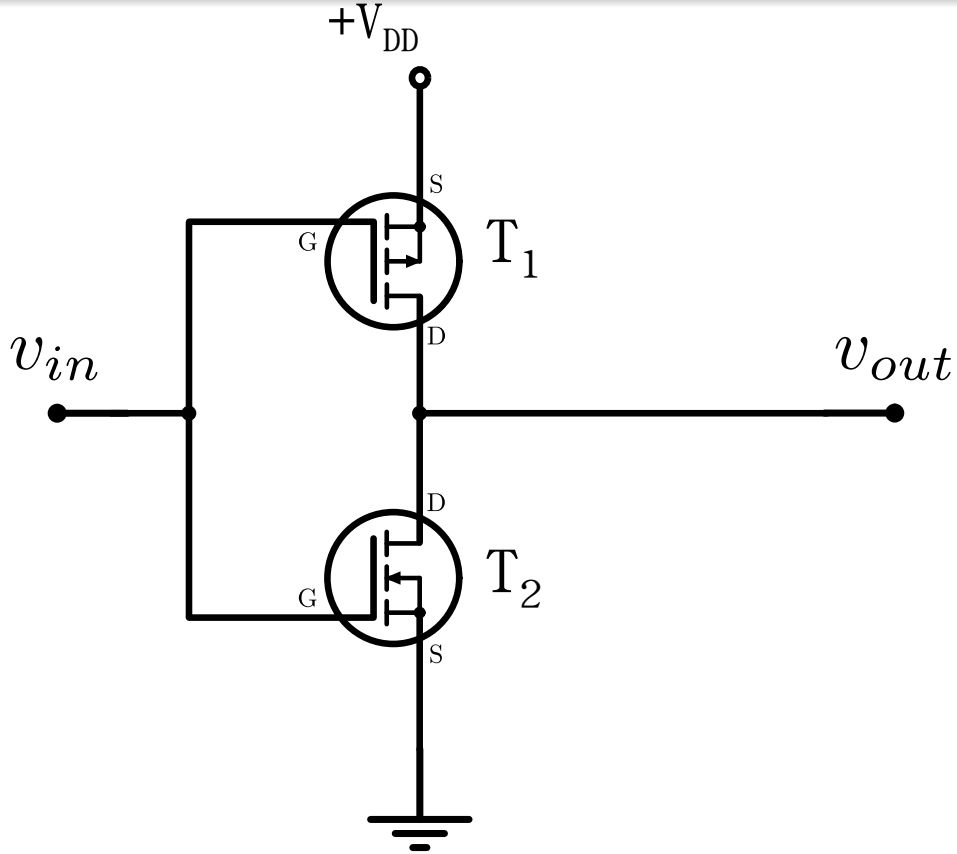
leakage

- La tecnología CMOS se destacó por su potencia de leakage prácticamente despreciable. Esa fue la razón de su adopción para transistores de circuitos lógicos.
- Conforme avanzó la tecnología de integración, el tamaño de los transistores se redujo notablemente (scaling)
- La Potencia dinámica se reduce linealmente con el tamaño del gate del transistor CMOS.
- La potencia de leakage no.
- La consecuencia de esta situación es que con el tiempo pasó a ser mas significativa que la potencia dinámica.
- En 2005 aproximadamente los diseñadores tenían como objetivo de diseño mantener la potencia de leakage en no mas del 25 % de la potencia disipada total.
- Luego de los 30nm a 40nm, la Potencia de leakage ya no disminuye con el scaling.

Conmutación con carga capacitiva

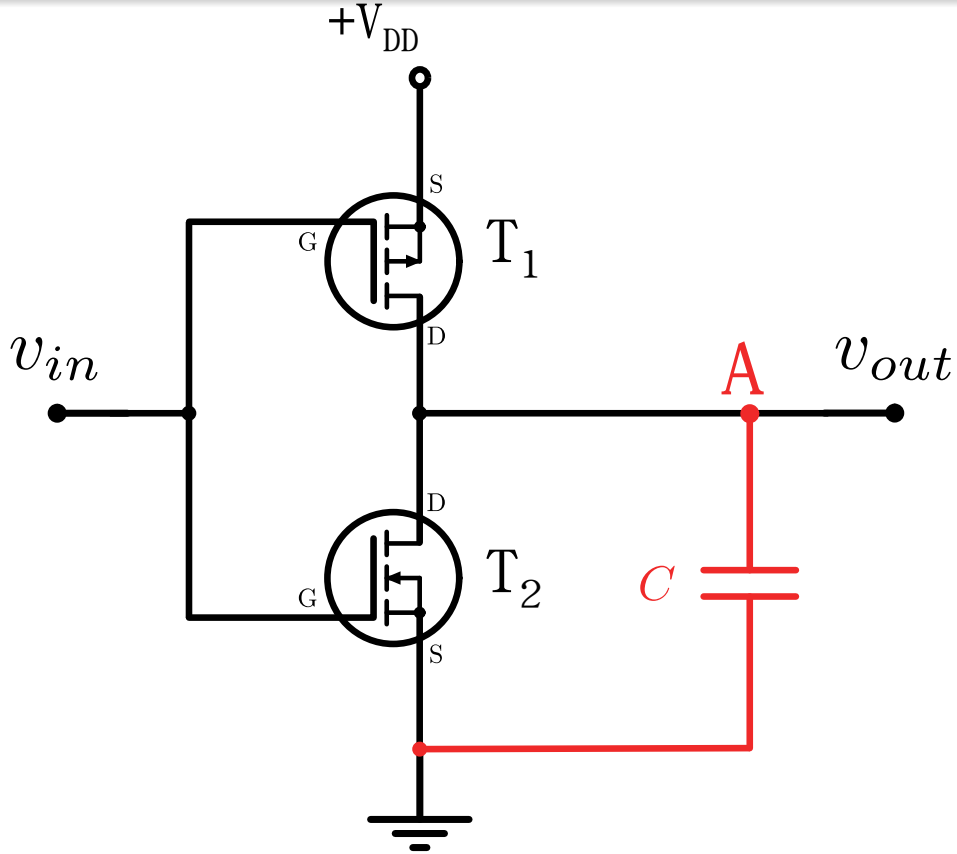
Conmutación con carga capacitiva

- A la derecha un circuito Inversor CMOS



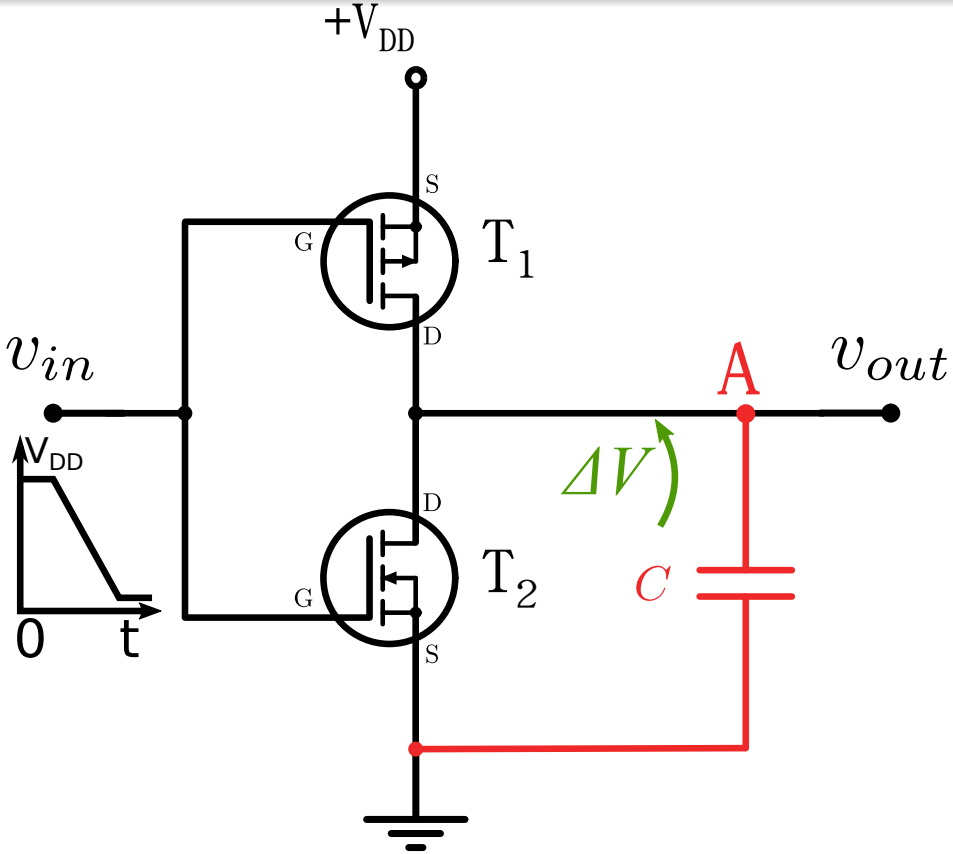
Conmutación con carga capacitiva

- A la derecha un circuito Inversor CMOS
- Si lo cargamos con otro CMOS la carga es capacitiva.



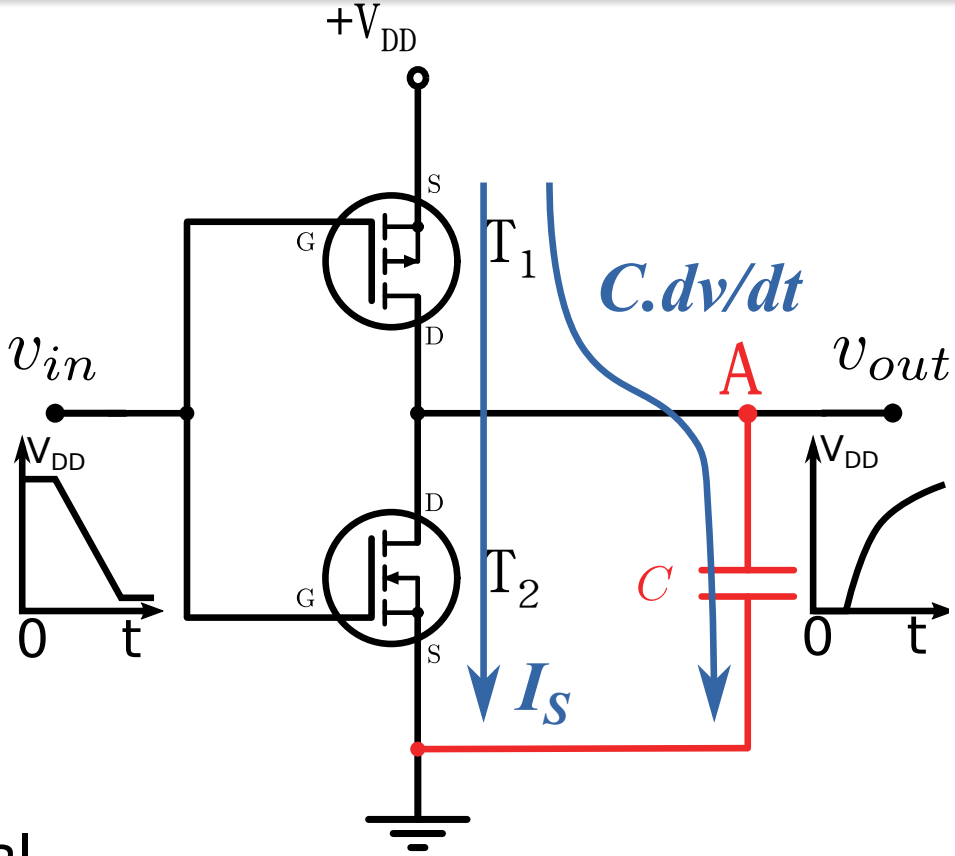
Conmutación con carga capacitiva

- A la derecha un circuito Inversor CMOS
- Si lo cargamos con otro CMOS la carga es capacitiva.
- Si en el nodo **A** se produce una variación de tensión ΔV, se genera una corriente para cargar el capacitor C ΔV Volts, y descargarlo a su valor de tensión original.



Conmutación con carga capacitiva

- A la derecha un circuito Inversor CMOS
- Si lo cargamos con otro CMOS la carga es capacitiva.
- Si en el nodo **A** se produce una variación de tensión ΔV, se genera una corriente para cargar el capacitor C ΔV Volts, y descargarlo a su valor de tensión original.
- Se genera un flujo de carga igual a $C.\Delta V$ desde el Nodo V_{DD},hasta el capacitor y desde este por la malla de descarga.



Conmutación con carga capacitiva

Finalizado el ciclo carga/descarga el CMOS y el capacitor movieron desde V_{DD} hasta tierra una cantidad de carga eléctrica igual a $C.\Delta V$. Esto significa que han utilizado una cantidad de Energía igual a $C.\Delta V.V_{DD}$
Y esa Energía es independiente del ciclo de tiempo en el que se realiza el movimiento de carga

Conmutación con carga capacitiva

La potencia dinámica promedio de este nodo es la velocidad a la que se consume esta energía, que viene dada por [1]:

$$P_{dyn} = \frac{C.\Delta V.V_{DD}.\alpha}{T}, \tag{9}$$

[1] Chapter 3. Low Power Digital CMOS Design. Chandrakasan & R. W. Brodersen. Publisher: Springer US, Year: 1995

Conmutación con carga capacitiva

La potencia dinámica promedio de este nodo es la velocidad a la que se consume esta energía, que viene dada por [1]:

$$P_{dyn} = \frac{C.\Delta V.V_{DD}.\alpha}{T}, \tag{9}$$

donde T es el período de carga / descarga, es decir, la inversa de la frecuencia de clock, y la *razón de actividad* $\alpha, /0 \leqslant \alpha \leqslant 1$, es la probabilidad que el nodo conmute, en cuyo caso consumirá energía (si el nodo no conmuta no se consume energía).

[1] Chapter 3. Low Power Digital CMOS Design. Chandrakasan & R. W. Brodersen. Publisher: Springer US, Year: 1995

Conmutación con carga capacitiva

La potencia dinámica promedio de este nodo es la velocidad a la que se consume esta energía, que viene dada por [1]:

$$P_{dyn} = \frac{C.\Delta V.V_{DD}.\alpha}{T}, \tag{9}$$

donde T es el período de carga / descarga, es decir, la inversa de la frecuencia de clock, y la *razón de actividad* $\alpha, /0 \leqslant \alpha \leqslant 1$, es la probabilidad que el nodo conmute, en cuyo caso consumirá energía (si el nodo no conmuta no se consume energía).
Incluir α permite estimar el consumo del nodo durante mucho mas que un período de la señal de clock, permitiendo calcular la potencia promedio durante horas enteras de computación, siempre que la *razón de actividad* se mantenga.

[1] Chapter 3. Low Power Digital CMOS Design. Chandrakasan & R. W. Brodersen. Publisher: Springer US, Year: 1995

Conmutación con carga capacitiva

La potencia dinámica promedio de este nodo es la velocidad a la que se consume esta energía, que viene dada por [1]:

$$P_{dyn} = \frac{C.\Delta V.V_{DD}.\alpha}{T}, \tag{9}$$

donde T es el período de carga / descarga, es decir, la inversa de la frecuencia de clock, y la *razón de actividad* $\alpha, /0 \leqslant \alpha \leqslant 1$, es la probabilidad que el nodo conmute, en cuyo caso consumirá energía (si el nodo no conmuta no se consume energía).

Incluir α permite estimar el consumo del nodo durante mucho mas que un período de la señal de clock, permitiendo calcular la potencia promedio durante horas enteras de computación, siempre que la *razón de actividad* se mantenga.

La suma de la ecuación (9) a lo largo de todos los nodos del chip dá como resultado la ***Potencia Dinámica*** total.

[1] Chapter 3. Low Power Digital CMOS Design. Chandrakasan & R. W. Brodersen. Publisher: Springer US, Year: 1995

Conmutación con carga capacitiva

De la ecuación (9) surge que si disminuyen la Capacidad de carga, o V_{DD}, disminuirá de manera directa la ***Potencia Dinámica***.

Conmutación con carga capacitiva

De la ecuación (9) surge que si disminuyen la Capacidad de carga, o V_{DD}, disminuirá de manera directa la ***Potencia Dinámica***.
En un circuito lógico la excursión de Tensión $C.\Delta V$ normalmente es desde un valor muy cercano a 0 Volts hasta V_{DD}, de modo que la ecuación (9) se transforma en:

$$P_{dyn} = C.V_{DD}^2.\alpha.f, \tag{10}$$

Conmutación con carga capacitiva

De la ecuación (9) surge que si disminuyen la Capacidad de carga, o V_{DD}, disminuirá de manera directa la ***Potencia Dinámica***.
En un circuito lógico la excursión de Tensión $C.\Delta V$ normalmente es desde un valor muy cercano a 0 Volts hasta V_{DD}, de modo que la ecuación (9) se transforma en:

$$P_{dyn} = C.V_{DD}^2.\alpha.f, \tag{10}$$

Además se verifica empíricamente que la *razón de actividad* en circuitos lógicos es ½, de modo que la ecuación (13) queda:

$$P_{dyn} = \frac{1}{2}.C.V_{DD}^2.f, \tag{11}$$

Conmutación con carga capacitiva

De la ecuación (9) surge que si disminuyen la Capacidad de carga, o V_{DD}, disminuirá de manera directa la ***Potencia Dinámica***.
En un circuito lógico la excursión de Tensión $C.\Delta V$ normalmente es desde un valor muy cercano a 0 Volts hasta V_{DD}, de modo que la ecuación (9) se transforma en:

$$P_{dyn} = C.V_{DD}^2.\alpha.f, \tag{10}$$

Además se verifica empíricamente que la *razón de actividad* en circuitos lógicos es ½, de modo que la ecuación (13) queda:

$$P_{dyn} = \frac{1}{2}.C.V_{DD}^2.f, \tag{11}$$

La ecuación (11) es la que mejor representa la ***Potencia Dinámica*** en un chip, considerando que es la suma de todos los nodos del chip.

leakage

leakage

- Como ya se dijo, es la imposibilidad de apagar por completo al transistor CMOS cuando éste está en el estado de corte.

leakage

- Como ya se dijo, es la imposibilidad de apagar por completo al transistor CMOS cuando éste está en el estado de corte.
- Así que conduce corriente por debajo del umbral de conducción.

leakage

- Como ya se dijo, es la imposibilidad de apagar por completo al transistor CMOS cuando éste está en el estado de corte.
- Así que conduce corriente por debajo del umbral de conducción.
- La compuerta se acopla al canal activo principalmente a través de la capacitancia de óxido de la compuerta.

leakage

- Como ya se dijo, es la imposibilidad de apagar por completo al transistor CMOS cuando éste está en el estado de corte.
- Así que conduce corriente por debajo del umbral de conducción.
- La compuerta se acopla al canal activo principalmente a través de la capacitancia de óxido de la compuerta.
- Hay otras capacitancias en un transistor que acoplan la compuerta del CMOS a una suerte de “carga fija”(una carga que no puede moverse) presente en el bloque y no asociada con el flujo de corriente.

leakage

- Como ya se dijo, es la imposibilidad de apagar por completo al transistor CMOS cuando éste está en el estado de corte.
- Así que conduce corriente por debajo del umbral de conducción.
- La compuerta se acopla al canal activo principalmente a través de la capacitancia de óxido de la compuerta.
- Hay otras capacitancias en un transistor que acoplan la compuerta del CMOS a una suerte de “carga fija”(una carga que no puede moverse) presente en el bloque y no asociada con el flujo de corriente.
- Si estas Capacidades son grandes pueden alterar la polarización de la compuerta cambiando la densidad de carga fija acumulada en el canal impidiendo que este corte.

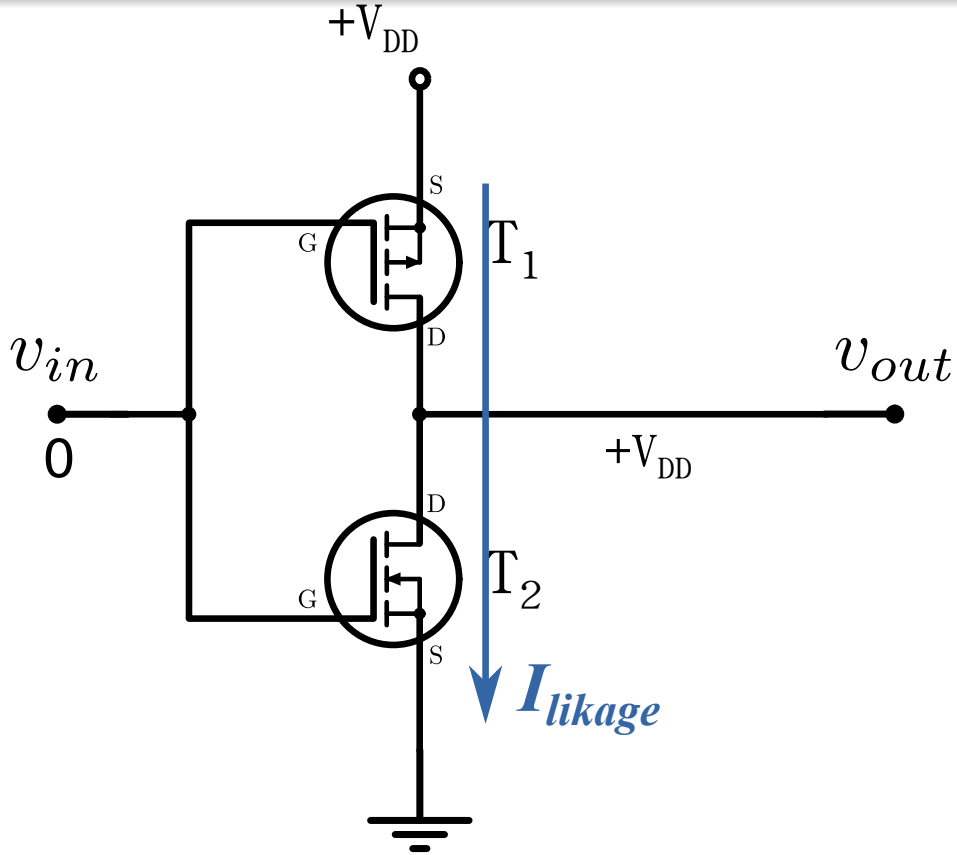
leakage

- Como ya se dijo, es la imposibilidad de apagar por completo al transistor CMOS cuando éste está en el estado de corte.
- Así que conduce corriente por debajo del umbral de conducción.
- La compuerta se acopla al canal activo principalmente a través de la capacitancia de óxido de la compuerta.
- Hay otras capacitancias en un transistor que acoplan la compuerta del CMOS a una suerte de “carga fija”(una carga que no puede moverse) presente en el bloque y no asociada con el flujo de corriente.
- Si estas Capacidades son grandes pueden alterar la polarización de la compuerta cambiando la densidad de carga fija acumulada en el canal impidiendo que este corte.
- Estas capacidades no se reducen con el scaling ya que no dependen de las dimensiones físicas del canal.

leakage

leakage

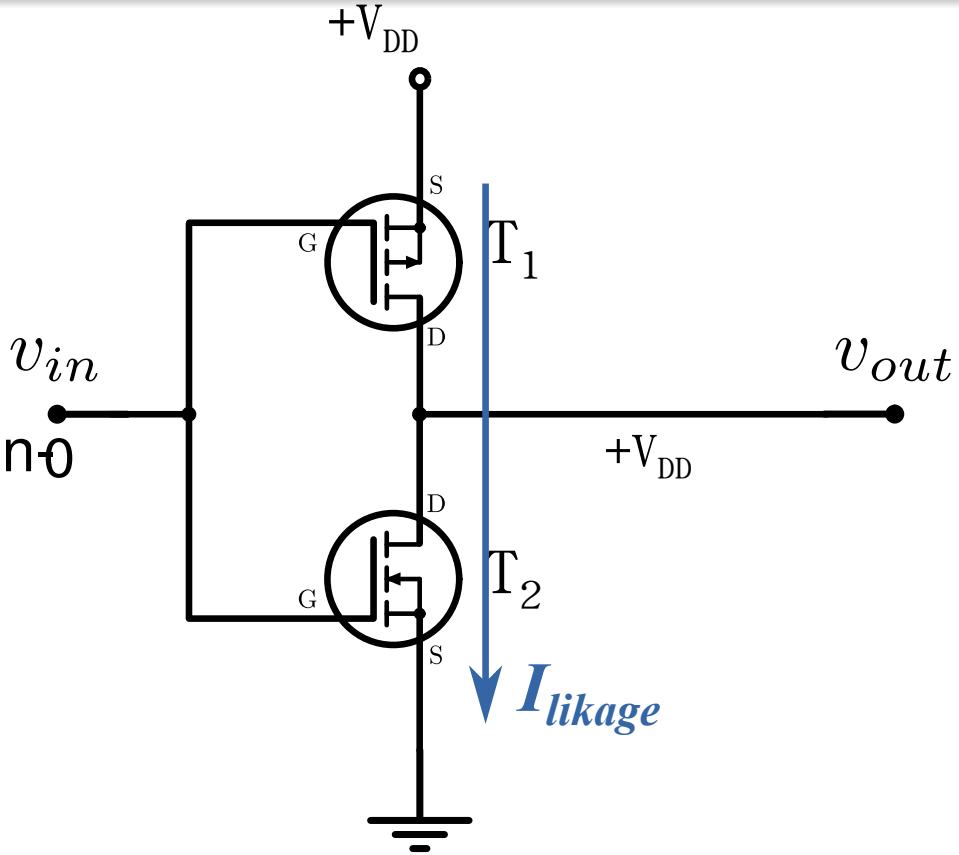
- Una vez terminada la conmutación el CMOS queda en un estado estable como el mostrado en la figura de la derecha.



leakage

- Una vez terminada la conmutación el CMOS queda en un estado estable como el mostrado en la figura de la derecha.
- La Potencia de leakage es linealmente dependiente de la tensión de alimentación, de acuerdo con una expresión bastante simple:

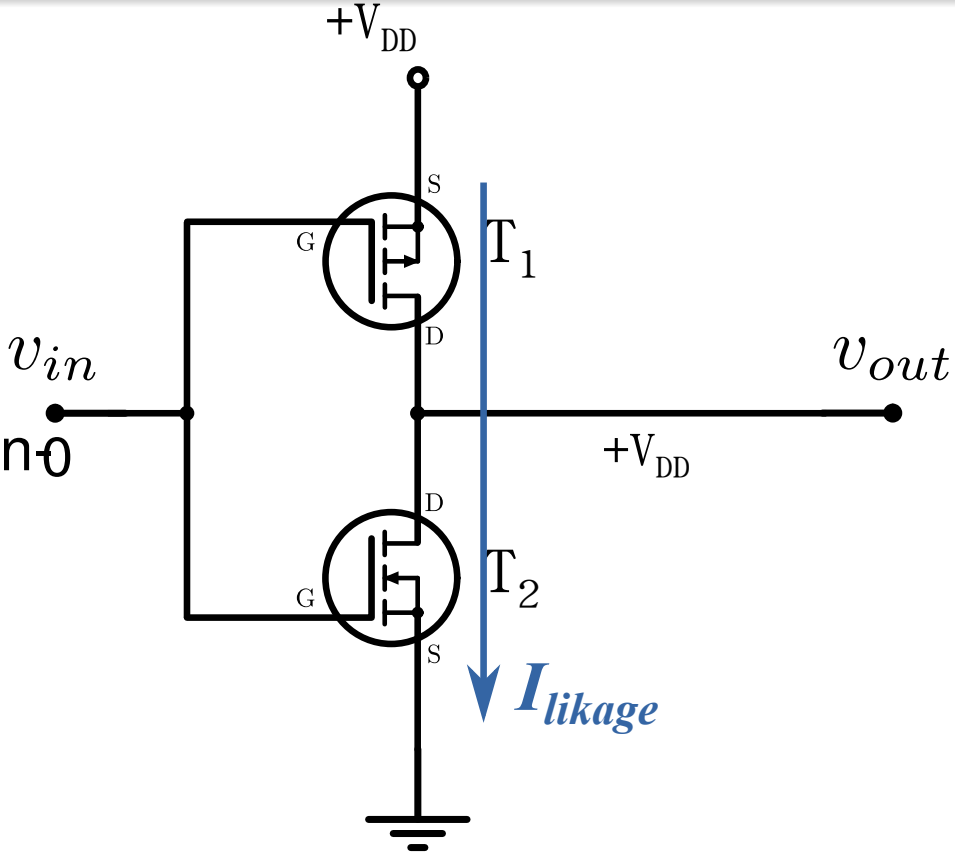
$$P_{stat} = I_{leakage}.V_{DD}, \qquad (12)$$



leakage

- Una vez terminada la conmutación el CMOS queda en un estado estable como el mostrado en la figura de la derecha.
- La Potencia de leakage es linealmente dependiente de la tensión de alimentación, de acuerdo con una expresión bastante simple:

$$P_{stat} = I_{leakage}.V_{DD}, \qquad (12)$$

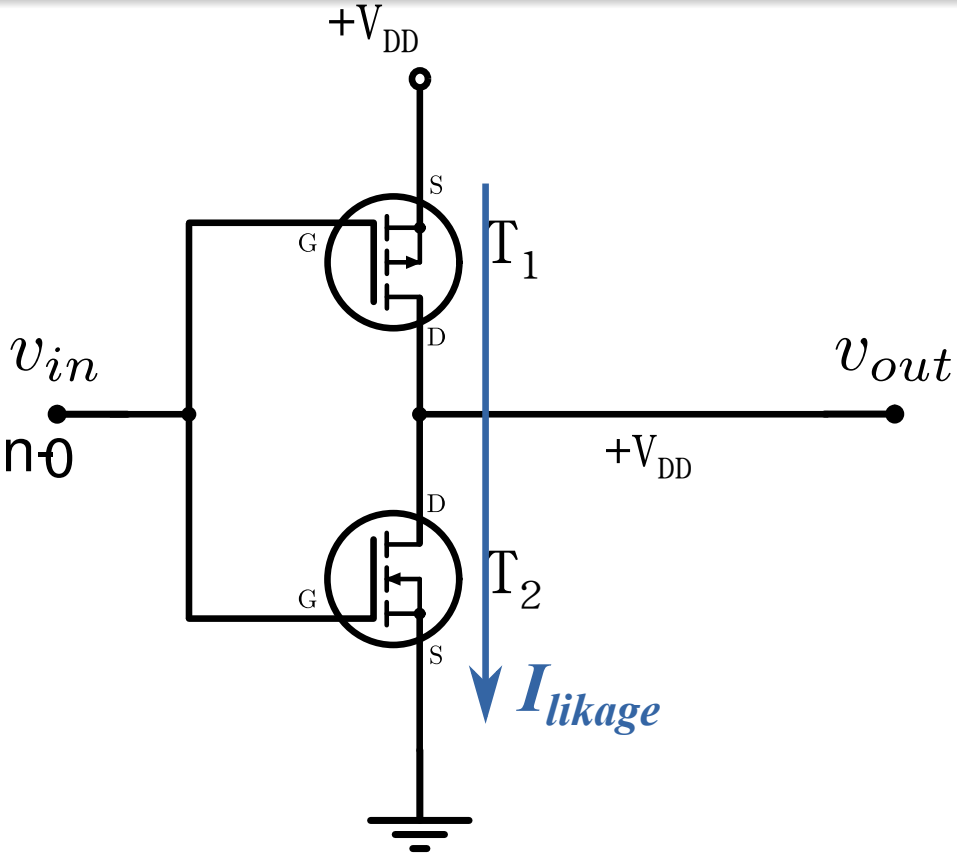


- Y la Energía de leakage es el producto de la Potencia de leakage por el período de operación.

leakage

- Una vez terminada la conmutación el CMOS queda en un estado estable como el mostrado en la figura de la derecha.
- La Potencia de leakage es linealmente dependiente de la tensión de alimentación, de acuerdo con una expresión bastante simple:

$$P_{stat} = I_{leakage} . V_{DD}, \qquad (12)$$



- Y la Energía de leakage es el producto de la Potencia de leakage por el período de operación.
- Para calcularlas a nivel del chip todas las fórmulas se multiplican por la cantidad N de nodos.

leakage

leakage

- Las dimensiones del ancho del gate y del espesor la capa de óxido del transistor CMOS disminuyen linealmente con el avance en el scaling.

leakage

- Las dimensiones del ancho del gate y del espesor la capa de óxido del transistor CMOS disminuyen linealmente con el avance en el scaling.
- No así la tensión de alimentación.

leakage

- Las dimensiones del ancho del gate y del espesor la capa de óxido del transistor CMOS disminuyen linealmente con el avance en el scaling.
- No así la tensión de alimentación.
- Por ello la potencia disipada no disminuye con el scaling, ya que la cantidad de Nodo por área aumenta mas de lo que disminuyen algunos de los drivers de la Potencia Disipada.

leakage

- Las dimensiones del ancho del gate y del espesor la capa de óxido del transistor CMOS disminuyen linealmente con el avance en el scaling.
- No así la tensión de alimentación.
- Por ello la potencia disipada no disminuye con el scaling, ya que la cantidad de Nodo por área aumenta mas de lo que disminuyen algunos de los drivers de la Potencia Disipada.
- Si bien estos problemas son mas graves en los microprocesadores, han afectado también a las memorias.

Estrategias para reducir Potencia y Energía

Estrategias para reducir Potencia y Energía

- En resumen, la energía Total consumida por un chip está dada por:

$$E_{Tot} = [\frac{1}{2}.C_{tot}.V_{DD}^2.f + N_{tot}.I_{leakage}.V_{DD}].T \quad (13)$$

Estrategias para reducir Potencia y Energía

- En resumen, la energía Total consumida por un chip está dada por:

$$E_{Tot} = [\frac{1}{2}.C_{tot}.V_{DD}^2.f + N_{tot}.I_{leakage}.V_{DD}].T \quad (13)$$

- N_{tot} es la cantidad de nodos del chip, C_{tot} es la suma de todas las cargas capacitivas en los N_{tot} nodos, y T el período de operación, durante el cual, la corriente de leakage no cesa de drenar.

Estrategias para reducir Potencia y Energía

- En resumen, la energía Total consumida por un chip está dada por:

$$E_{Tot} = [\frac{1}{2}.C_{tot}.V_{DD}^2.f + N_{tot}.I_{leakage}.V_{DD}].T \quad (13)$$

- N_{tot} es la cantidad de nodos del chip, C_{tot} es la suma de todas las cargas capacitivas en los N_{tot} nodos, y T el período de operación, durante el cual, la corriente de leakage no cesa de drenar.
- Cuando la carga de trabajo es baja los circuitos integrados siguen conmutando aunque no cambien el estado de sus salidas, consumiendo energía para nada.

Estrategias para reducir Potencia y Energía

- En resumen, la energía Total consumida por un chip está dada por:

$$E_{Tot} = [\frac{1}{2}.C_{tot}.V_{DD}^2.f + N_{tot}.I_{leakage}.V_{DD}].T \quad (13)$$

- N_{tot} es la cantidad de nodos del chip, C_{tot} es la suma de todas las cargas capacitivas en los N_{tot} nodos, y T el período de operación, durante el cual, la corriente de leakage no cesa de drenar.
- Cuando la carga de trabajo es baja los circuitos integrados siguen conmutando aunque no cambien el estado de sus salidas, consumiendo energía para nada.
- Una solución es inhibir la frecuencia de clock (clock gating) en los momentos de baja carga para disminuir la pérdida de energía cuando el chip está inactivo.

Estrategias para reducir Potencia y Energía

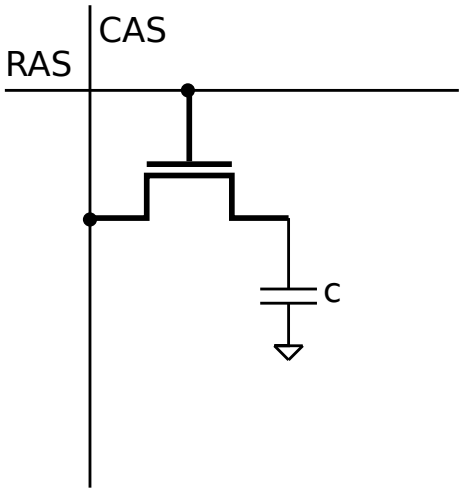
- En resumen, la energía Total consumida por un chip está dada por:

$$E_{Tot} = [\frac{1}{2}.C_{tot}.V_{DD}^2.f + N_{tot}.I_{leakage}.V_{DD}].T \quad (13)$$

- N_{tot} es la cantidad de nodos del chip, C_{tot} es la suma de todas las cargas capacitivas en los N_{tot} nodos, y T el período de operación, durante el cual, la corriente de leakage no cesa de drenar.
- Cuando la carga de trabajo es baja los circuitos integrados siguen conmutando aunque no cambien el estado de sus salidas, consumiendo energía para nada.
- Una solución es inhibir la frecuencia de clock (clock gating) en los momentos de baja carga para disminuir la pérdida de energía cuando el chip está inactivo.
- Para bajar la componente estática solo queda disminuir V_{DD}.

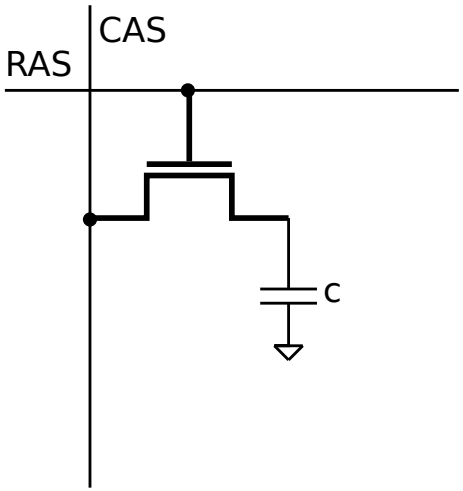
Temario

Memorias dinámicas



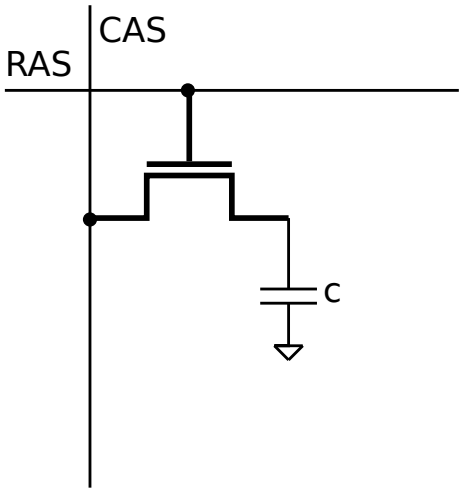
- Almacena la información en forma de estado de carga en un capacitor y la sostiene durante un breve lapso con la ayuda de un transistor.

Memorias dinámicas



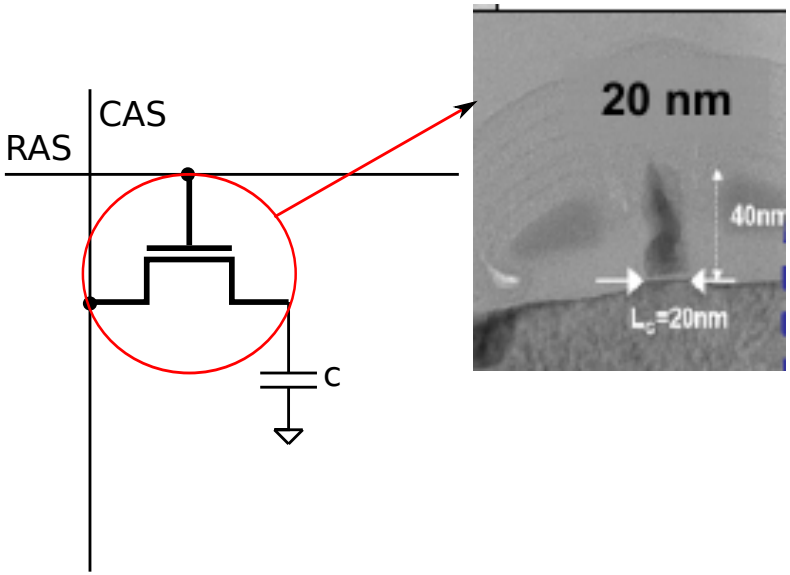
- Almacena la información en forma de estado de carga en un capacitor y la sostiene durante un breve lapso con la ayuda de un transistor.
- Una celda (un bit) se implementa con un solo transistor $=>$ máxima capacidad de almacenamiento por CI.

Memorias dinámicas



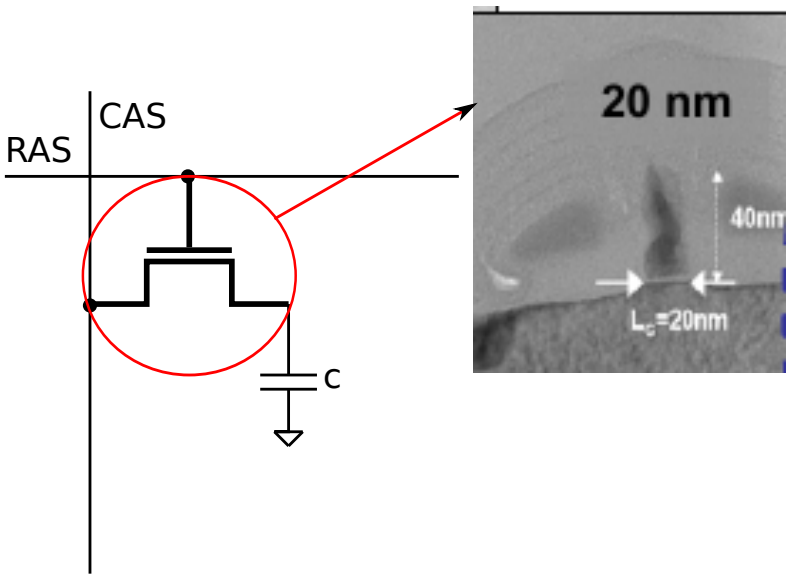
- Almacena la información en forma de estado de carga en un capacitor y la sostiene durante un breve lapso con la ayuda de un transistor.
- Una celda (un bit) se implementa con un solo transistor $=>$ máxima capacidad de almacenamiento por CI.
- Ese transistor está generalmente en estado de Corte. Consume mínima energía.

Memorias dinámicas



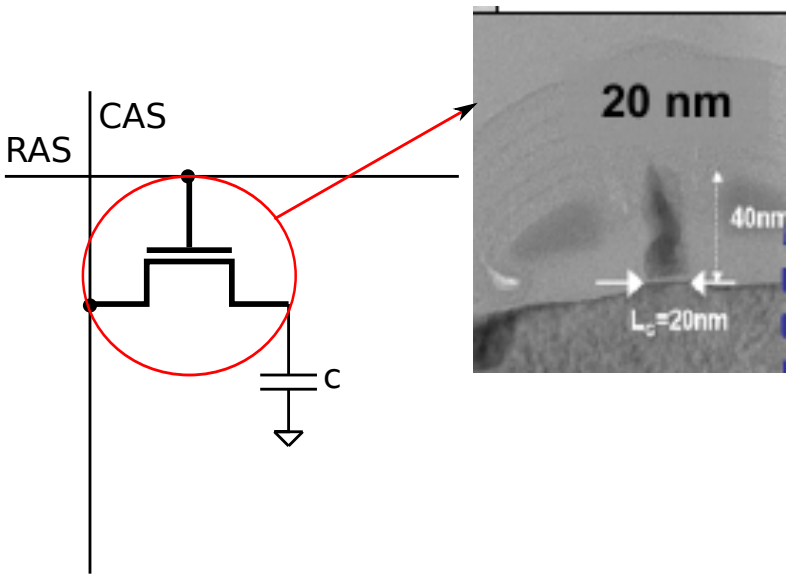
- Almacena la información en forma de estado de carga en un capacitor y la sostiene durante un breve lapso con la ayuda de un transistor.

Memorias dinámicas



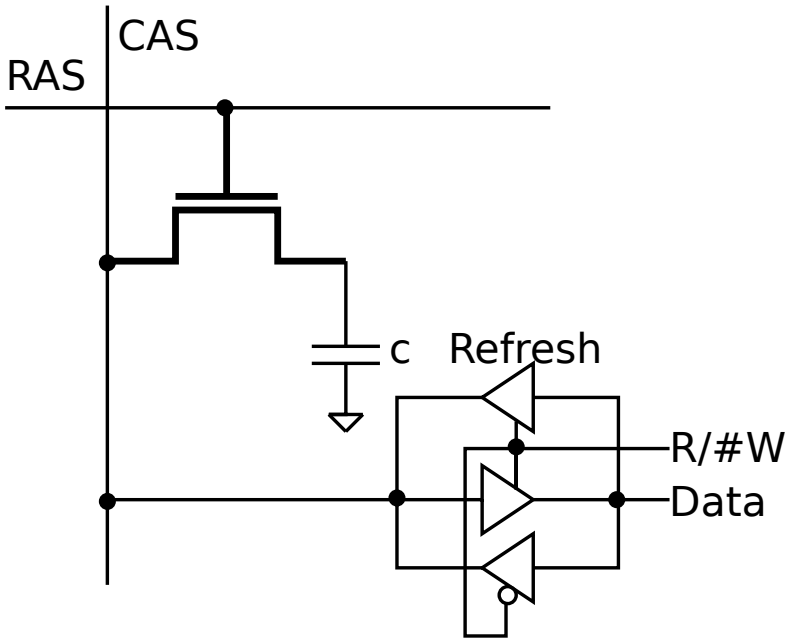
- Almacena la información en forma de estado de carga en un capacitor y la sostiene durante un breve lapso con la ayuda de un transistor.
- Una celda (un bit) se implementa con un solo transistor $=>$ máxima capacidad de almacenamiento por CI.

Memorias dinámicas



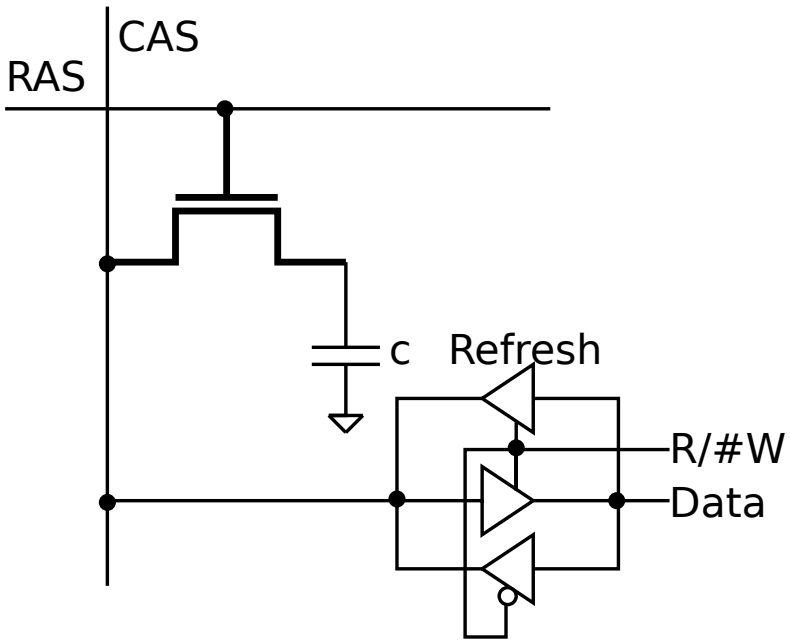
- Almacena la información en forma de estado de carga en un capacitor y la sostiene durante un breve lapso con la ayuda de un transistor.
- Una celda (un bit) se implementa con un solo transistor $=>$ máxima capacidad de almacenamiento por CI.
- Ese transistor está generalmente en estado de Corte. Consume mínima energía.

Memorias dinámicas



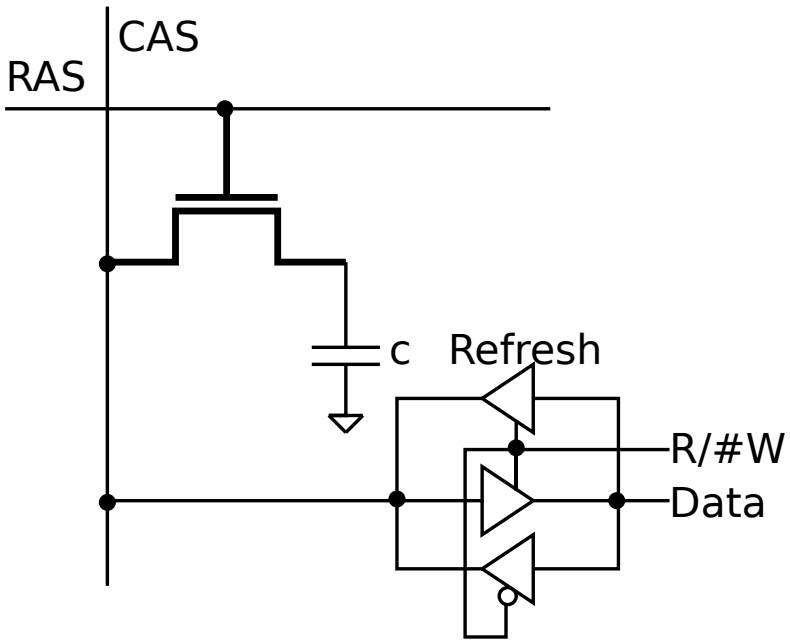
- Al leer el bit, se descarga la capacidad (lectura destructiva).

Memorias dinámicas



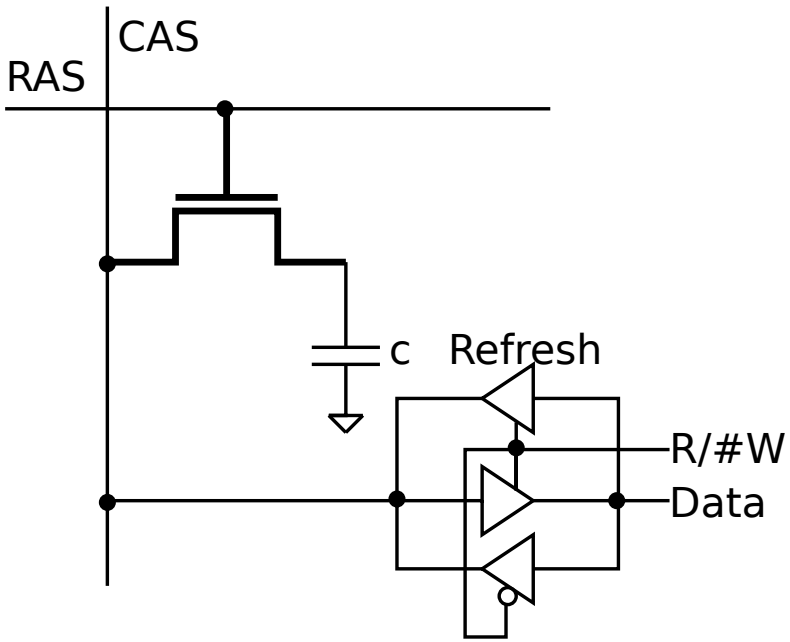
- Al leer el bit, se descarga la capacidad (lectura destructiva).
- **Necesita regenerar la carga** cada vez que se la lee.

Memorias dinámicas



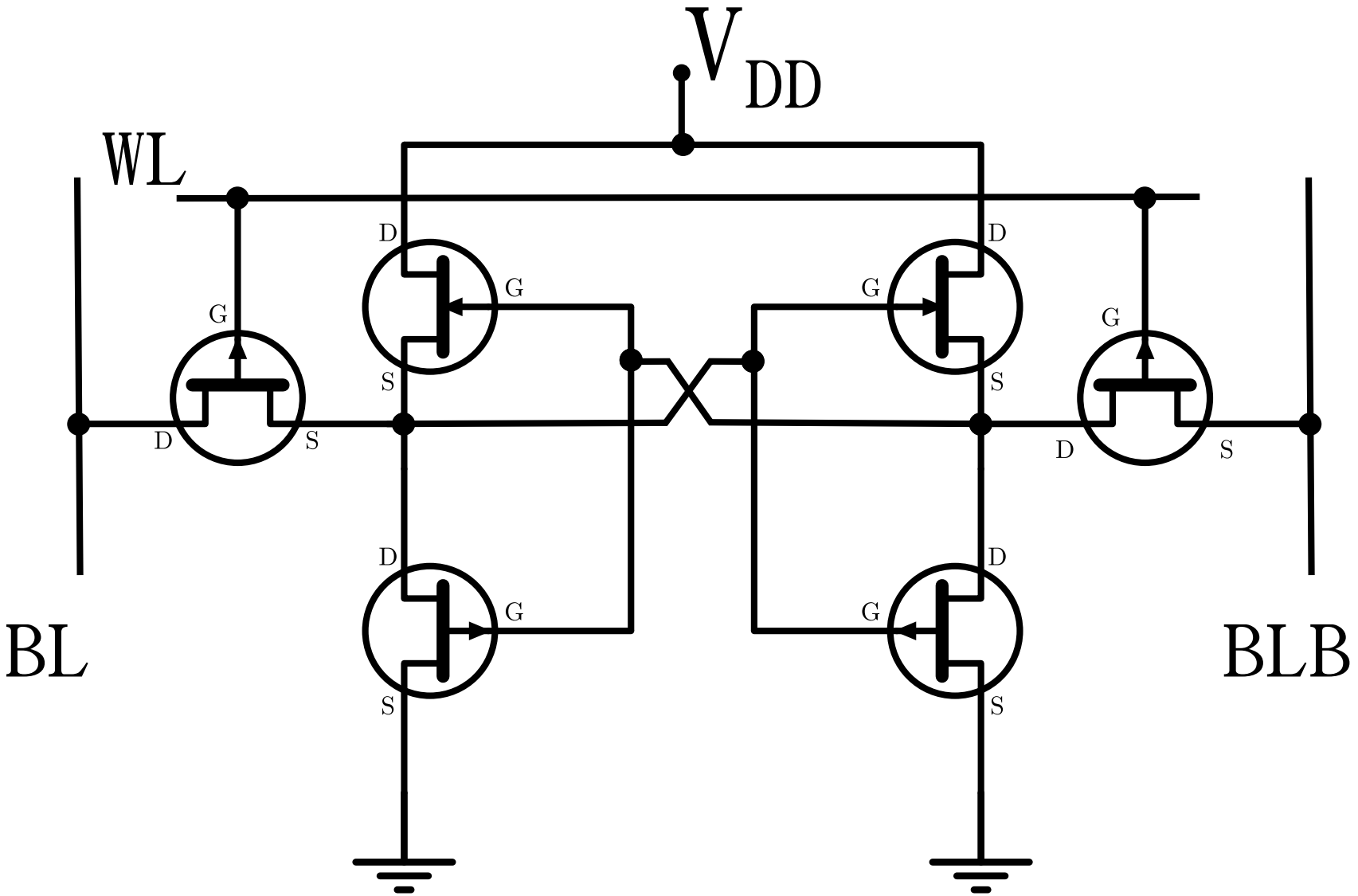
- Al leer el bit, se descarga la capacidad (lectura destructiva).
- **Necesita regenerar la carga** cada vez que se la lee.
- Esta operación se realiza por realimentación mediante buffers.

Memorias dinámicas



- Al leer el bit, se descarga la capacidad (lectura destructiva).
- **Necesita regenerar la carga** cada vez que se la lee.
- Esta operación se realiza por realimentación mediante buffers.
- Aumenta entonces el tiempo total que demanda el acceso de la celda, ya que no libera la operación hasta no haber repuesto el estado de carga del capacitor.

Memorias estáticas



Memorias estáticas

- Almacena la información en un biestable.

Memorias estáticas

- Almacena la información en un biestable.
- Una celda (un bit) se compone de seis transistores. Por lo tanto tiene menor capacidad de almacenamiento por CI.

Memorias estáticas

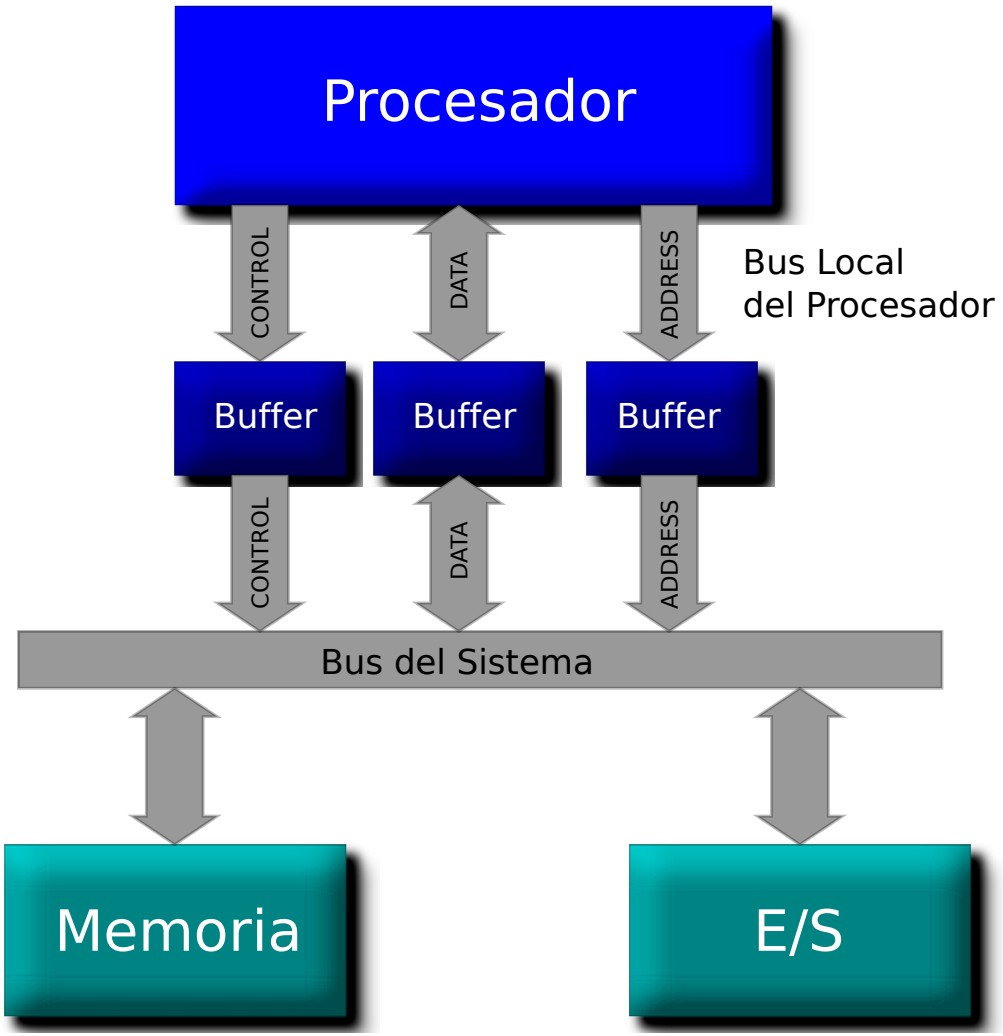
- Almacena la información en un biestable.
- Una celda (un bit) se compone de seis transistores. Por lo tanto tiene menor capacidad de almacenamiento por CI.
- Tres de los seis transistores están saturados (conducen la máxima corriente posible en forma permanente) y los otros tres al corte (conducen una corriente prácticamente insignificante, pero no nula). Esto genera mayor consumo de energía por celda.

Memorias estáticas

- Almacena la información en un biestable.
- Una celda (un bit) se compone de seis transistores. Por lo tanto tiene menor capacidad de almacenamiento por CI.
- Tres de los seis transistores están saturados (conducen la máxima corriente posible en forma permanente) y los otros tres al corte (conducen una corriente prácticamente insignificante, pero no nula). Esto genera mayor consumo de energía por celda.
- La lectura es directa y no destructiva por lo cual el tiempo de acceso es muy bajo en comparación con las memorias dinámicas. De hecho, luego de los registros del procesador, son el medio de almacenamiento de menor tiempo de acceso.

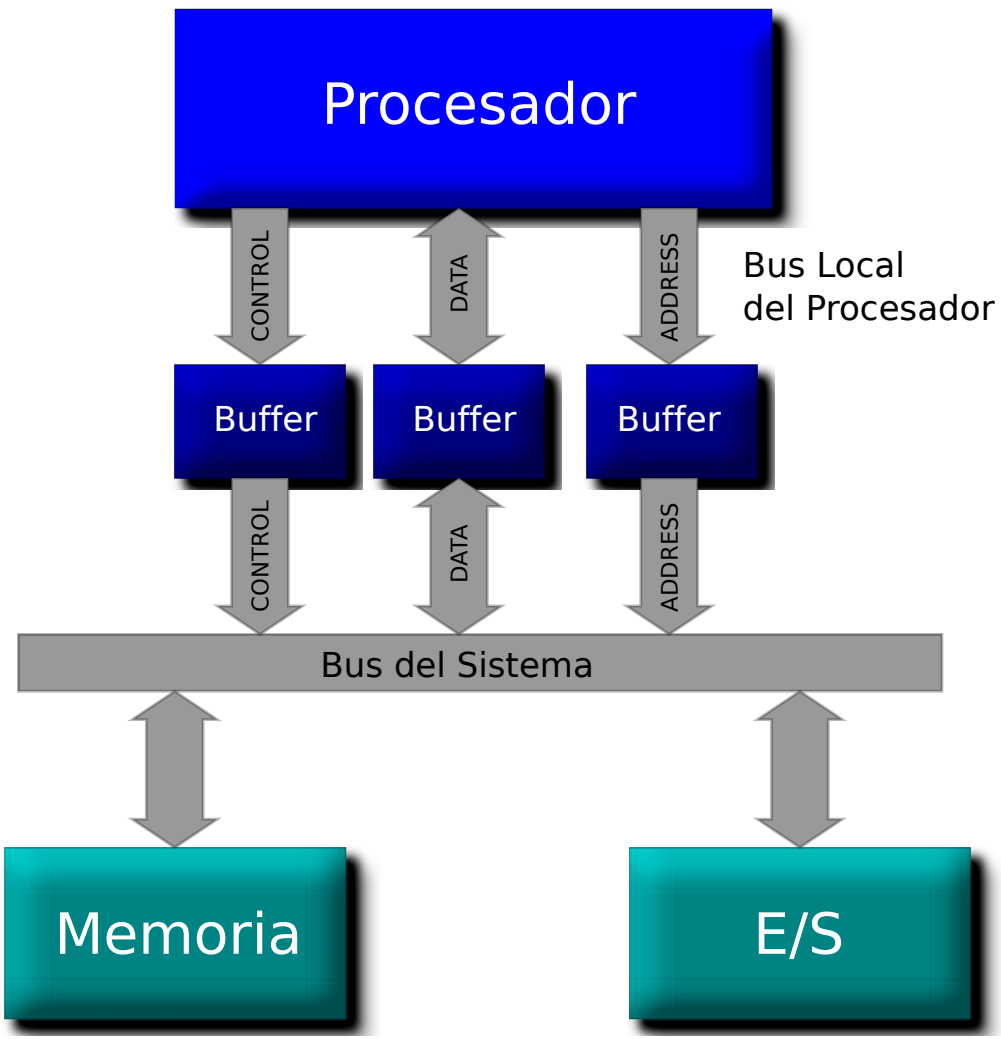
Temario

Conexión básica (Según Von Newmann)



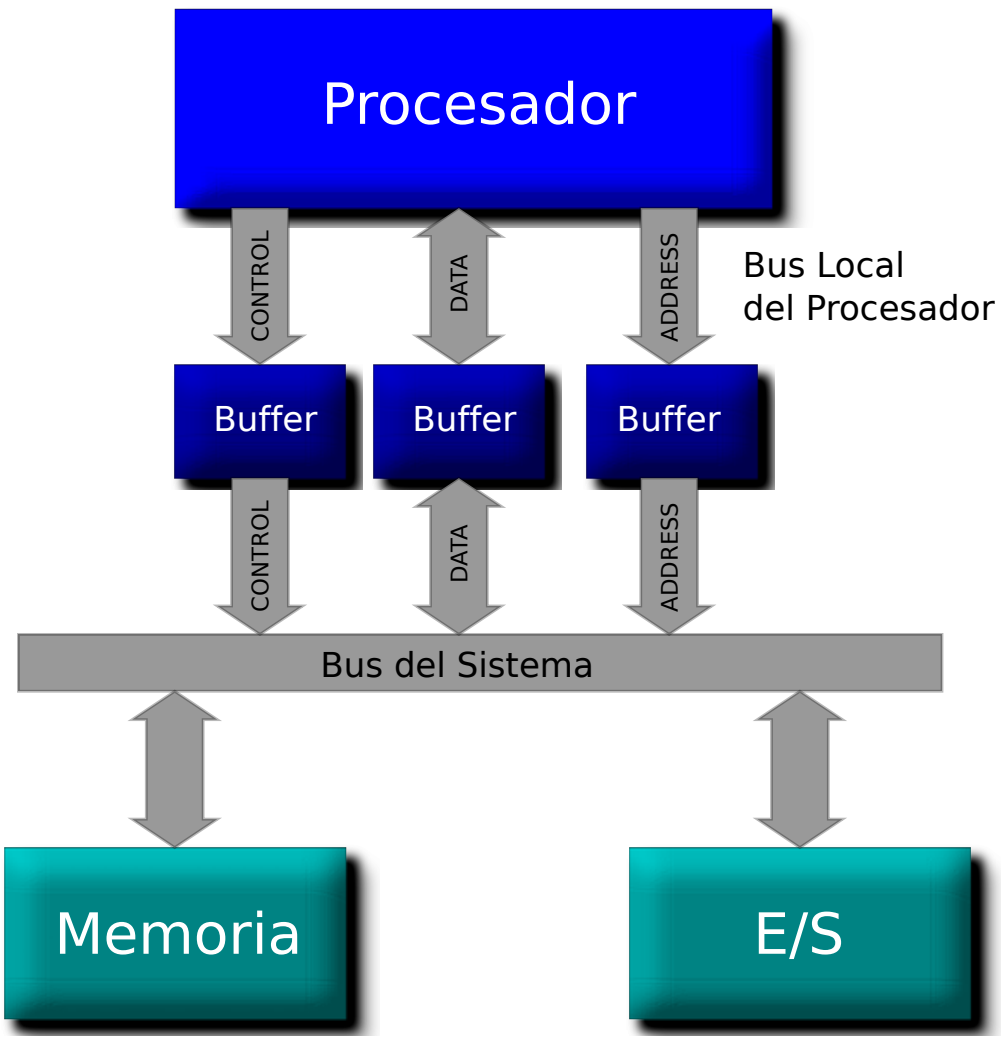
- Desde fines de los años 80, los procesadores desarrollaban velocidades muy superiores a los tiempos de acceso a memoria.

Conexión básica (Según Von Newmann)



- Desde fines de los años 80, los procesadores desarrollaban velocidades muy superiores a los tiempos de acceso a memoria.
- En este escenario, el procesador necesita generar wait states para esperar que la memoria esté lista (“READY”) para el acceso.

Conexión básica (Según Von Newmann)



- Desde fines de los años 80, los procesadores desarrollaban velocidades muy superiores a los tiempos de acceso a memoria.
- En este escenario, el procesador necesita generar wait states para esperar que la memoria esté lista (“READY”) para el acceso.
- ¿Tiene sentido lograr altos clocks en los procesadores si no puede aprovecharlos por tener que esperar (wait) a la memoria?

El problema...

El problema consiste en decidir que tipo de RAM usar en el sistema.
Hay dos opciones...

El problema...

El problema consiste en decidir que tipo de RAM usar en el sistema.
Hay dos opciones...

- **RAM dinámica (DRAM)**
- **RAM estática (SRAM)**

El problema...

El problema consiste en decidir que tipo de RAM usar en el sistema.
Hay dos opciones...

- **RAM dinámica (DRAM)**
  - Consumo mínimo.
- **RAM estática (SRAM)**
  - Alto consumo relativo.

El problema...

El problema consiste en decidir que tipo de RAM usar en el sistema.
Hay dos opciones...

- **RAM dinámica (DRAM)**
  - Consumo mínimo.
  - Capacidad de almacenamiento comparativamente alta.
- **RAM estática (SRAM)**
  - Alto consumo relativo.
  - Capacidad de almacenamiento comparativamente baja.

El problema...

El problema consiste en decidir que tipo de RAM usar en el sistema.
Hay dos opciones...

- **RAM dinámica (DRAM)**
  - Consumo mínimo.
  - Capacidad de almacenamiento comparativamente alta.
  - Costo por bit bajo.

- **RAM estática (SRAM)**
  - Alto consumo relativo.
  - Capacidad de almacenamiento comparativamente baja.
  - Costo por bit alto.

El problema...

El problema consiste en decidir que tipo de RAM usar en el sistema.
Hay dos opciones...

- **RAM dinámica (DRAM)**
  - Consumo mínimo.
  - Capacidad de almacenamiento comparativamente alta.
  - Costo por bit bajo.
  - Tiempo de acceso alto (lento), debido al circuito de regeneración de carga.
- **RAM estática (SRAM)**
  - Alto consumo relativo.
  - Capacidad de almacenamiento comparativamente baja.
  - Costo por bit alto.
  - Tiempo de acceso bajo (es mas rápida).

El problema...

El problema consiste en decidir que tipo de RAM usar en el sistema.
Hay dos opciones...

- **RAM dinámica (DRAM)**
  - Consumo mínimo.
  - Capacidad de almacenamiento comparativamente alta.
  - Costo por bit bajo.
  - Tiempo de acceso alto (lento), debido al circuito de regeneración de carga.
- **RAM estática (SRAM)**
  - Alto consumo relativo.
  - Capacidad de almacenamiento comparativamente baja.
  - Costo por bit alto.
  - Tiempo de acceso bajo (es mas rápida).

Conclusión:

Si construimos el banco de memoria utilizando RAM estática, el costo y el consumo de la computadora son altos. Si construimos el banco de memoria utilizando RAM dinámica, no aprovechamos la velocidad del procesador.

Temario

La solución. . . Memoria Cache

La solución. . . Memoria Cache

- Se trata de un banco de SRAM de muy alta velocidad, que contiene una copia de los datos e instrucciones que están en memoria principal.

La solución. . . Memoria Cache

- Se trata de un banco de SRAM de muy alta velocidad, que contiene una copia de los datos e instrucciones que están en memoria principal.
- El arte consiste en que esta copia esté disponible justo cuando el procesador la necesita permitiéndole acceder a esos ítems sin recurrir a wait states.

La solución. . . Memoria Cache

- Se trata de un banco de SRAM de muy alta velocidad, que contiene una copia de los datos e instrucciones que están en memoria principal.
- El arte consiste en que esta copia esté disponible justo cuando el procesador la necesita permitiéndole acceder a esos ítems sin recurrir a wait states.
- Combinada con una gran cantidad de memoria DRAM, para almacenar el resto de códigos y datos, resuelve el problema mediante una solución de compromiso típica.

La solución. . . Memoria Cache

- Se trata de un banco de SRAM de muy alta velocidad, que contiene una copia de los datos e instrucciones que están en memoria principal.
- El arte consiste en que esta copia esté disponible justo cuando el procesador la necesita permitiéndole acceder a esos ítems sin recurrir a wait states.
- Combinada con una gran cantidad de memoria DRAM, para almacenar el resto de códigos y datos, resuelve el problema mediante una solución de compromiso típica.
- Requiere de hardware adicional que asegure que este pequeño banco de memoria cache contenga los datos e instrucciones mas frecuentemente utilizados por el procesador.

Características y métricas

El tamaño del banco de memoria cache debe ser:

1. Suficientemente grande para que el procesador resuelva la mayor cantidad posible de búsquedas de código y datos en esta memoria asegurando una alta performance.
2. Suficientemente pequeña para no afectar el consumo ni el costo del sistema.

Hit cuando se accede a un ítem (dato o código) y éste *se encuentra* en la memoria cache

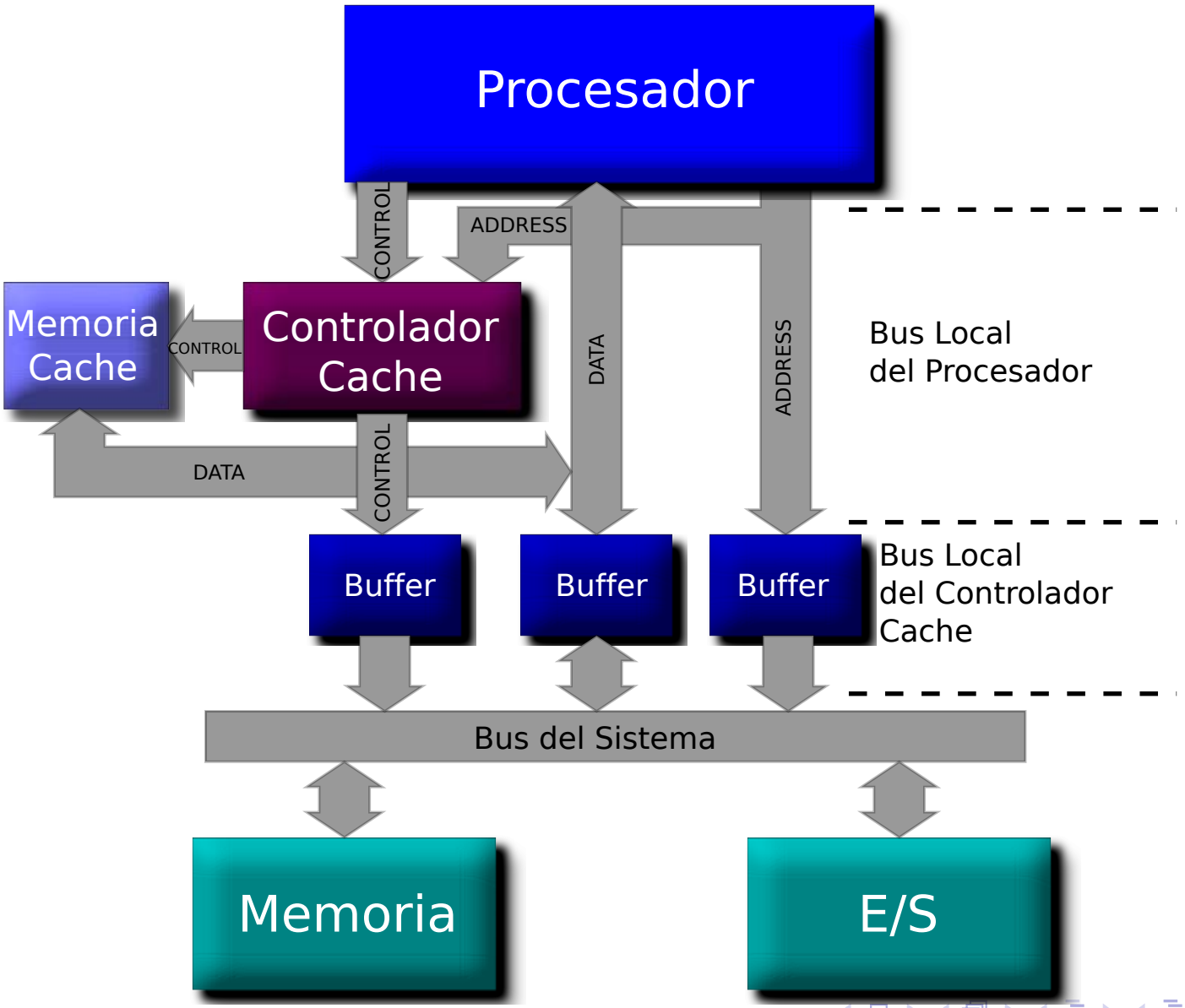
Miss cuando se accede a un ítem (dato o código) y éste *no se encuentra* en la memoria cache

hit rate $hitrate = \frac{Cantidad\ de\ Accesos\ con\ hit}{Cantidad\ de\ Accesos\ Totales}$

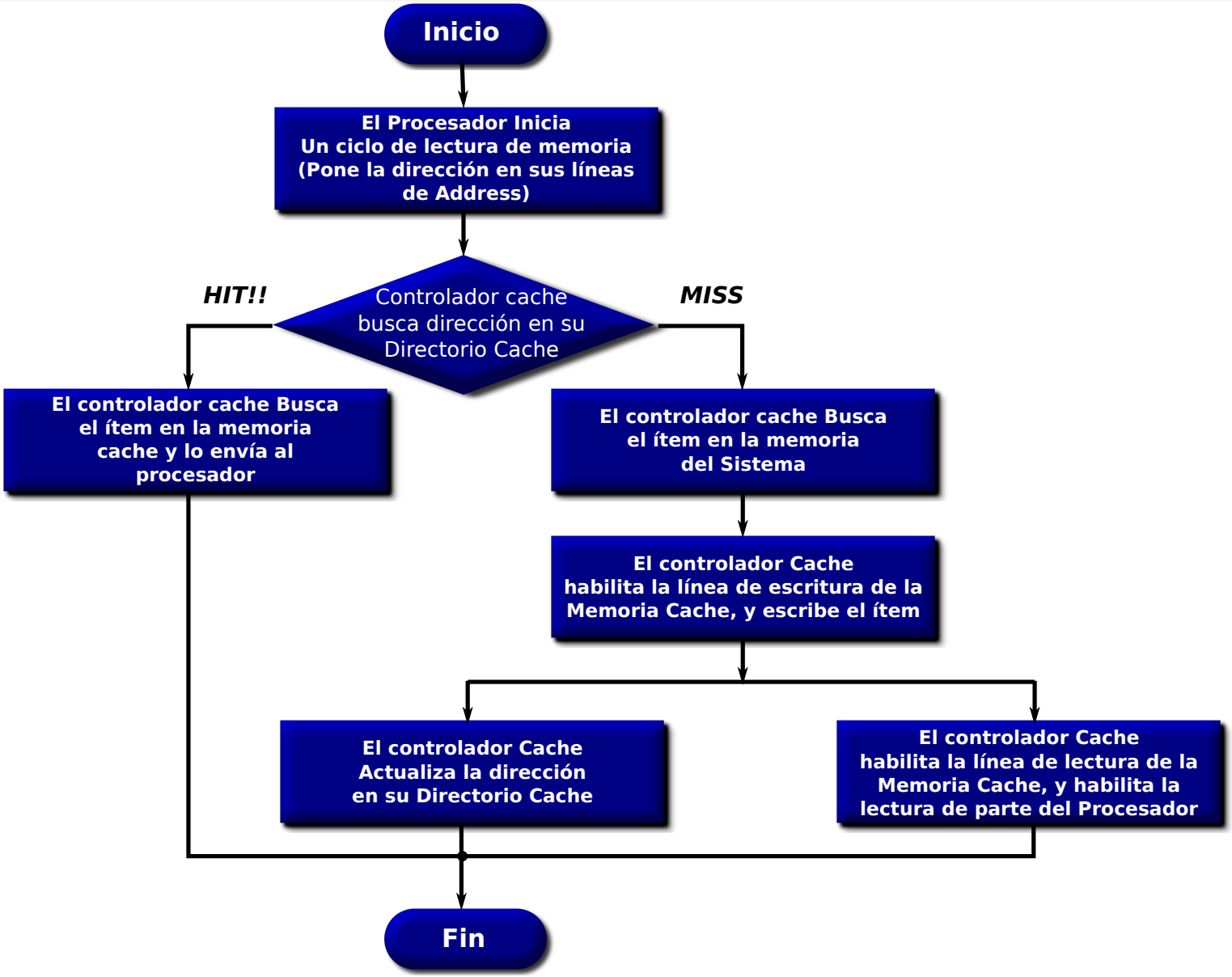
Se espera un hit rate lo mas alto posible

Temario

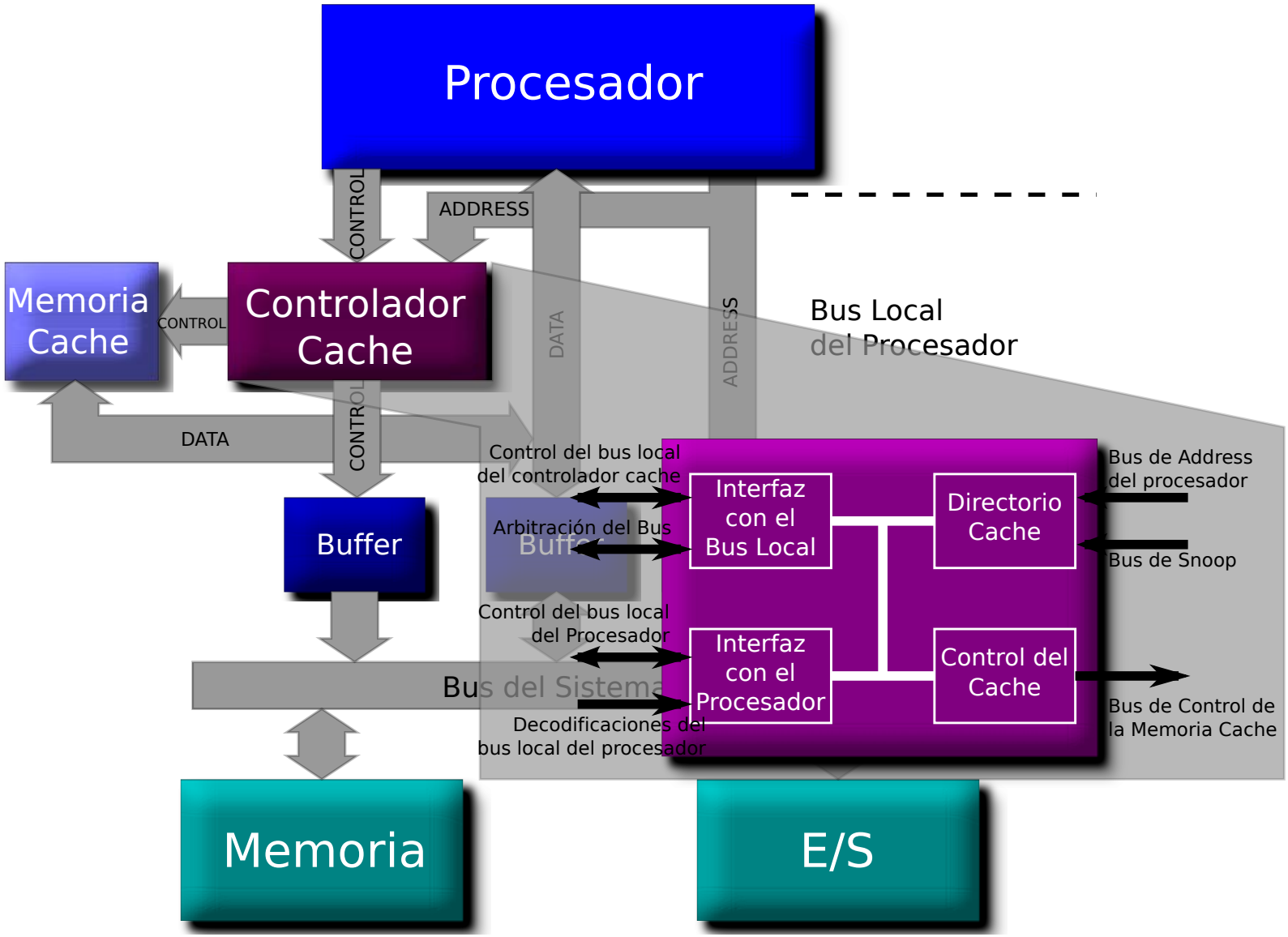
Subsistema Cache de Hardware



Operación de acceso a memoria para lectura



El Controlador Cache



Temario

Problemas derivados del scaling

- La disminución del ancho del gate de un CMOS fuerza la disminución de la tensión de umbral que contra-intuitivamente termina aumentando la corriente de leakage.

Problemas derivados del scaling

- La disminución del ancho del gate de un CMOS fuerza la disminución de la tensión de umbral que contra-intuitivamente termina aumentando la corriente de leakage.
- El estado lógico de un bit de memoria SRAM es finalmente una acumulación de cargas en un circuito capaz de mantener su estado. Si los transistores son mas pequeños, la cantidad de carga que compone un estado lógico es menor lo que la hace mas factible de ser alteradas por fenómenos externos (EMI Por ejemplo).

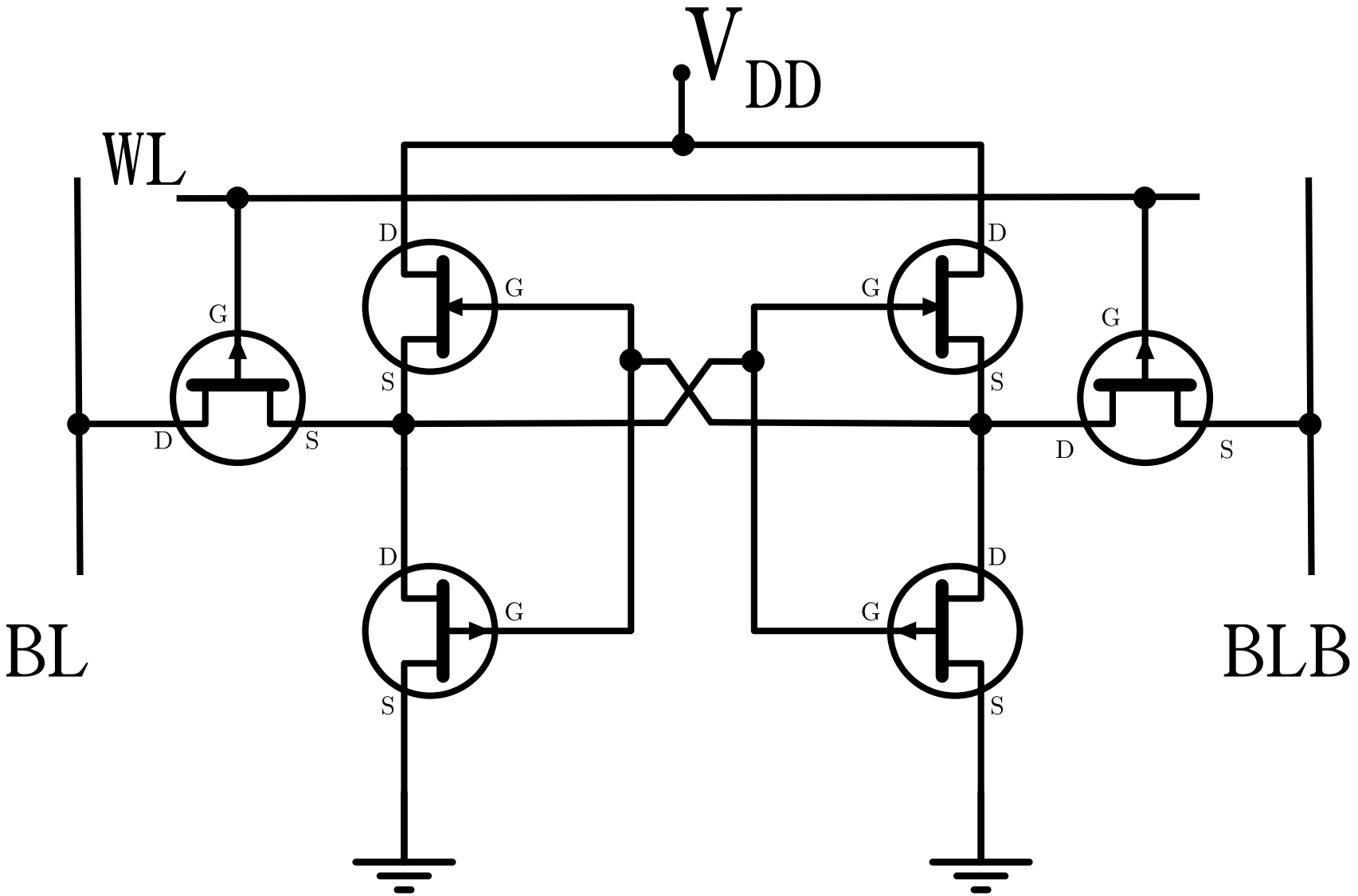
Problemas derivados del scaling

- La disminución del ancho del gate de un CMOS fuerza la disminución de la tensión de umbral que contra-intuitivamente termina aumentando la corriente de leakage.
- El estado lógico de un bit de memoria SRAM es finalmente una acumulación de cargas en un circuito capaz de mantener su estado. Si los transistores son mas pequeños, la cantidad de carga que compone un estado lógico es menor lo que la hace mas factible de ser alteradas por fenómenos externos (EMI Por ejemplo).
- El proceso de fabricación debe necesariamente variar cuando se reduce el transistor. Esto genera diferencias entre el modelo diseñado y el obtenido

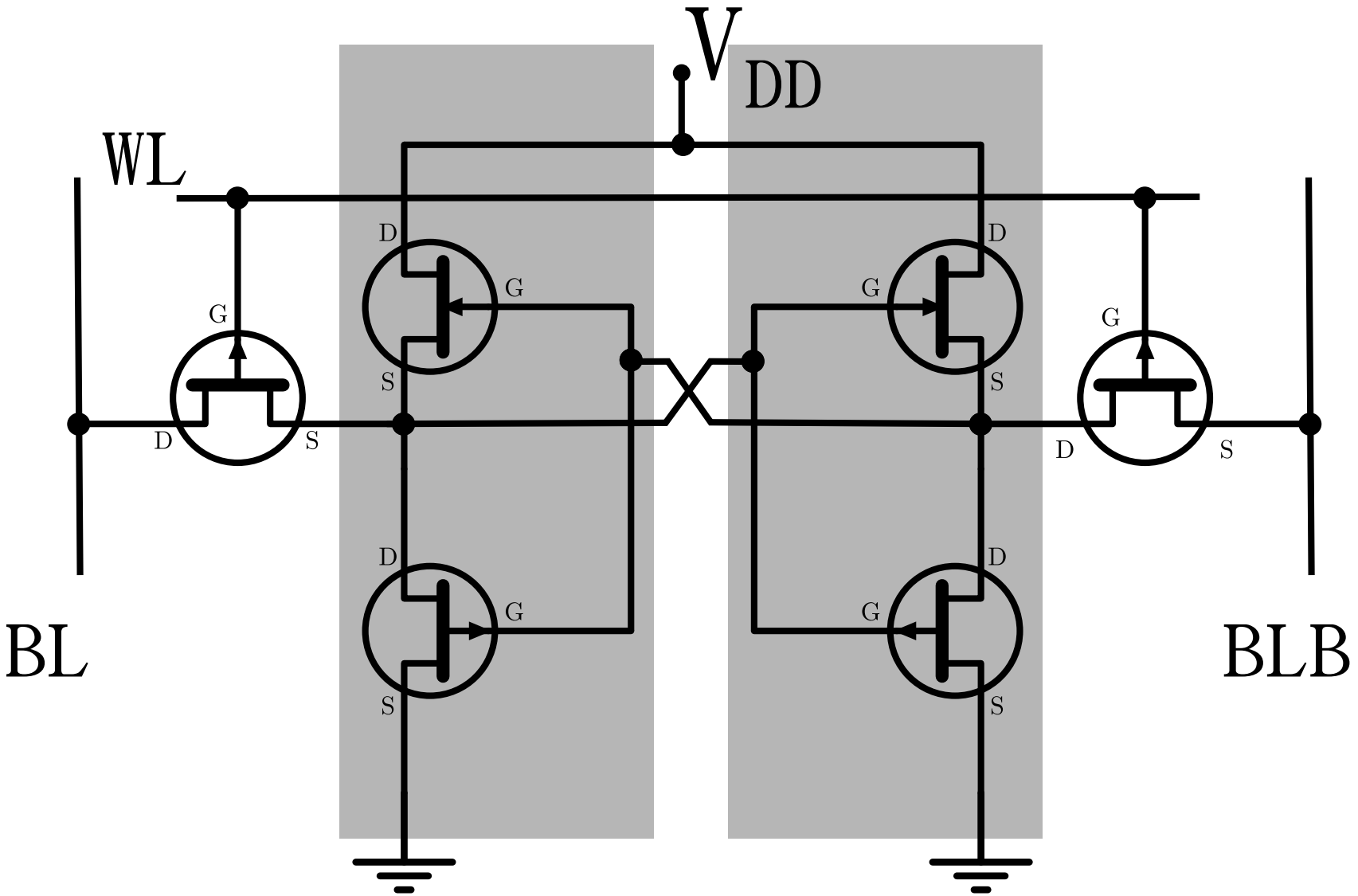
Problemas derivados del scaling

- La disminución del ancho del gate de un CMOS fuerza la disminución de la tensión de umbral que contra-intuitivamente termina aumentando la corriente de leakage.
- El estado lógico de un bit de memoria SRAM es finalmente una acumulación de cargas en un circuito capaz de mantener su estado. Si los transistores son mas pequeños, la cantidad de carga que compone un estado lógico es menor lo que la hace mas factible de ser alteradas por fenómenos externos (EMI Por ejemplo).
- El proceso de fabricación debe necesariamente variar cuando se reduce el transistor. Esto genera diferencias entre el modelo diseñado y el obtenido
- Los caminos de conexión de transistores (wires), comienzan a ser significativos en términos de delay y de disipación.

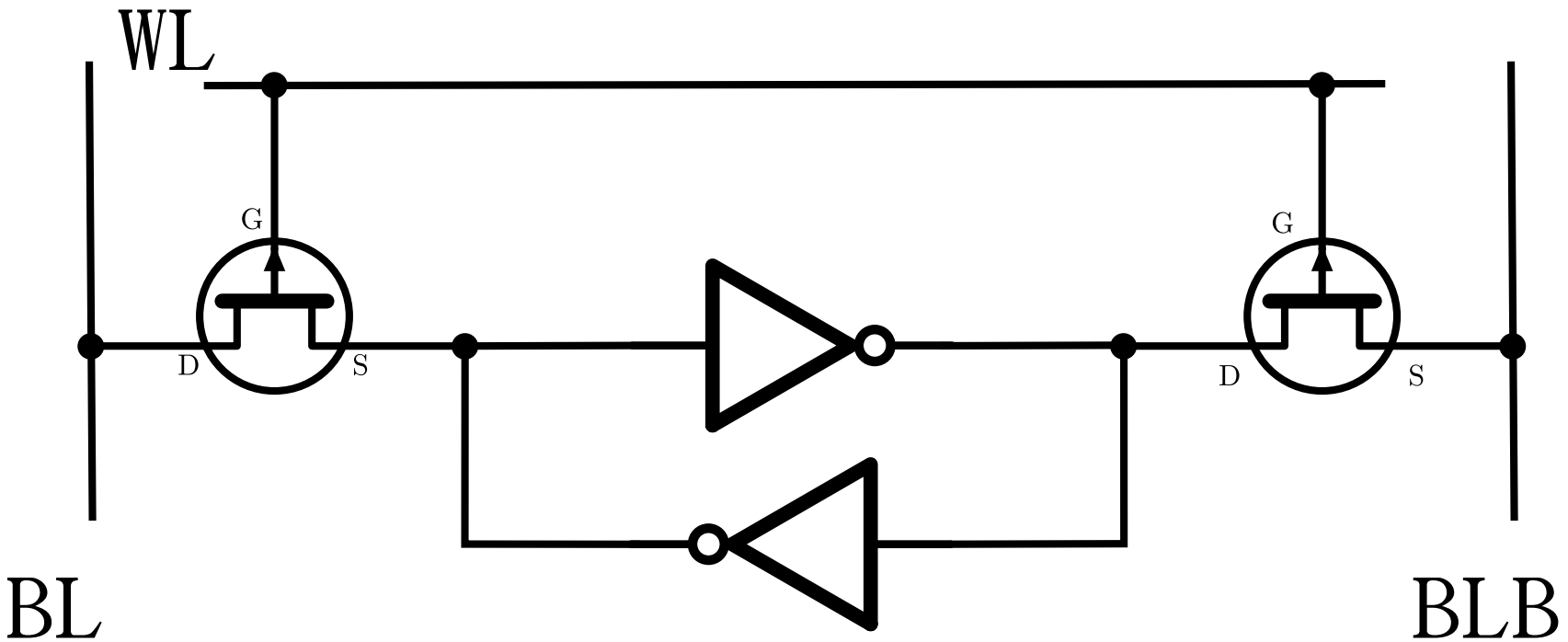
Celda de un bit de SRAM



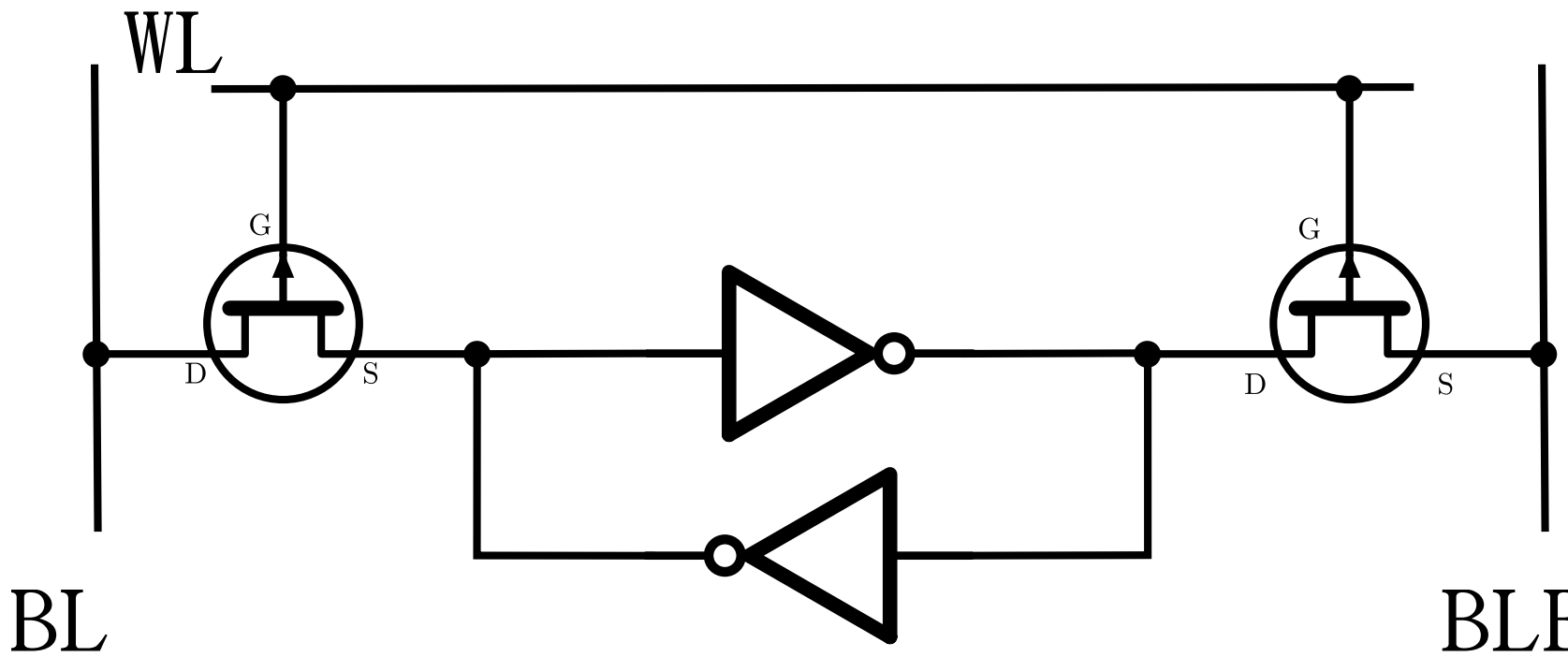
Celda de un bit de SRAM



Celda de un bit de SRAM

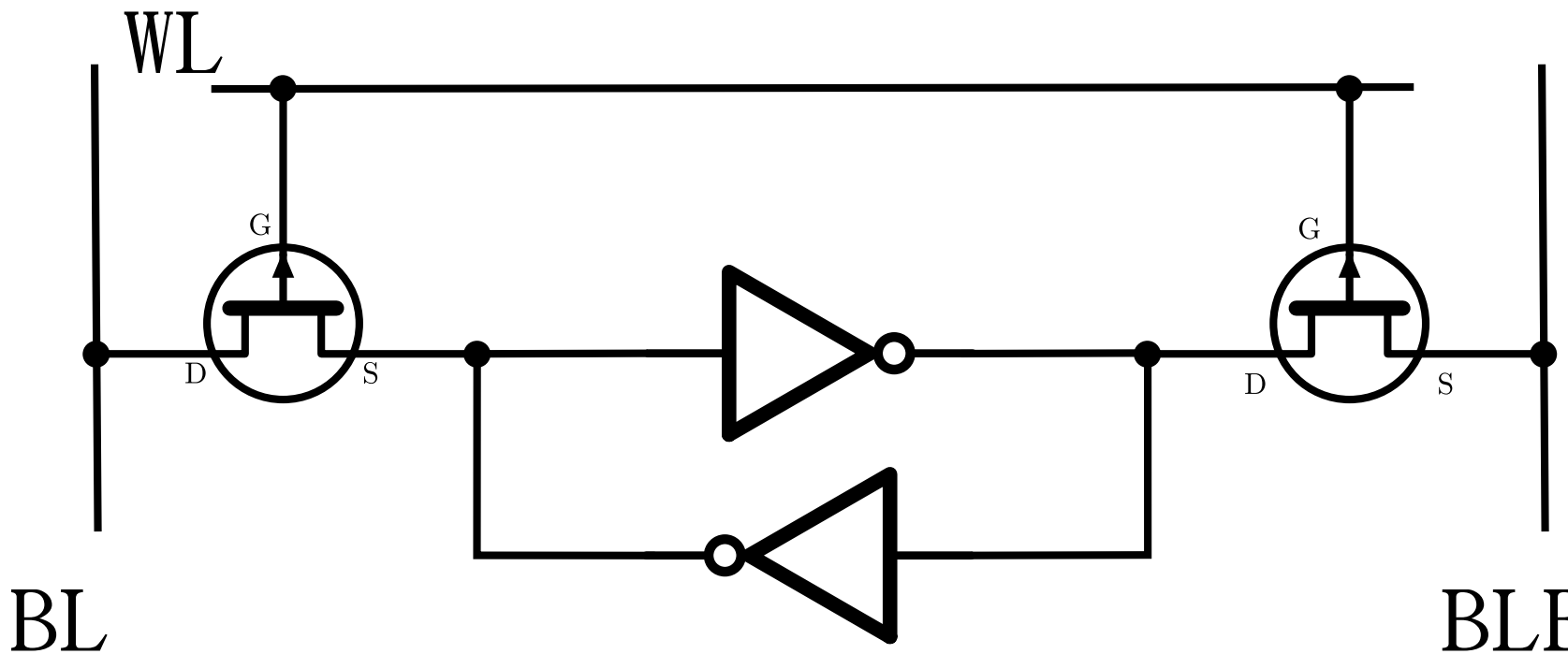


Celda de un bit de SRAM



La lectura del bit se realiza activando **WL**, y detectando la tensión diferencial entre el par de Líneas de Bit (**BL** y **BLB**), inicialmente precargadas para entregar un **'1'**.

Celda de un bit de SRAM



La escritura del bit se realiza activando **WL**, y colocando una tensión diferencial entre el par de Líneas de Bit (**BL** y **BLB**) proveniente de una fuente externa que fuerce el nuevo estado en el biestable.

Temario

Decodificación de Direcciones

Decodificación de Direcciones

- La decodificación de direcciones es un proceso sencillo que consiste en tomar el valor lógico de una dirección de memoria y a partir de éste valor y las señales de control del bus activar las señales lógicas necesarias para realizar la operación solicitada en la posición de memoria indicada.

Decodificación de Direcciones

- La decodificación de direcciones es un proceso sencillo que consiste en tomar el valor lógico de una dirección de memoria y a partir de éste valor y las señales de control del bus activar las señales lógicas necesarias para realizar la operación solicitada en la posición de memoria indicada.
- La forma mas simple de implementarla es inyectando la dirección de n bits en un decodificador de n a 2^n.

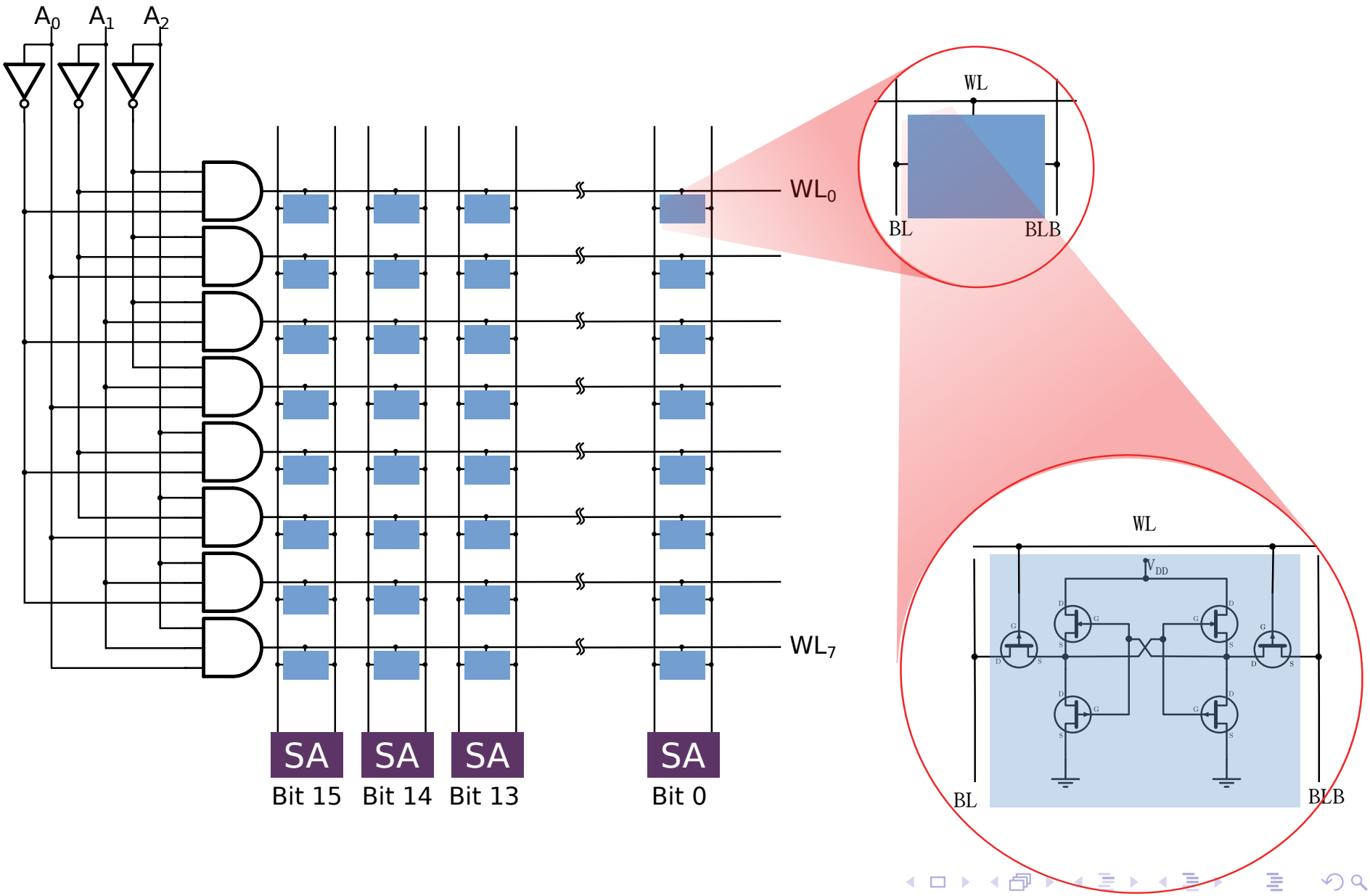
Decodificación de Direcciones

- La decodificación de direcciones es un proceso sencillo que consiste en tomar el valor lógico de una dirección de memoria y a partir de éste valor y las señales de control del bus activar las señales lógicas necesarias para realizar la operación solicitada en la posición de memoria indicada.
- La forma mas simple de implementarla es inyectando la dirección de n bits en un decodificador de n a 2^n.
- Esto involucra una operación **AND** sobre los valores posibles de entrada cuya salida active solo la `WL` correspondiente a la celda direccionada.

Decodificación de Direcciones

- La decodificación de direcciones es un proceso sencillo que consiste en tomar el valor lógico de una dirección de memoria y a partir de éste valor y las señales de control del bus activar las señales lógicas necesarias para realizar la operación solicitada en la posición de memoria indicada.
- La forma mas simple de implementarla es inyectando la dirección de n bits en un decodificador de n a 2^n.
- Esto involucra una operación **AND** sobre los valores posibles de entrada cuya salida active solo la `WL` correspondiente a la celda direccionada.
- A continuación un caso sencillo para un bus de address de 3 líneas, A_2, A_1, A_0

Decodificación de Direcciones



Decodificación de Direcciones

Decodificación de Direcciones

- La principal preocupación de diseño son los requerimientos de Fan-in y Fan-out en situaciones reales en donde la matriz representada en el gráfico anterior escala geométricamente confirme aumentan las lineas de Address.

Decodificación de Direcciones

- La principal preocupación de diseño son los requerimientos de Fan-in y Fan-out en situaciones reales en donde la matriz representada en el gráfico anterior escala geométricamente confirme aumentan las lineas de Address.
- Esto impide por ineficiente un diseño con un solo nivel de compuertas AND.

Decodificación de Direcciones

- La principal preocupación de diseño son los requerimientos de Fan-in y Fan-out en situaciones reales en donde la matriz representada en el gráfico anterior escala geométricamente confirme aumentan las lineas de Address.
- Esto impide por ineficiente un diseño con un solo nivel de compuertas AND.
- Los decodificadores reales Se implementan con estructuras multinivel de compuertas AND.

Decodificación de Direcciones

- La principal preocupación de diseño son los requerimientos de Fan-in y Fan-out en situaciones reales en donde la matriz representada en el gráfico anterior escala geométricamente confirme aumentan las lineas de Address.
- Esto impide por ineficiente un diseño con un solo nivel de compuertas AND.
- Los decodificadores reales Se implementan con estructuras multinivel de compuertas AND.
- Es tan no trivial este tema que el diseño del decodificador interno es crítico tanto en el delay de escritura y lectura (tiempo de acceso) como en disipación de energía.

Predecoding

Predecoding

- Es la salida al problema descripto.

Predecoding

- Es la salida al problema descripto.
- Se basa en evitar recurrir a compuertas de alto Fan-in, debido a las enormes dificultades de diseño y a su baja eficiencia.

Predecoding

- Es la salida al problema descripto.
- Se basa en evitar recurrir a compuertas de alto Fan-in, debido a las enormes dificultades de diseño y a su baja eficiencia.
- Predecoding hace una decodificación en dos etapas.

Predecoding

- Es la salida al problema descripto.
- Se basa en evitar recurrir a compuertas de alto Fan-in, debido a las enormes dificultades de diseño y a su baja eficiencia.
- Predecoding hace una decodificación en dos etapas.
- La primera involucra la función AND sobre un grupo de entradas (no todas sino por ejemplo las mas significativas).

Predecoding

- Es la salida al problema descripto.
- Se basa en evitar recurrir a compuertas de alto Fan-in, debido a las enormes dificultades de diseño y a su baja eficiencia.
- Predecoding hace una decodificación en dos etapas.
- La primera involucra la función AND sobre un grupo de entradas (no todas sino por ejemplo las mas significativas).
- Las salidas de esta primer etapa se combinan con compuertas AND de bajo Fan-in que son las que van a generar la salida del decodificador, es decir las Wordlines que atacan las filas de la matriz de bits de SRAM

Predecoding

- Es la salida al problema descripto.
- Se basa en evitar recurrir a compuertas de alto Fan-in, debido a las enormes dificultades de diseño y a su baja eficiencia.
- Predecoding hace una decodificación en dos etapas.
- La primera involucra la función AND sobre un grupo de entradas (no todas sino por ejemplo las mas significativas).
- Las salidas de esta primer etapa se combinan con compuertas AND de bajo Fan-in que son las que van a generar la salida del decodificador, es decir las Wordlines que atacan las filas de la matriz de bits de SRAM
- Para comprenderlo, consideremos un caso de un decodificador de direcciones de 8 bits.

Predecoding

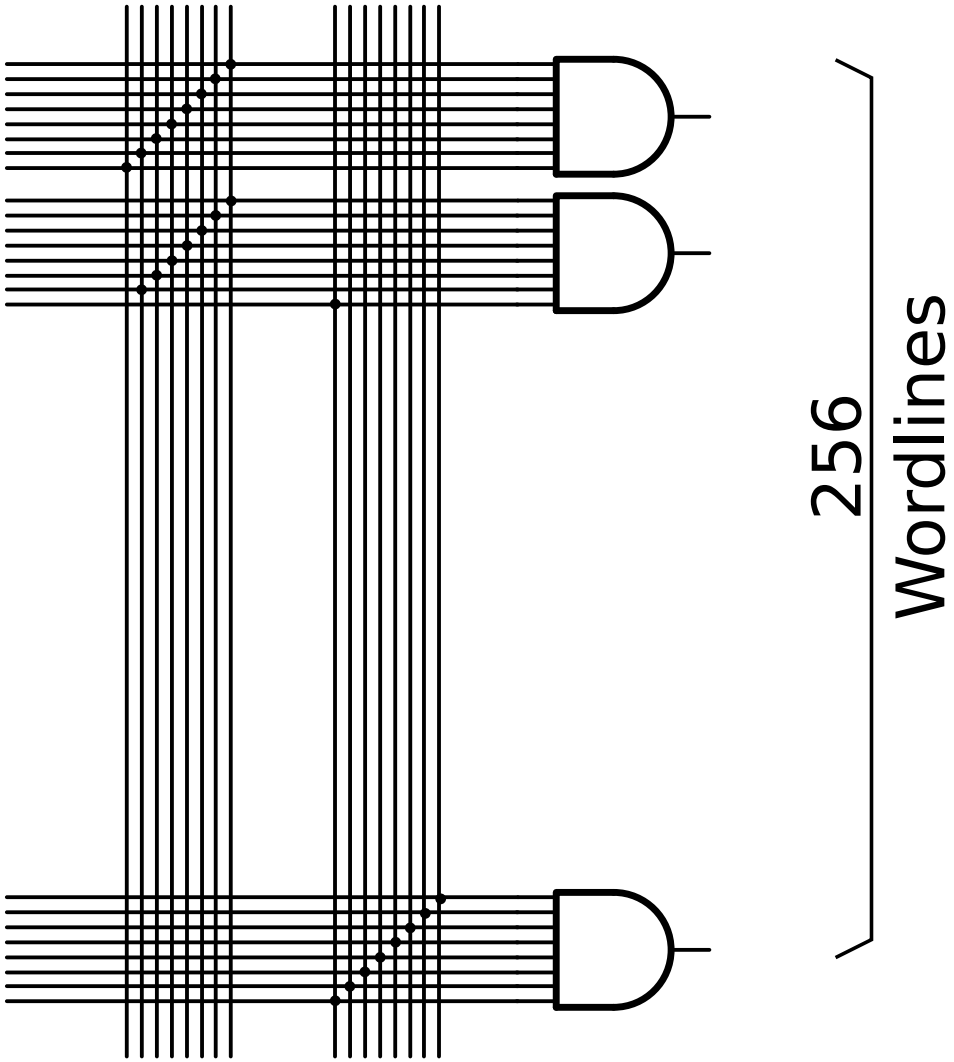
- Es la salida al problema descripto.
- Se basa en evitar recurrir a compuertas de alto Fan-in, debido a las enormes dificultades de diseño y a su baja eficiencia.
- Predecoding hace una decodificación en dos etapas.
- La primera involucra la función AND sobre un grupo de entradas (no todas sino por ejemplo las mas significativas).
- Las salidas de esta primer etapa se combinan con compuertas AND de bajo Fan-in que son las que van a generar la salida del decodificador, es decir las Wordlines que atacan las filas de la matriz de bits de SRAM
- Para comprenderlo, consideremos un caso de un decodificador de direcciones de 8 bits.
- Un approach simplista es usar 256 compuertas de 8 líneas de entrada.

Predecoding

- Es la salida al problema descripto.
- Se basa en evitar recurrir a compuertas de alto Fan-in, debido a las enormes dificultades de diseño y a su baja eficiencia.
- Predecoding hace una decodificación en dos etapas.
- La primera involucra la función AND sobre un grupo de entradas (no todas sino por ejemplo las mas significativas).
- Las salidas de esta primer etapa se combinan con compuertas AND de bajo Fan-in que son las que van a generar la salida del decodificador, es decir las Wordlines que atacan las filas de la matriz de bits de SRAM
- Para comprenderlo, consideremos un caso de un decodificador de direcciones de 8 bits.
- Un approach simplista es usar 256 compuertas de 8 líneas de entrada.
- El resultado es la siguiente matriz

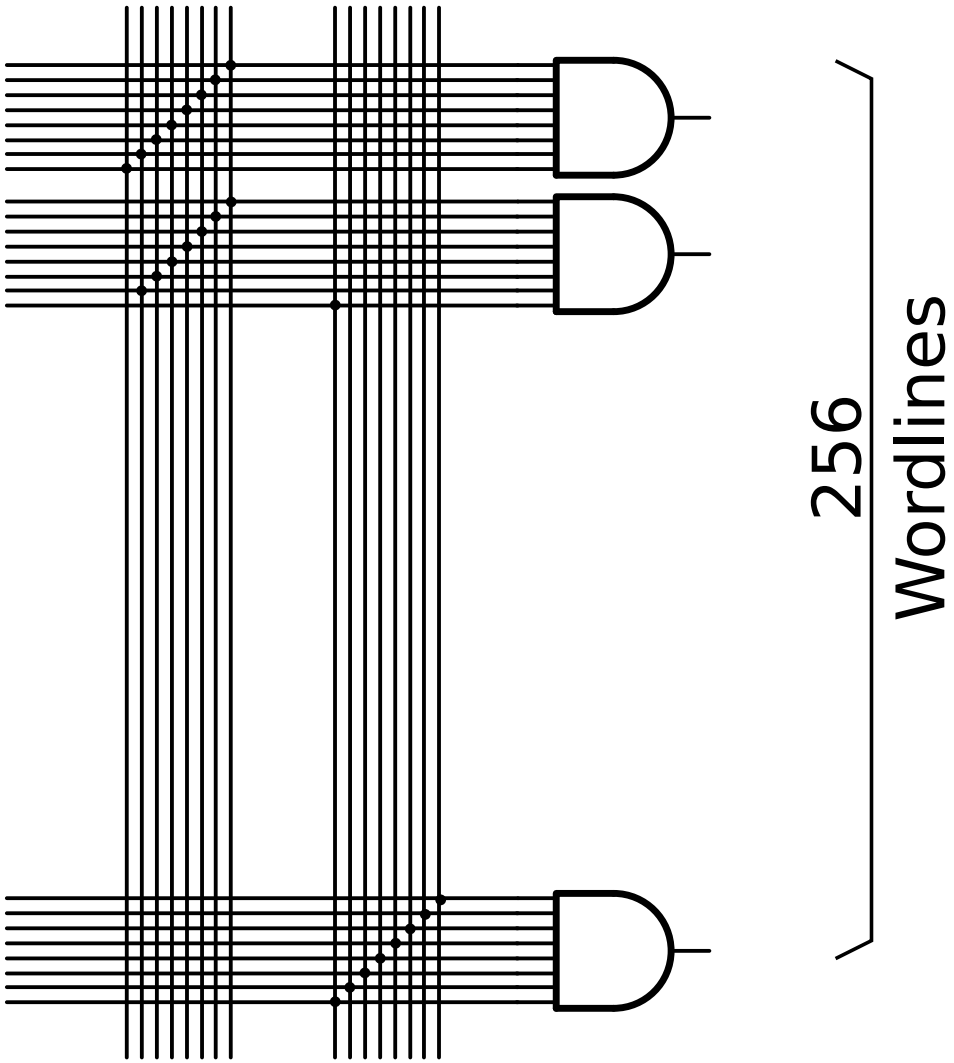
Predecoding

Dirección no invertida de 8 bits



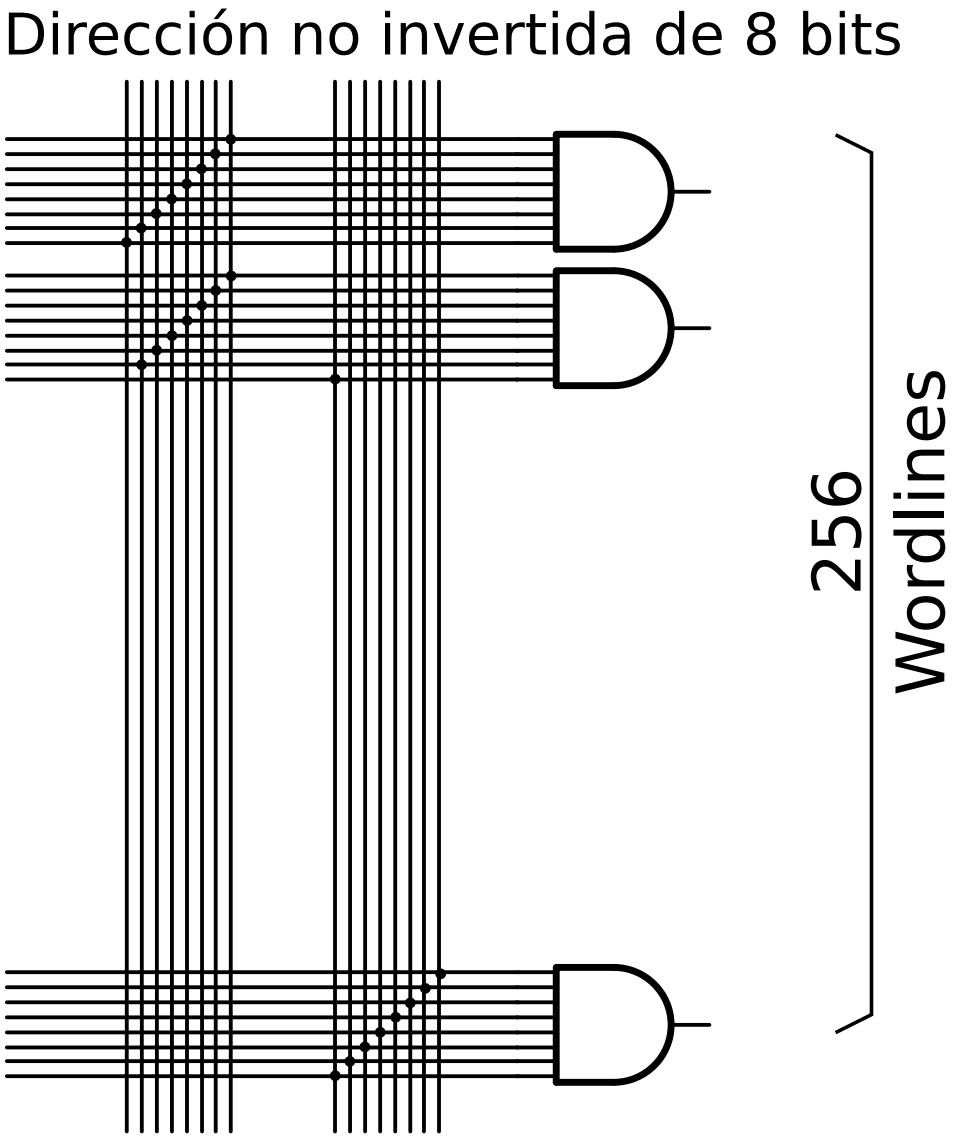
Predecoding

Dirección no invertida de 8 bits



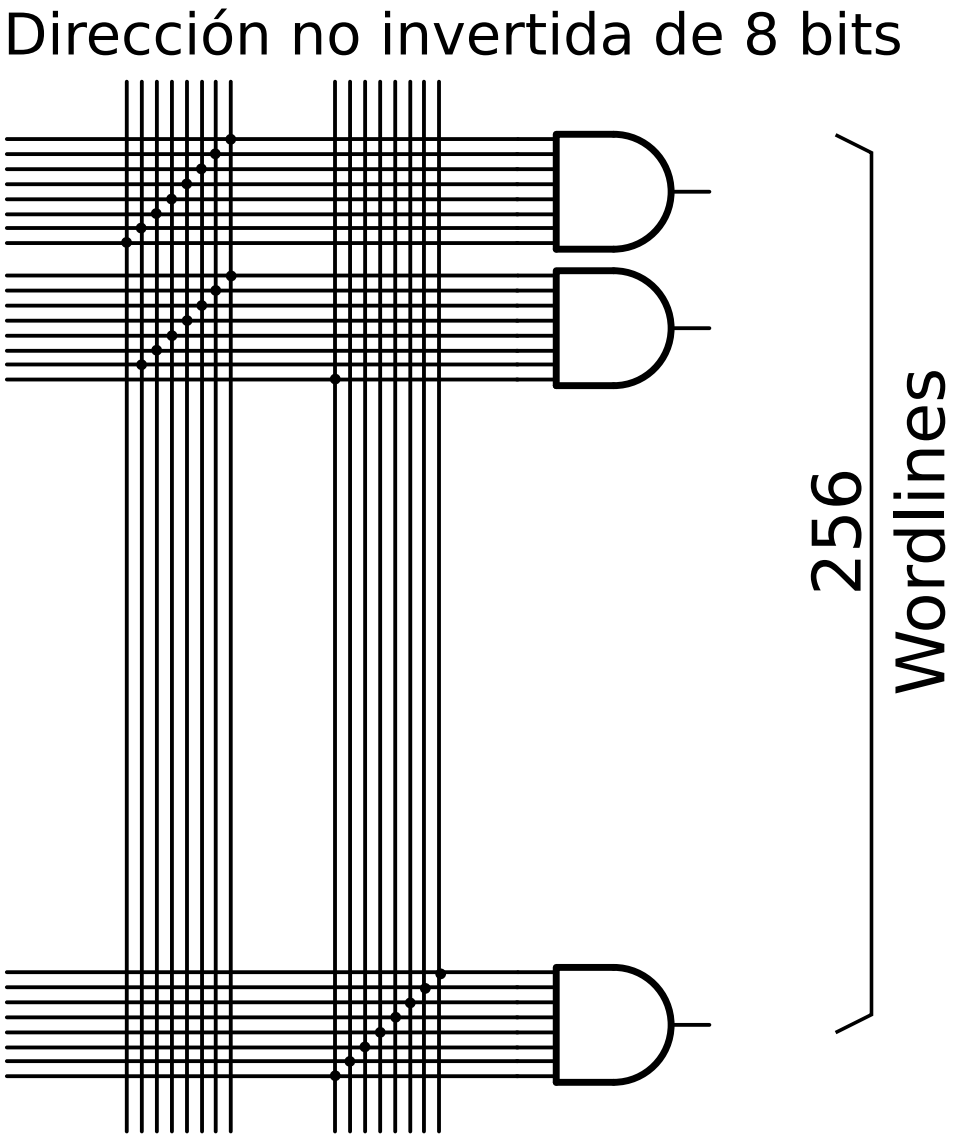
- Si bien la configuración lógica es correcta, la capacidad de entrada de las AND de 8 entradas es mucho mayor que la de una AND simple de dos entradas.

Predecoding



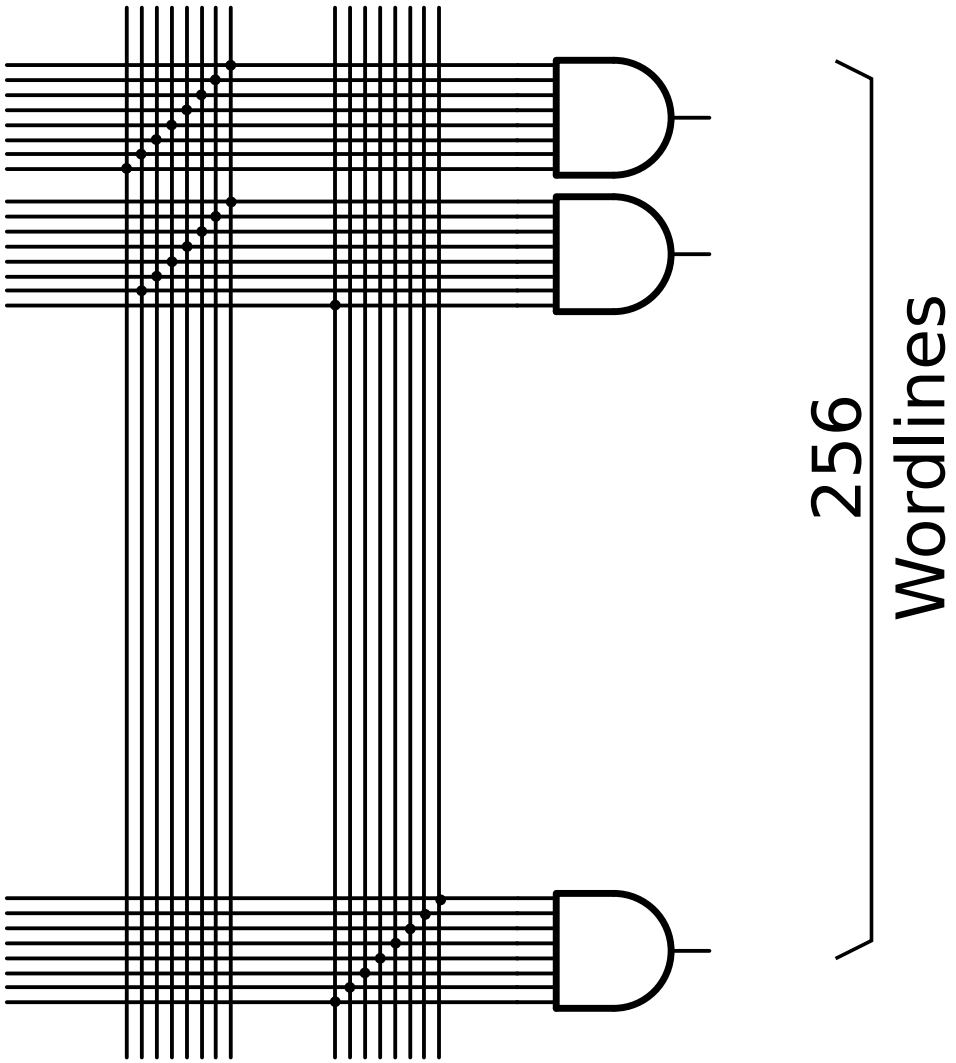
- Si bien la configuración lógica es correcta, la capacidad de entrada de las AND de 8 entradas es mucho mayor que la de una AND simple de dos entradas.
- Esto aumenta el delay (tiempo de acceso), la potencia disipada, y el área de *Si*.

Predecoding



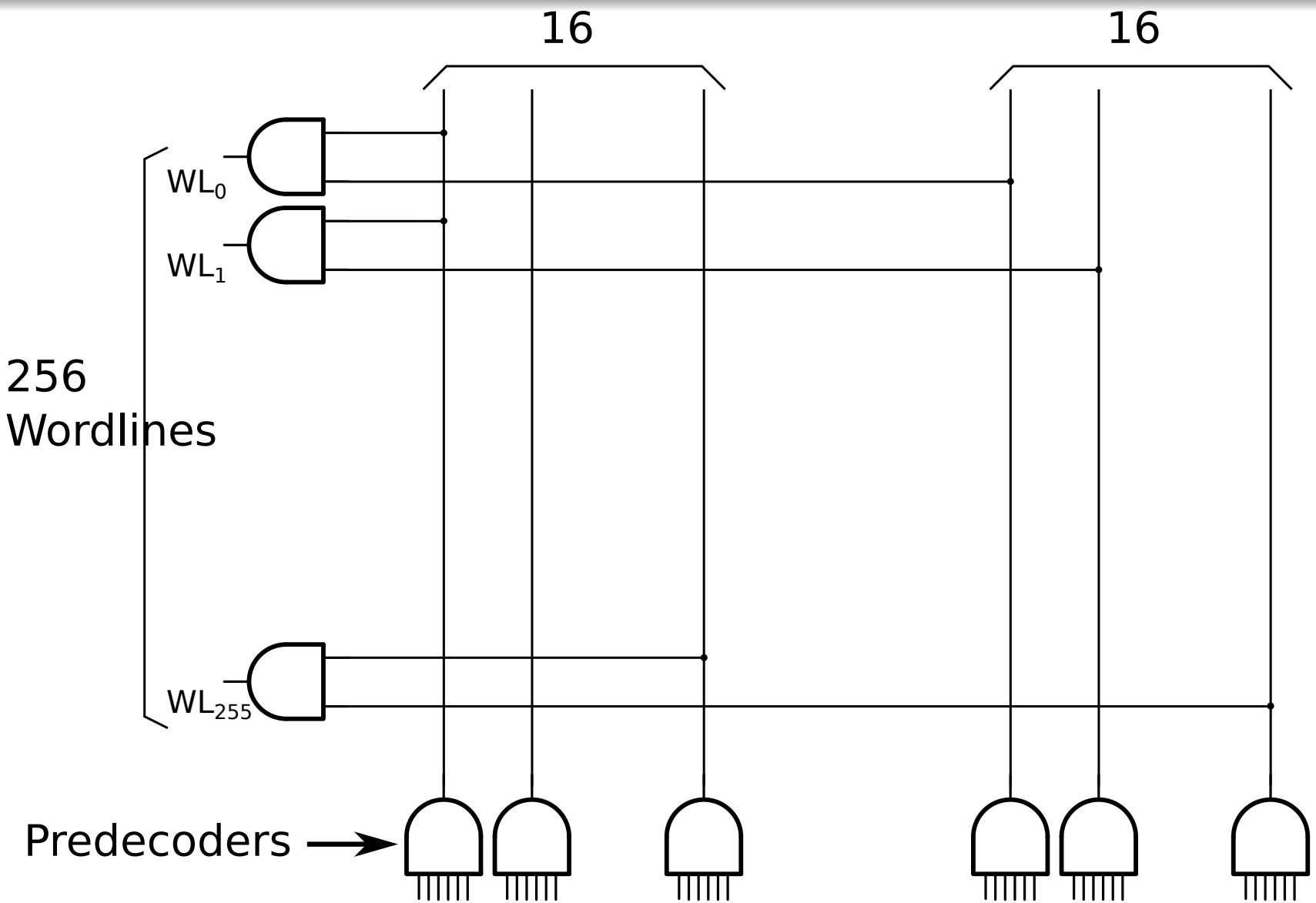
- Si bien la configuración lógica es correcta, la capacidad de entrada de las AND de 8 entradas es mucho mayor que la de una AND simple de dos entradas.
- Esto aumenta el delay (tiempo de acceso), la potencia disipada, y el área de *Si*.
- Tan solo imaginen éste problema llevado a buses reales con 32 lineas. Imposible

Predecoding



- Si bien la configuración lógica es correcta, la capacidad de entrada de las AND de 8 entradas es mucho mayor que la de una AND simple de dos entradas.
- Esto aumenta el delay (tiempo de acceso), la potencia disipada, y el área de *Si*.
- Tan solo imaginen éste problema llevado a buses reales con 32 lineas. Imposible
- En el siguiente slide hay una alternativa dividiendo las entradas en dos subsets de 4 bits.

Predecoding



Predecoding

Predecoding

- Cada subset de 4 líneas atada a 16 compuertas AND de 4 entradas, de modo que cada AND puede generar una salida activa única para cada combinación posible.

Predecoding

- Cada subset de 4 líneas atada a 16 compuertas AND de 4 entradas, de modo que cada AND puede generar una salida activa única para cada combinación posible.
- Luego utiliza 256 compuertas AND de dos entradas para generar las 256 wordlines.

Predecoding

- Cada subset de 4 líneas atada a 16 compuertas AND de 4 entradas, de modo que cada AND puede generar una salida activa única para cada combinación posible.
- Luego utiliza 256 compuertas AND de dos entradas para generar las 256 wordlines.
- Además de ser una lógica sencilla, Predecoding tiene numerosas ventajas frente al arreglo original.

Predecoding

- Cada subset de 4 líneas atada a 16 compuertas AND de 4 entradas, de modo que cada AND puede generar una salida activa única para cada combinación posible.
- Luego utiliza 256 compuertas AND de dos entradas para generar las 256 wordlines.
- Además de ser una lógica sencilla, Predecoding tiene numerosas ventajas frente al arreglo original.
- Las compuertas de dos entradas permiten lograr un circuito mucho mas rápido con menor disipación de potencia, que menos superficie que el primero.

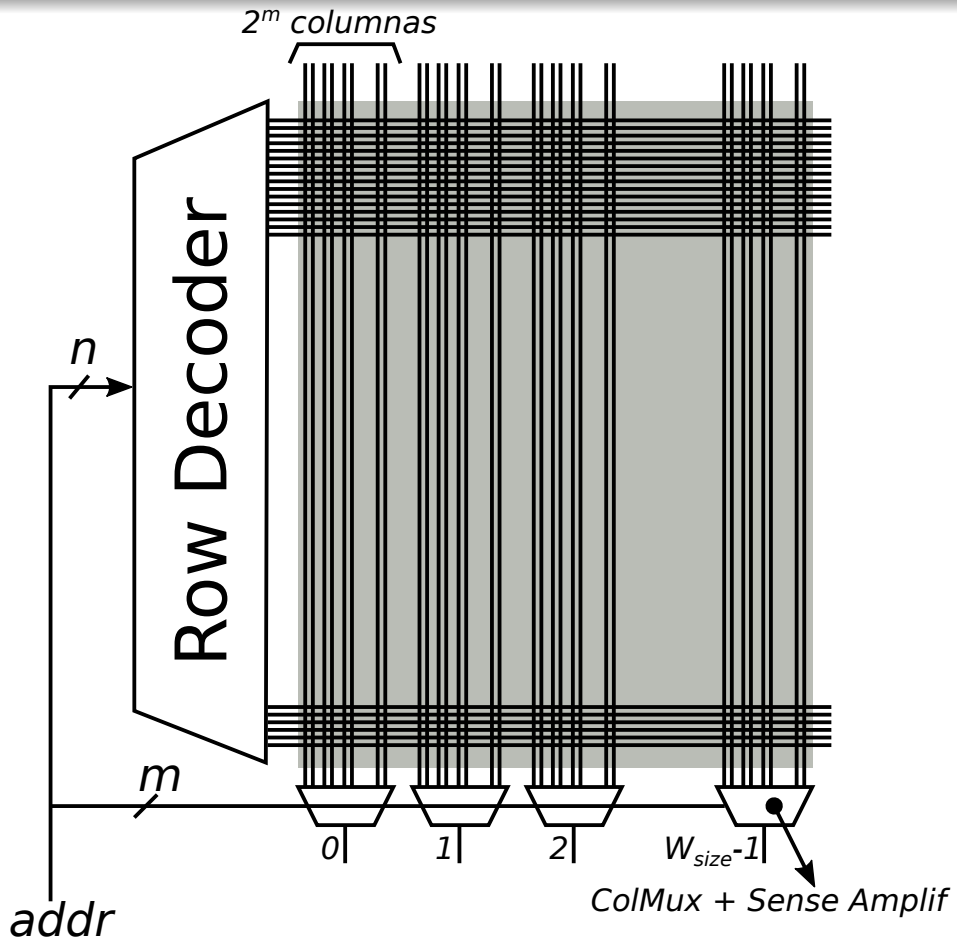
Predecoding

- Cada subset de 4 líneas atada a 16 compuertas AND de 4 entradas, de modo que cada AND puede generar una salida activa única para cada combinación posible.
- Luego utiliza 256 compuertas AND de dos entradas para generar las 256 wordlines.
- Además de ser una lógica sencilla, Predecoding tiene numerosas ventajas frente al arreglo original.
- Las compuertas de dos entradas permiten lograr un circuito mucho mas rápido con menor disipación de potencia, que menos superficie que el primero.
- Pero fundamentalmente es mucho mas escalable, aunque en memorias de mayor capacidad la cantidad de conductores se transforma en un limitante.

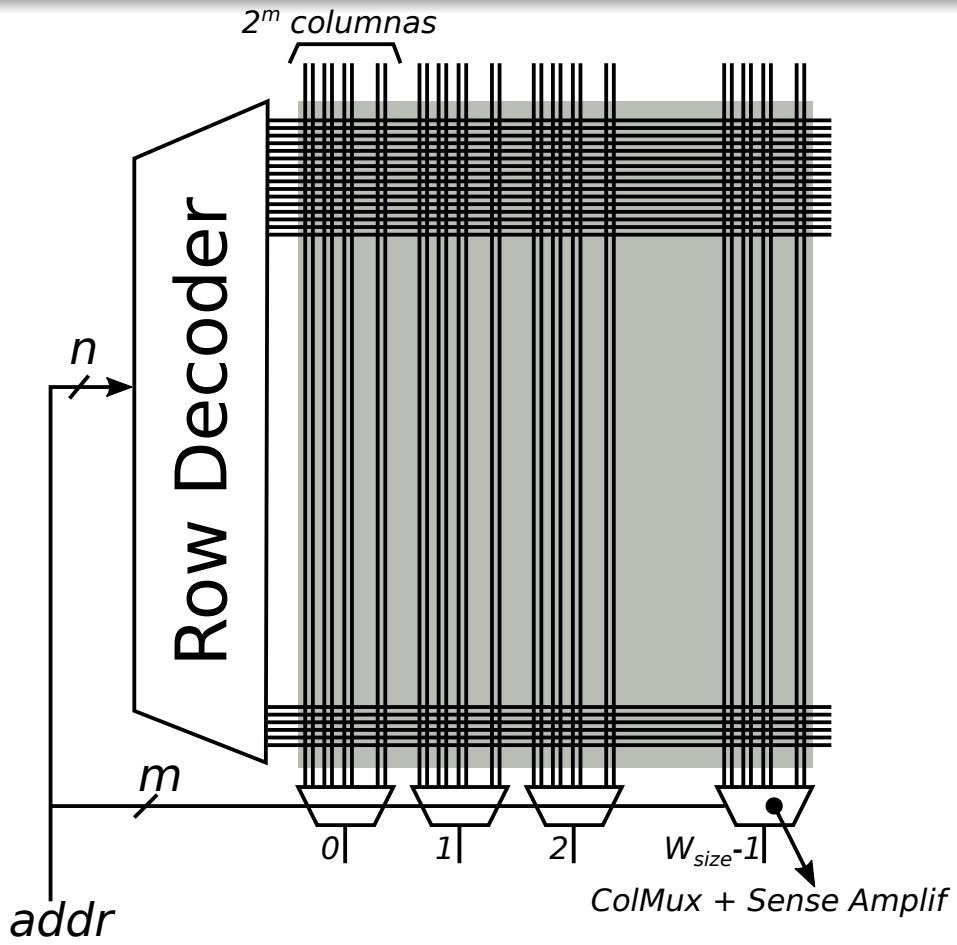
Predecoding

- Cada subset de 4 líneas atada a 16 compuertas AND de 4 entradas, de modo que cada AND puede generar una salida activa única para cada combinación posible.
- Luego utiliza 256 compuertas AND de dos entradas para generar las 256 wordlines.
- Además de ser una lógica sencilla, Predecoding tiene numerosas ventajas frente al arreglo original.
- Las compuertas de dos entradas permiten lograr un circuito mucho mas rápido con menor disipación de potencia, que menos superficie que el primero.
- Pero fundamentalmente es mucho mas escalable, aunque en memorias de mayor capacidad la cantidad de conductores se transforma en un limitante.
- Por otra parte la cantidad de compuertas tambien crece geométricamente conforme aumenta la capacidad de la memoria.

Predecoding

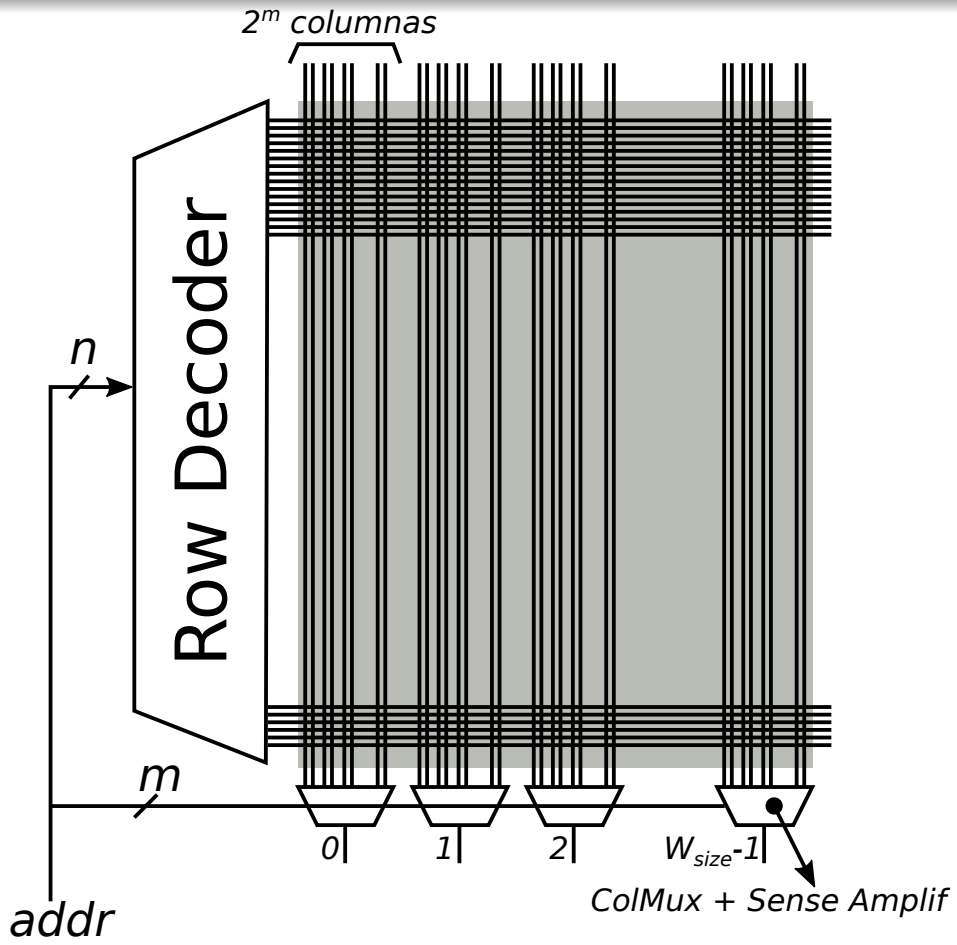


Predecoding



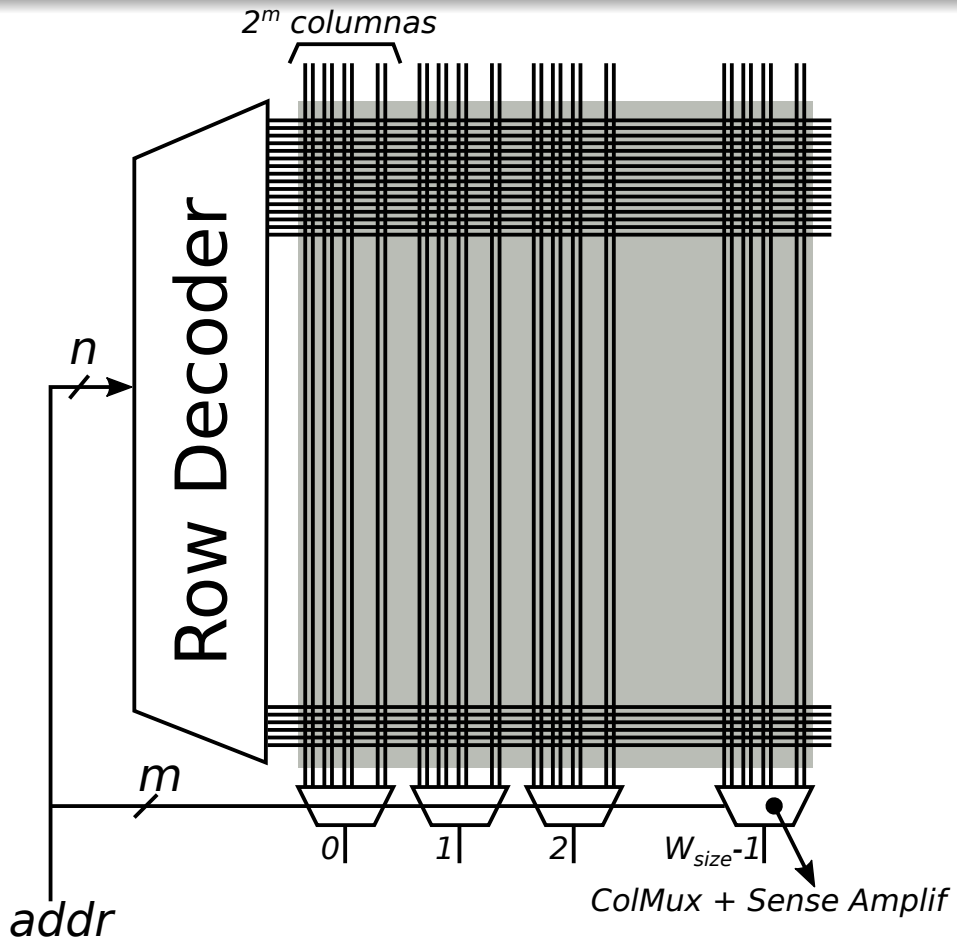
- Para otorgar mas flexibilidad las SRAM se terminan organizando al igual que las DRAM en matrices de bits.

Predecoding



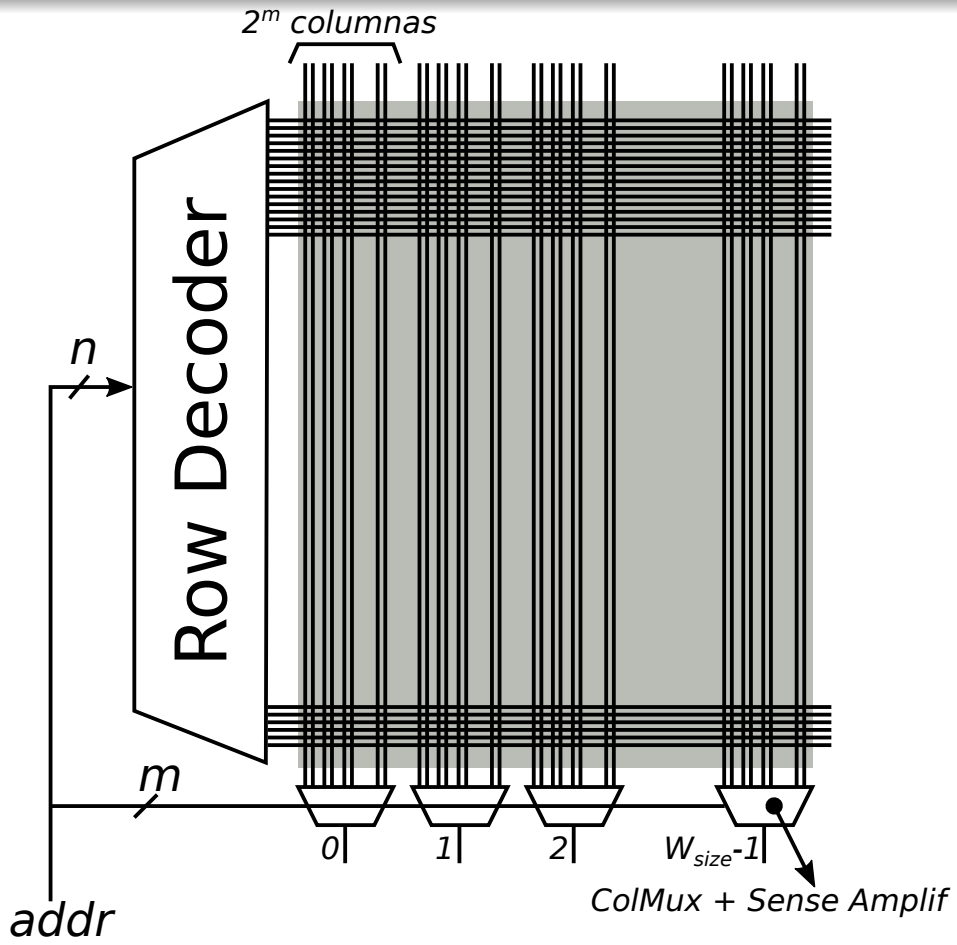
- Para otorgar mas flexibilidad las SRAM se terminan organizando al igual que las DRAM en matrices de bits.
- De este modo un grupo de líneas de direcciones se utiliza para codificar la fila y el restante grupo seleccionará la columna.

Predecoding



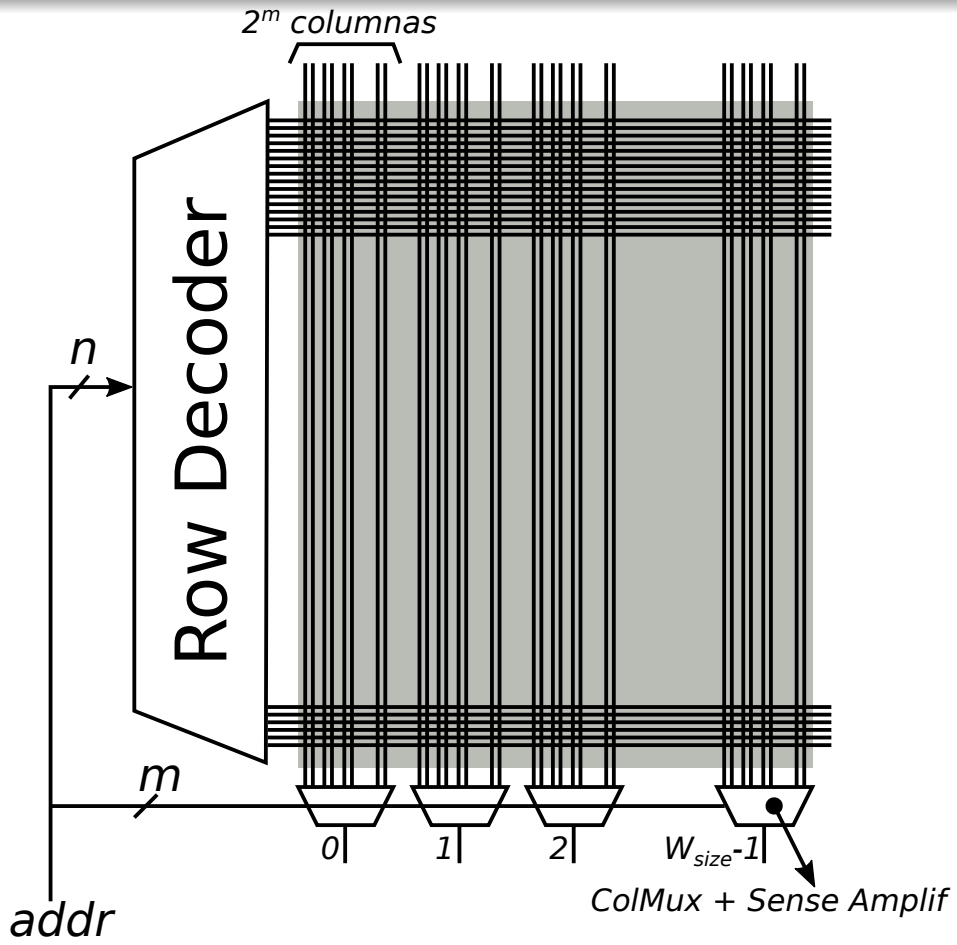
- Para otorgar mas flexibilidad las SRAM se terminan organizando al igual que las DRAM en matrices de bits.
- De este modo un grupo de líneas de direcciones se utiliza para codificar la fila y el restante grupo seleccionará la columna.
- Tanto la fila como la columna decodifican empleando Predecoding.

Predecoding



- Para otorgar mas flexibilidad las SRAM se terminan organizando al igual que las DRAM en matrices de bits.
- De este modo un grupo de líneas de direcciones se utiliza para codificar la fila y el restante grupo seleccionará la columna.
- Tanto la fila como la columna decodifican empleando Predecoding.
- Se omite el detalle en la figura por simplicidad.

Predecoding



- Para otorgar mas flexibilidad las SRAM se terminan organizando al igual que las DRAM en matrices de bits.
- De este modo un grupo de líneas de direcciones se utiliza para codificar la fila y el restante grupo seleccionará la columna.
- Tanto la fila como la columna decodifican empleando Predecoding.
- Se omite el detalle en la figura por simplicidad.

- Los multiplexores permiten seleccionar una de las *m* salidas de Predecoding para enviarla al ***sense amplifier***.

No Particionado

No Particionado

- Es una organización en la que todas las celdas de una fila se activan mediante una única wordline saliente del Row Decoder

No Particionado

- Es una organización en la que todas las celdas de una fila se activan mediante una única wordline saliente del Row Decoder
- O sea un esquema como el del slide anterior.

No Particionado

- Es una organización en la que todas las celdas de una fila se activan mediante una única wordline saliente del Row Decoder
- O sea un esquema como el del slide anterior.
- Es un esquema eficiente para memoria de alrededor de 1KByte. Mas allá de este tamaño aparecen algunos problemas:

No Particionado

- Es una organización en la que todas las celdas de una fila se activan mediante una única wordline saliente del Row Decoder
- O sea un esquema como el del slide anterior.
- Es un esquema eficiente para memoria de alrededor de 1KByte. Mas allá de este tamaño aparecen algunos problemas:
  1. Aumenta el tamaño de la fila y por lo tanto la cantidad de transistores que cargan con su entrada en la wordline, y a la column line, aumentando la capacitancia sobre las mismas con los consiguiente efectos adversos en delay y disipación de potencia.

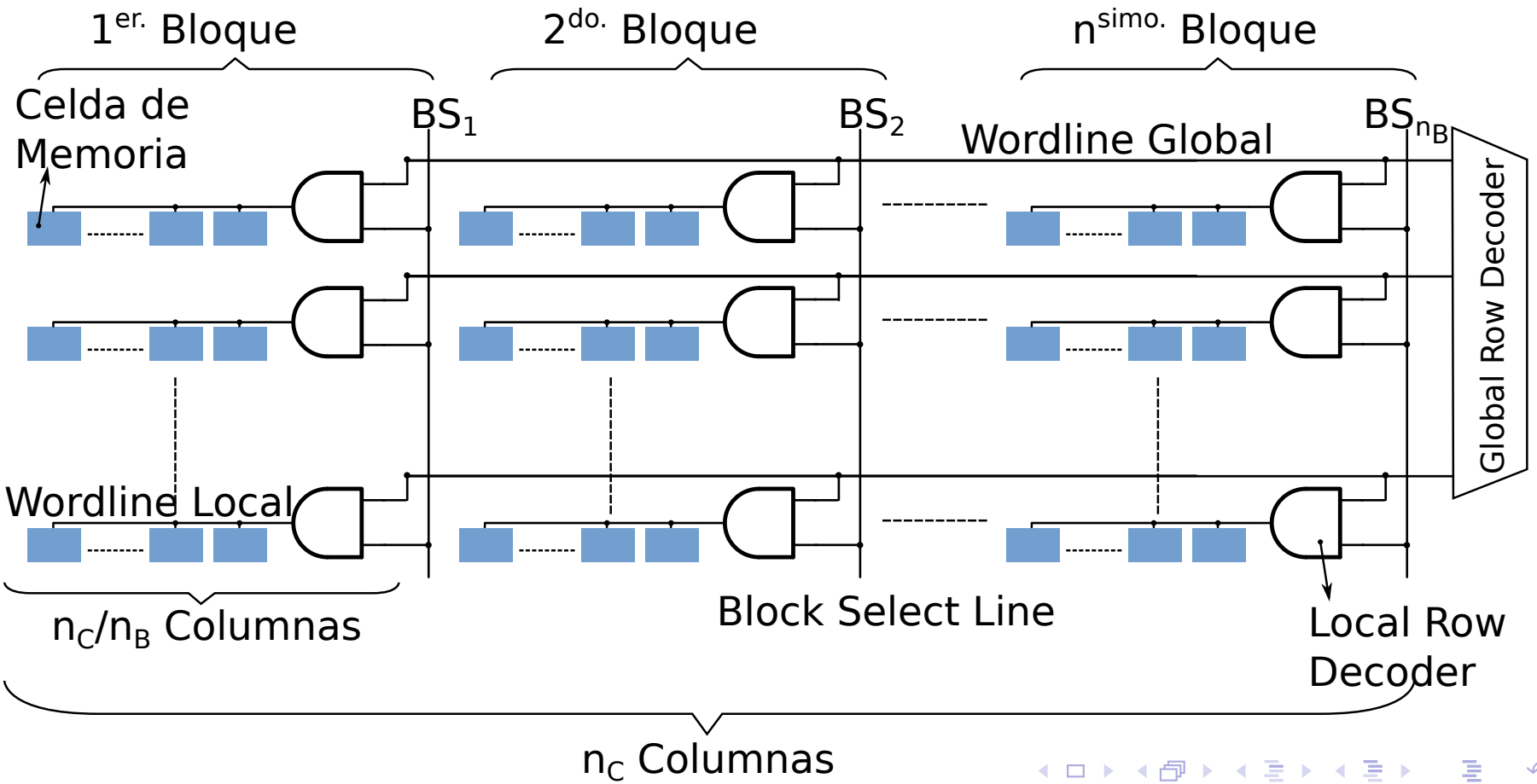
No Particionado

- Es una organización en la que todas las celdas de una fila se activan mediante una única wordline saliente del Row Decoder
- O sea un esquema como el del slide anterior.
- Es un esquema eficiente para memoria de alrededor de 1KByte. Mas allá de este tamaño aparecen algunos problemas:
  1. Aumenta el tamaño de la fila y por lo tanto la cantidad de transistores que cargan con su entrada en la wordline, y a la column line, aumentando la capacitancia sobre las mismas con los consiguiente efectos adversos en delay y disipación de potencia.
  2. Aumenta la longitud de las wordlines y por lo tanto su capacitancia y resistencia distribuida. Mas delay y mas disipación de potencia.

No Particionado

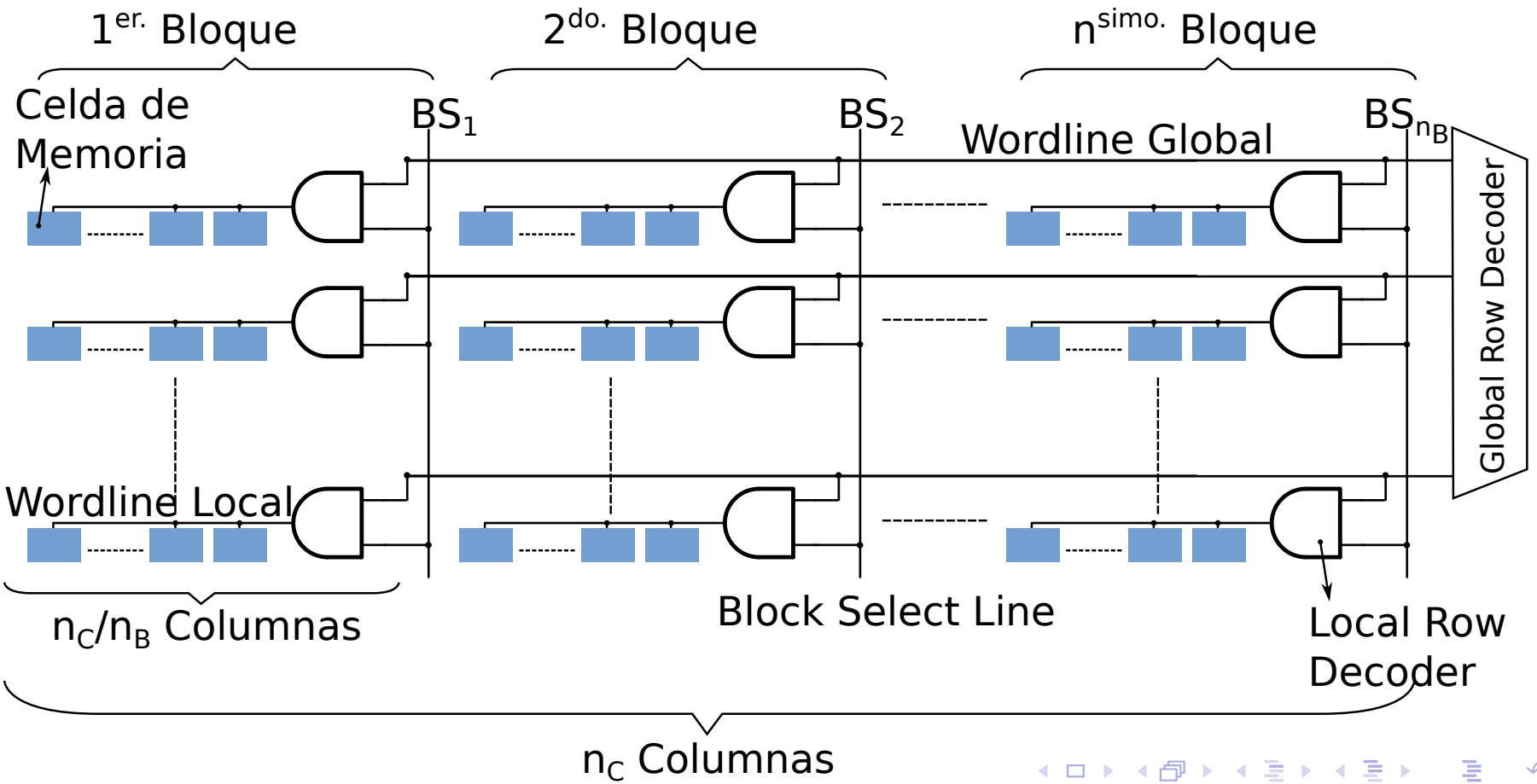
- Es una organización en la que todas las celdas de una fila se activan mediante una única wordline saliente del Row Decoder
- O sea un esquema como el del slide anterior.
- Es un esquema eficiente para memoria de alrededor de 1KByte. Mas allá de este tamaño aparecen algunos problemas:
  1. Aumenta el tamaño de la fila y por lo tanto la cantidad de transistores que cargan con su entrada en la wordline, y a la column line, aumentando la capacitancia sobre las mismas con los consiguiente efectos adversos en delay y disipación de potencia.
  2. Aumenta la longitud de las wordlines y por lo tanto su capacitancia y resistencia distribuida. Mas delay y mas disipación de potencia.
  3. Aumenta el número de columnas que se activan mediante una sola wordline. Esto suma mas disipación de potencia a los dos puntos anteriores.

División de Wordline



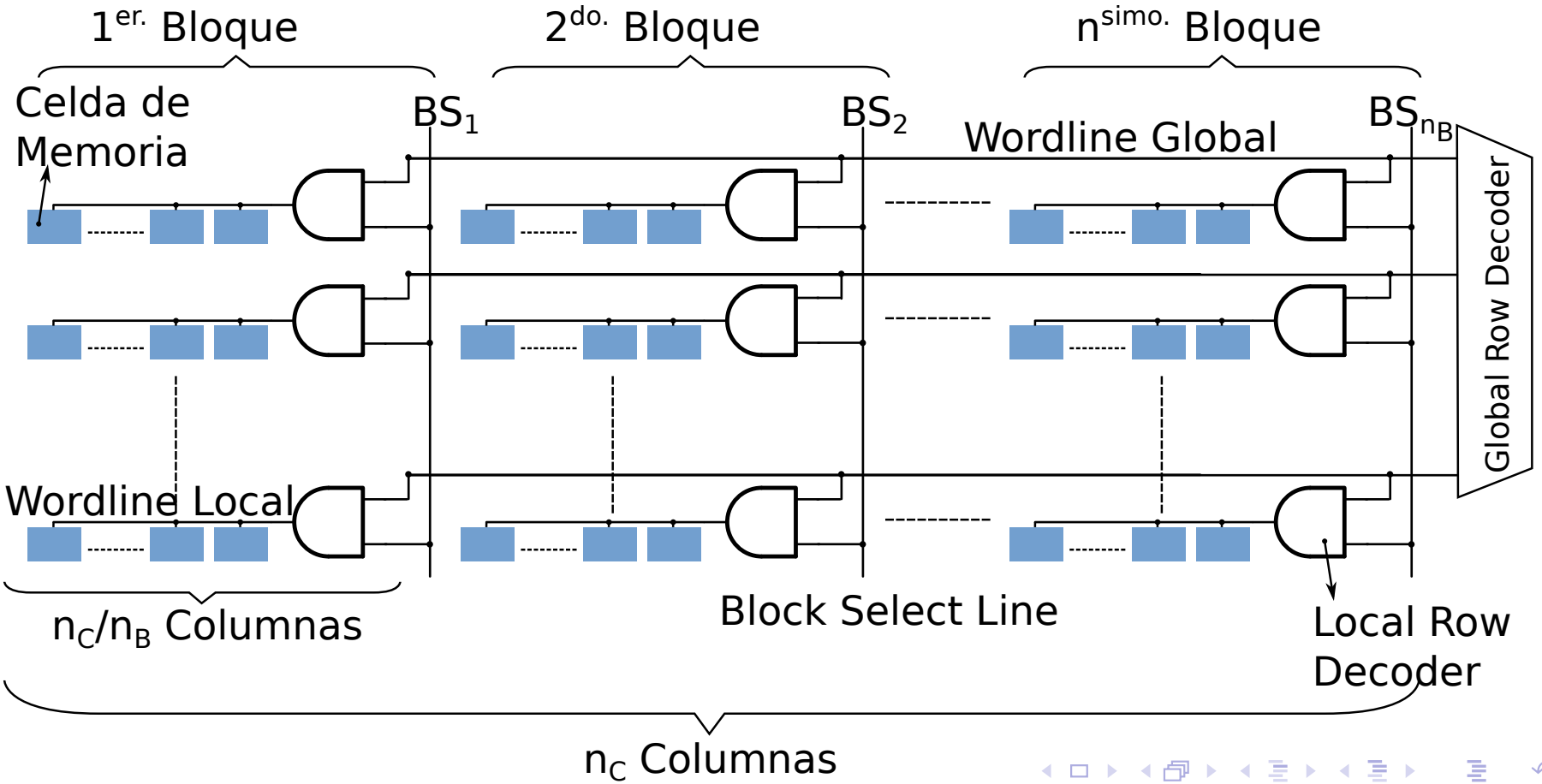
División de Wordline

- Se trata de método mas empleado debido a su utilidad y mínimas desventajas.



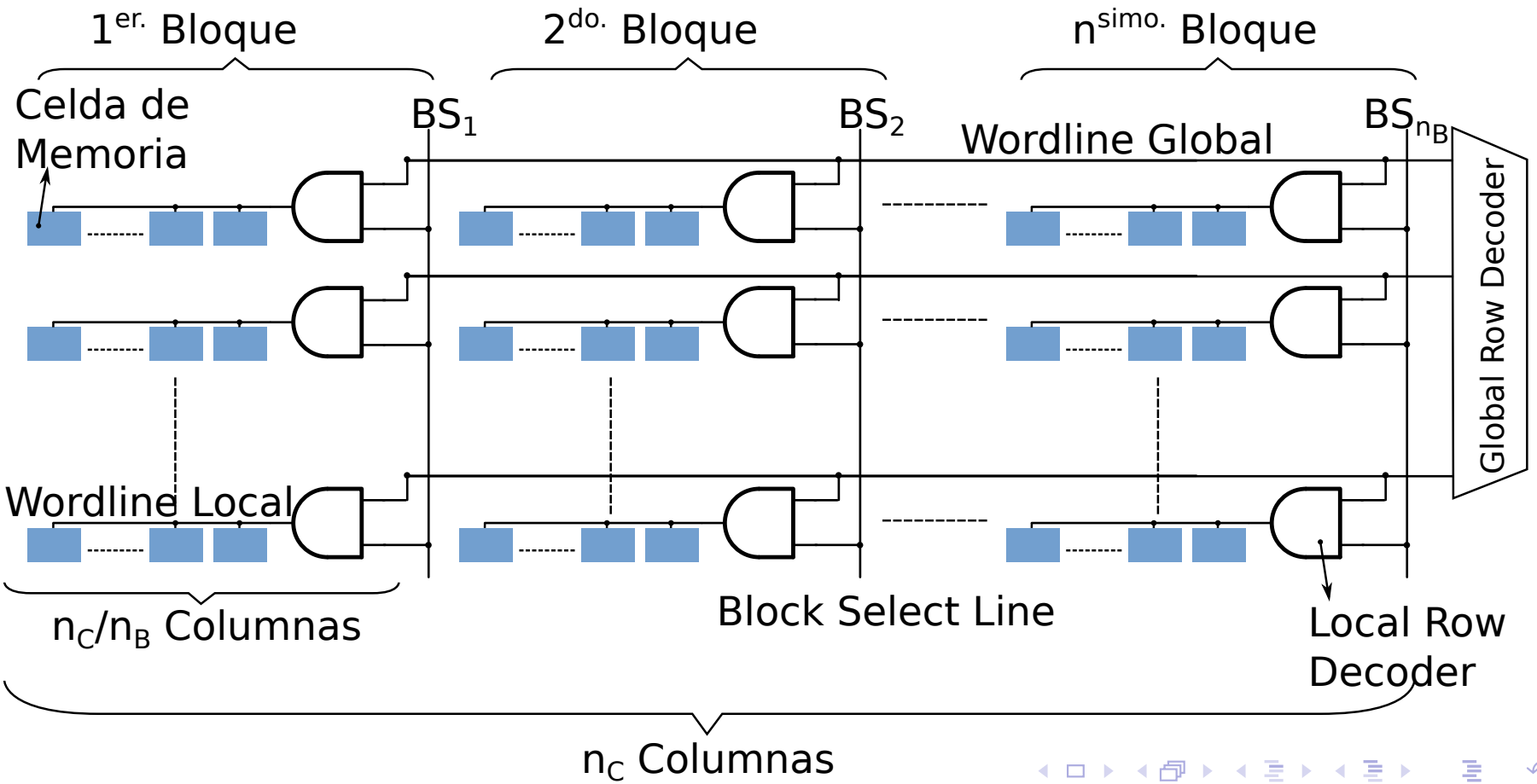
División de Wordline

- Se trata de método mas empleado debido a su utilidad y mínimas desventajas.
- Consiste en dividir la Wordline (**G**lobal **W**ord**l**ine **GWL**), en un número fijo de bloques.



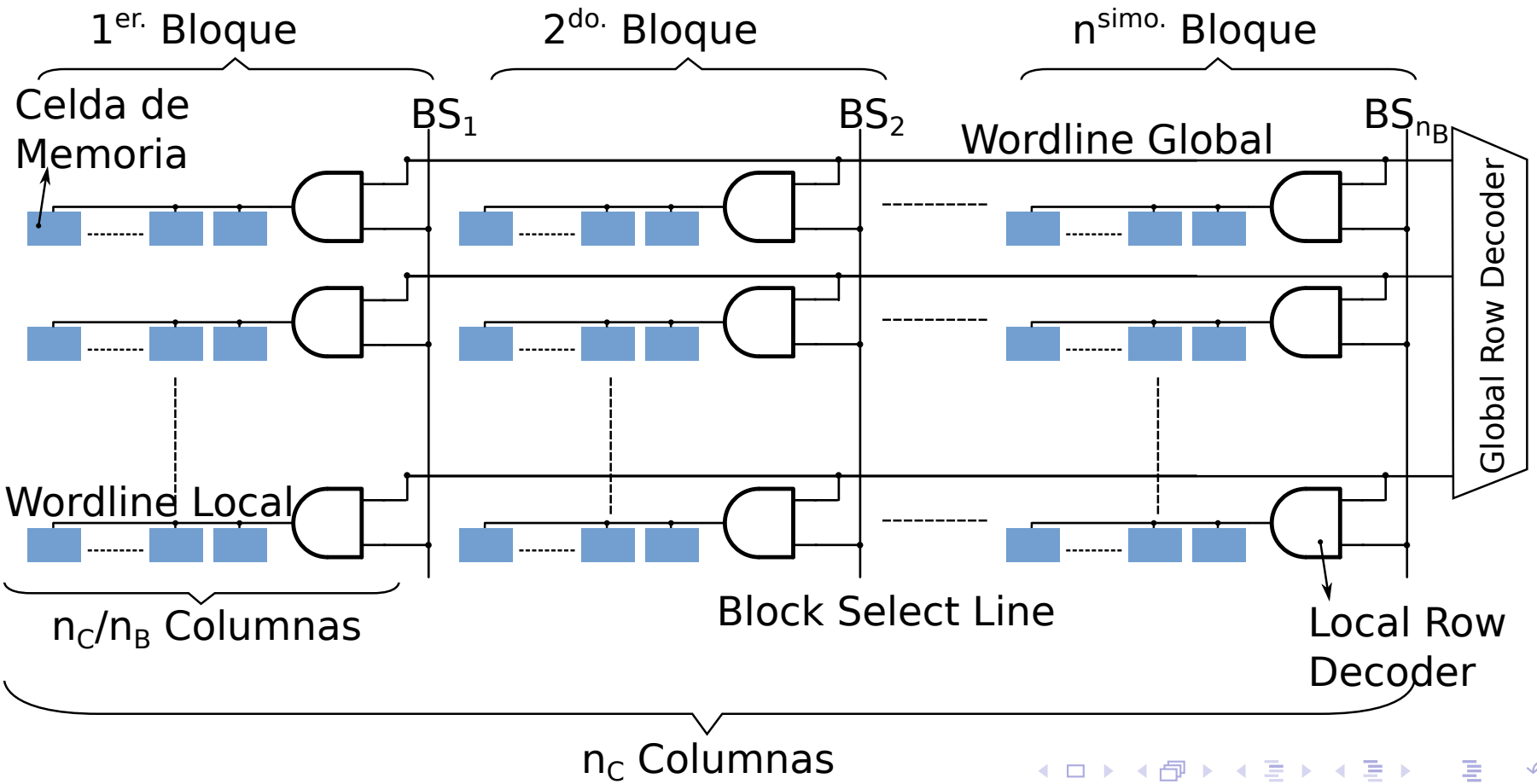
División de Wordline

- No se habilitan todas las celdas de una fila. En vez de esto, la **GWL** se "**AND**ea"con una Señal Block Select Line, proveniente de un subset de líneas de entrada, generando una **L**ocal **W**ord**l**ine (**LWL**).***Solo se habilitan las celdas conectadas a esta LWL***



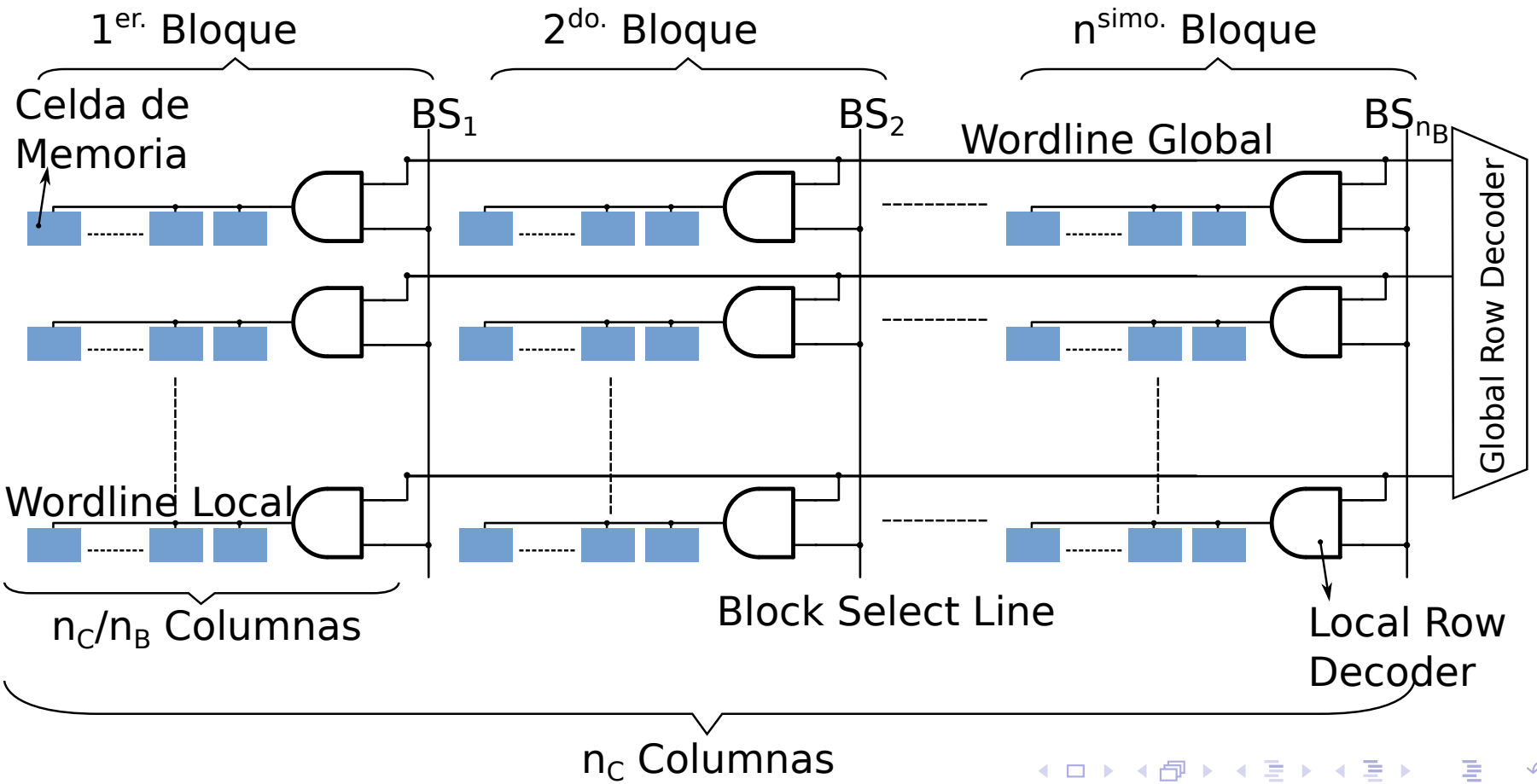
División de Wordline

- Las n_C columnas del array de memoria se divide en n_B bloques.



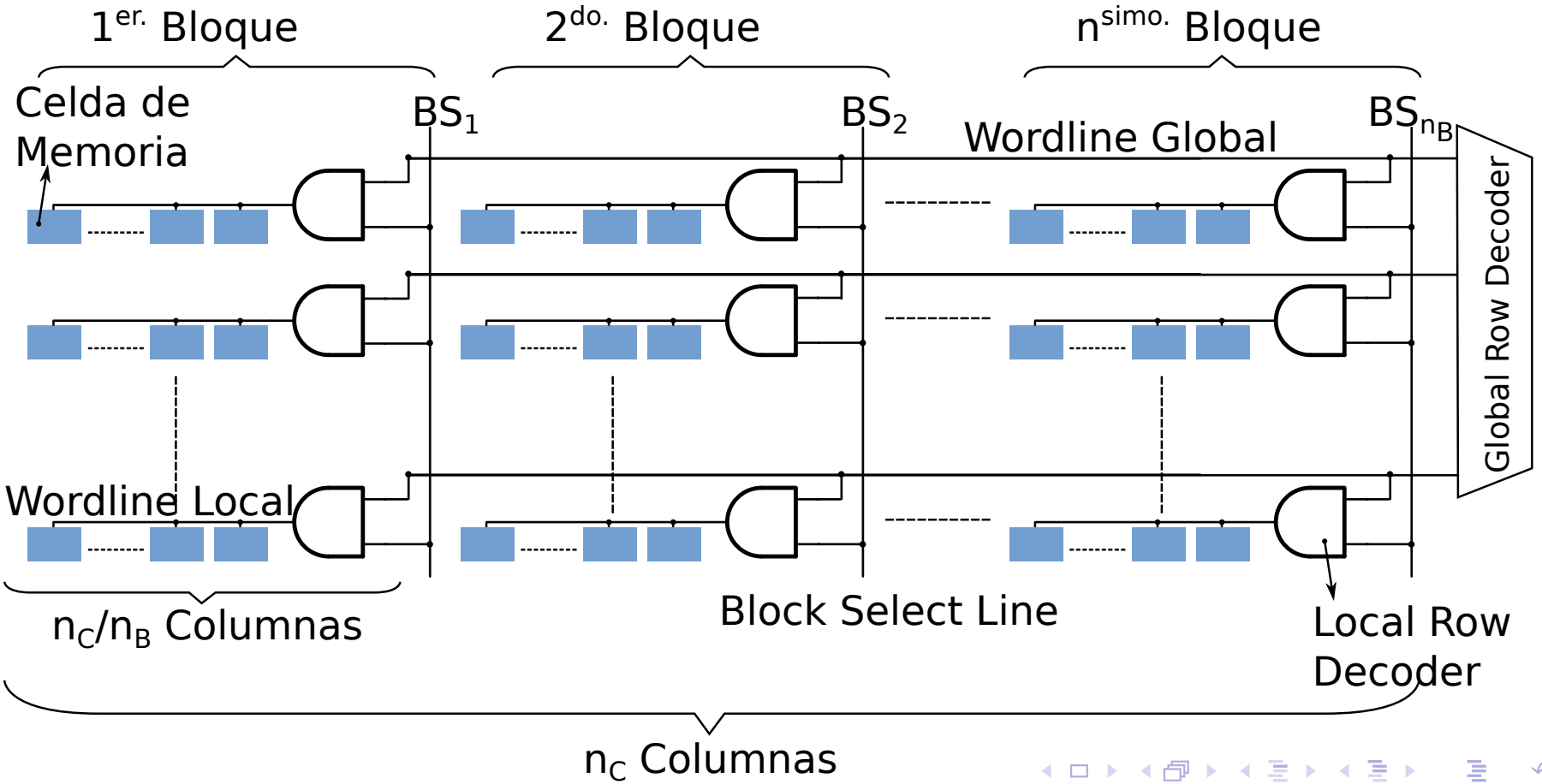
División de Wordline

- Las n_C columnas del array de memoria se divide en n_B bloques.
- Cada bloque entonces se compone de n_C/n_B columnas.



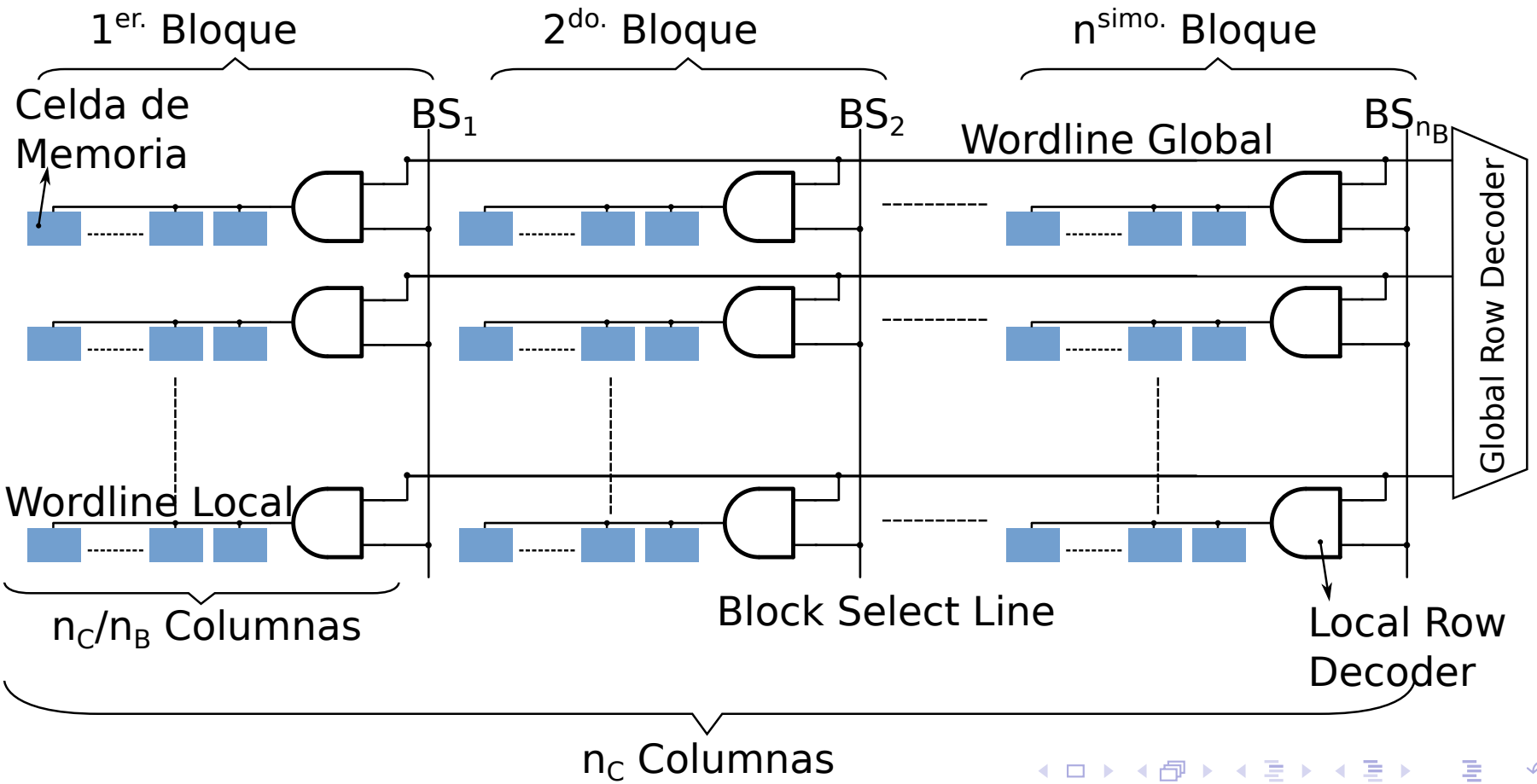
División de Wordline

- Las n_C columnas del array de memoria se divide en n_B bloques.
- Cada bloque entonces se compone de n_C/n_B columnas.
- La fracción de celdas conectadas a cada **LWL** es $1/n_B$ respecto de Predecoding. Por lo tanto la potencia disipada es $1/n_B$.



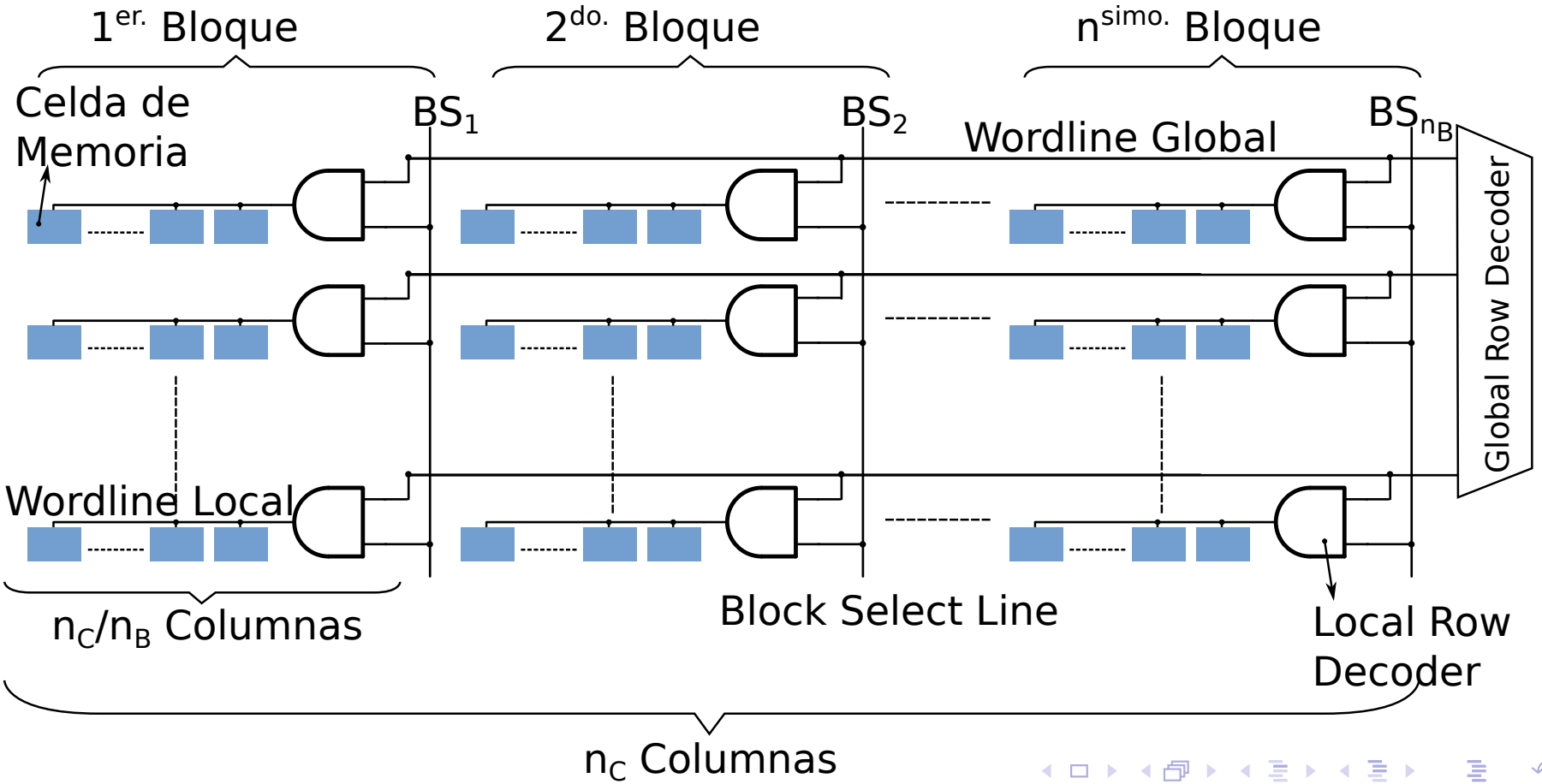
División de Wordline

- La relación de potencia disipada es algo menos eficiente en las escrituras ya se disipa mas potencia en la celda en si.



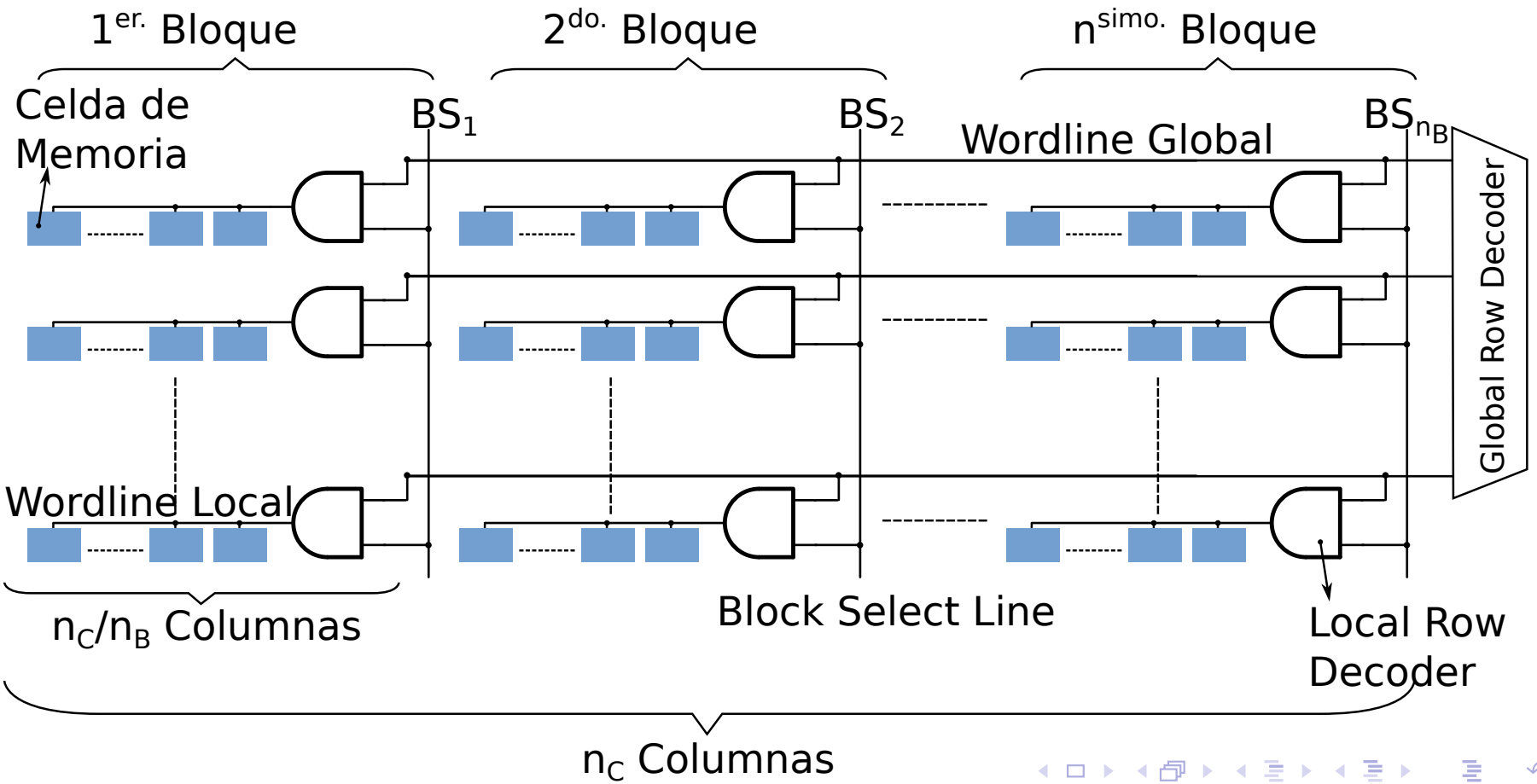
División de Wordline

- La relación de potencia disipada es algo menos eficiente en las escrituras ya se disipa mas potencia en la celda en si.
- El delay también disminuye por tener muchas menos cargas en cada **LWL**.



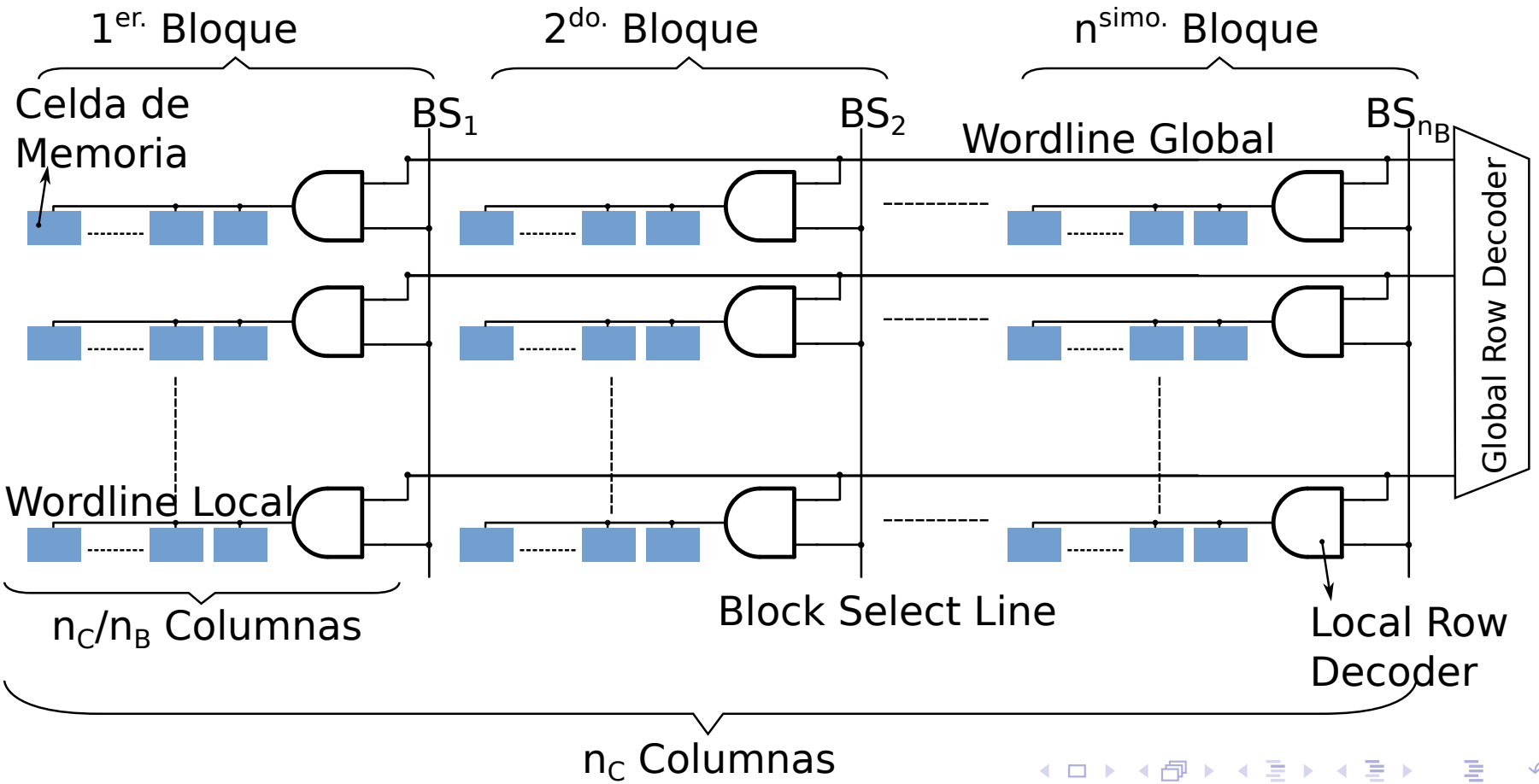
División de Wordline

- La **GWL** maneja un número de decodificadores de **LWL** mucho menor al del caso no particionado, ahorrando mucha disipación



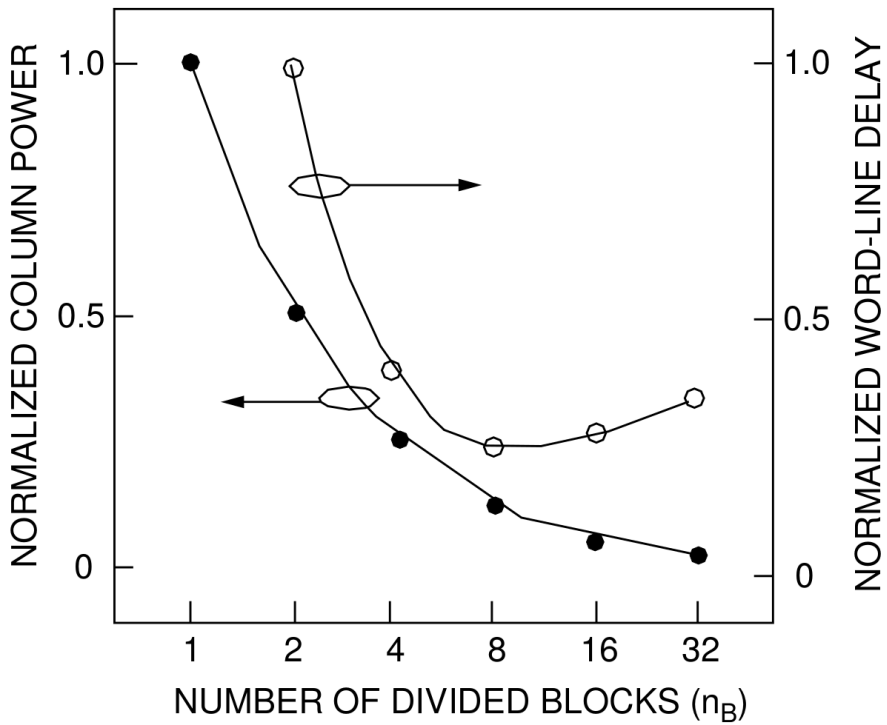
División de Wordline

- La **GWL** maneja un número de decodificadores de **LWL** mucho menor al del caso no particionado, ahorrando mucha disipación
- El delay en el **GWL** mas el **LWL** activo es mucho menor que el del caso No Particionado, en especial a mayor tamaño de la SRAM.



División de Wordline

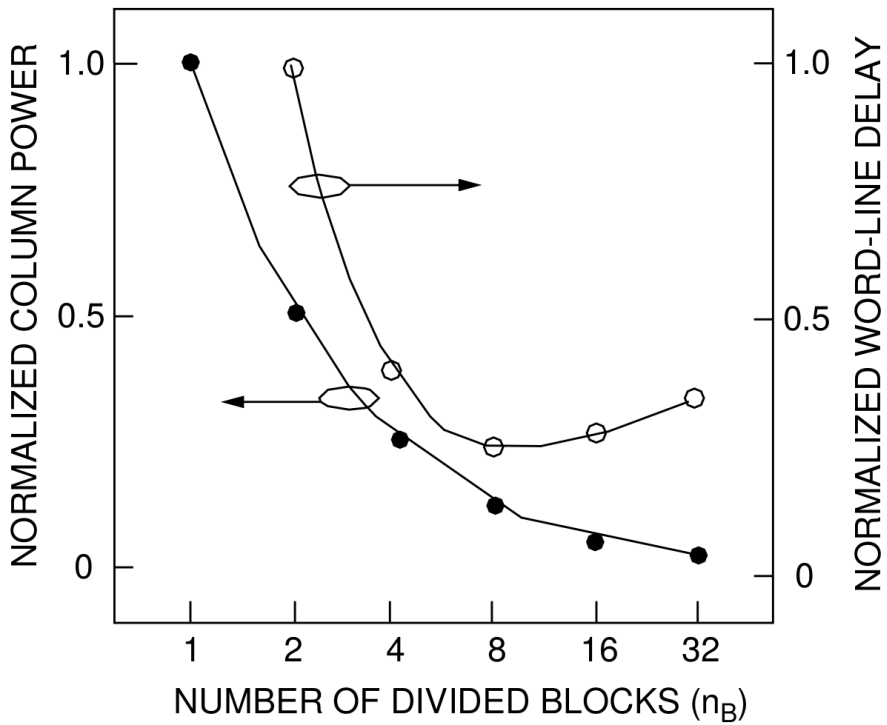
- El gráfico muestra el efecto en disipación d potencia y delay de columna cambiando el número de n_B para una SRAM de 8192 celdas de 8 bits c/u[2].



[2]M. Yoshimoto et al. 1983. A divided word-line structure in the static RAM and its application to a 64K full CMOS RAM, IEEE J. Solid-State Circuits, SC-18(5), 479–485,Oct. 1983

División de Wordline

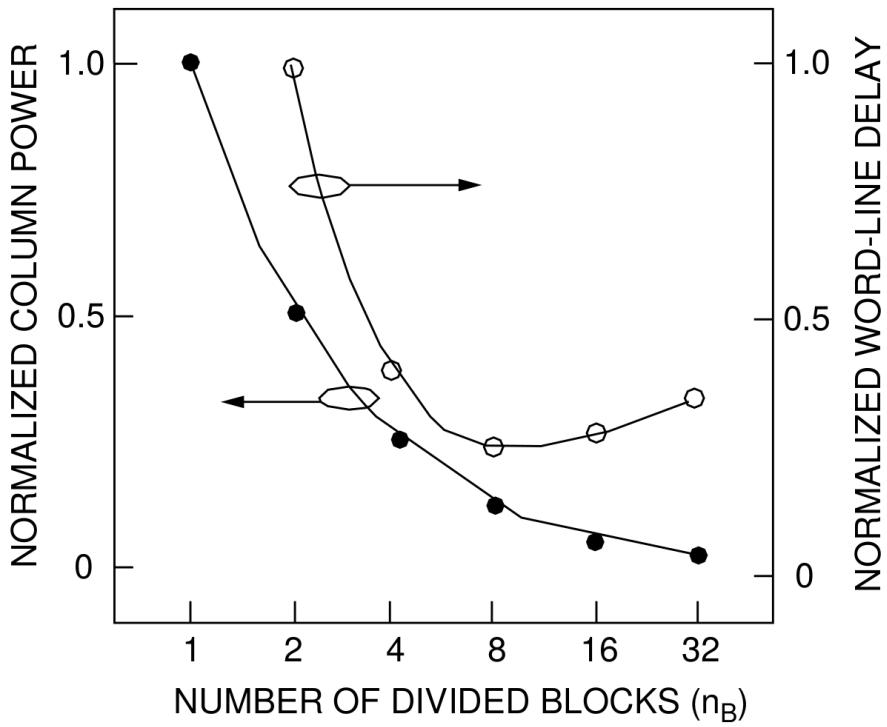
- El gráfico muestra el efecto en disipación d potencia y delay de columna cambiando el número de n_B para una SRAM de 8192 celdas de 8 bits c/u[2].
- 8Kbytes, resulta muy aplicativo ya que una SRAM de este tamaño se utiliza como bloque básico para armar caches de tamaño mayor.



[2]M. Yoshimoto et al. 1983. A divided word-line structure in the static RAM and its application to a 64K full CMOS RAM, IEEE J. Solid-State Circuits, SC-18(5), 479–485,Oct. 1983

División de Wordline

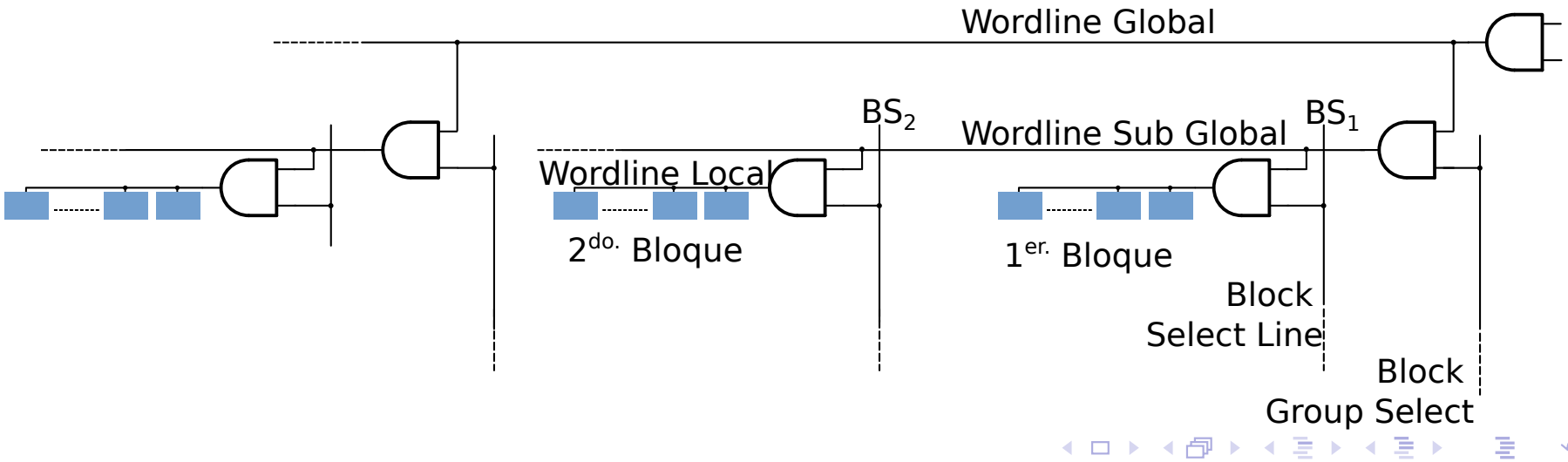
- El gráfico muestra el efecto en disipación d potencia y delay de columna cambiando el número de n_B para una SRAM de 8192 celdas de 8 bits c/u[2].
- 8Kbytes, resulta muy aplicativo ya que una SRAM de este tamaño se utiliza como bloque básico para armar caches de tamaño mayor.
- Un bloque de 128 Kbytes se construye a partir de 16 bloques básicos de 8Kbytes.



[2] M. Yoshimoto et al. 1983. A divided word-line structure in the static RAM and its application to a 64K full CMOS RAM, IEEE J. Solid-State Circuits, SC-18(5), 479–485,Oct. 1983

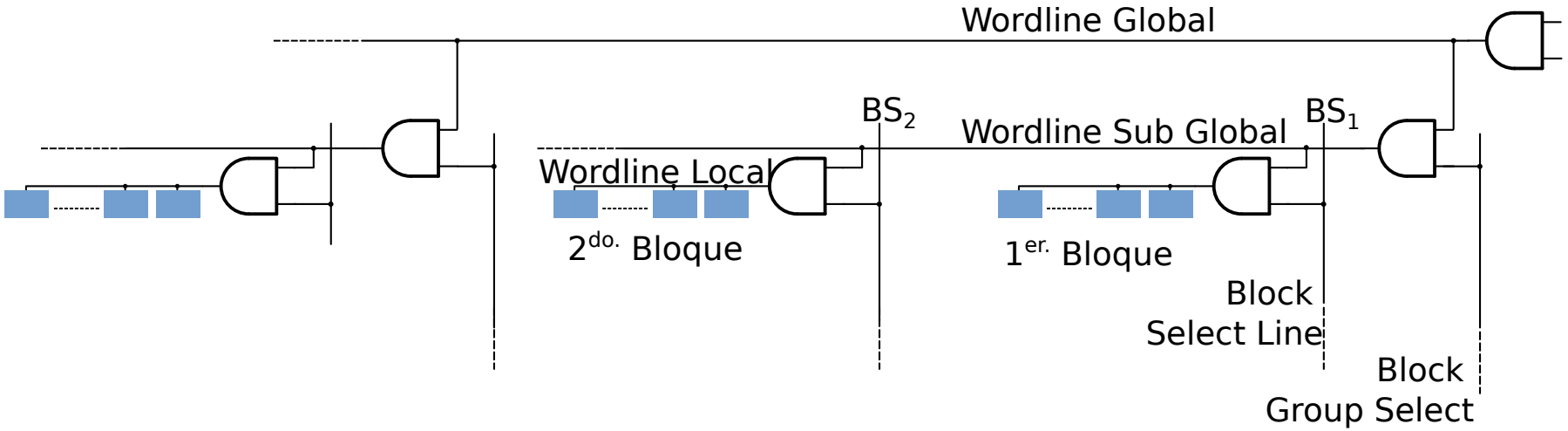
Decodificación de Wordline Jerárquica

- **HWD** es una extensión natural de **WDL** cuando aumenta la capacidad de la SRAM.



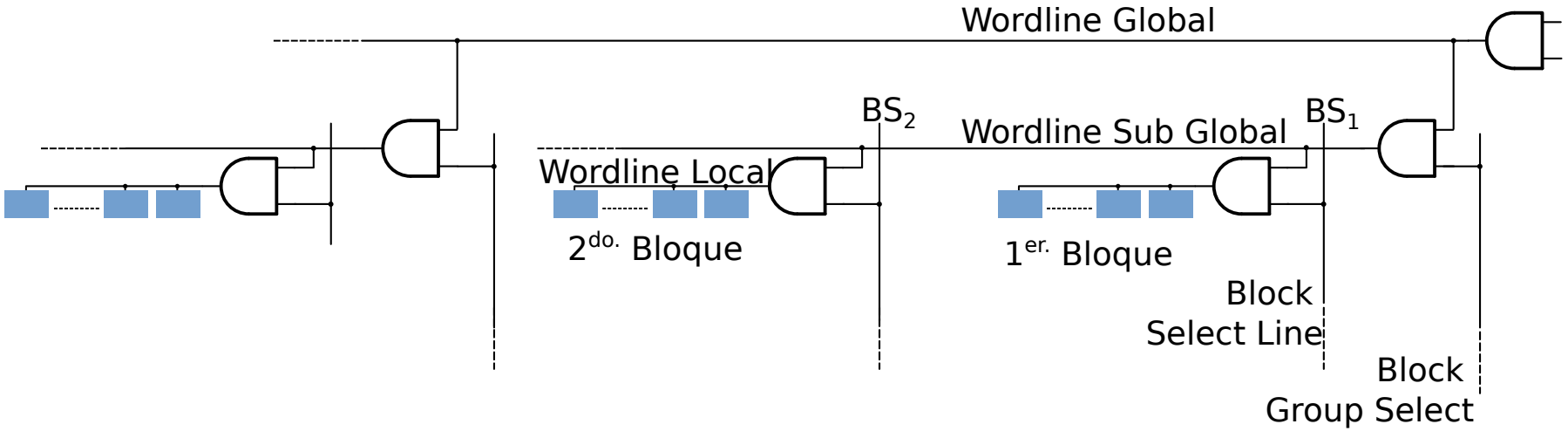
Decodificación de Wordline Jerárquica

- **HWD** es una extensión natural de **WDL** cuando aumenta la capacidad de la SRAM.
- Básicamente distribuye las capacitancias de manera mas eficiente, ya que las cargas por Sub Global Wordline se mantienen bajas (en el esquema **DWL** la Wordline hubiese sido muy grande y la capacidad de carga inadmisible).



Decodificación de Wordline Jerárquica

- **HWD** es una extensión natural de **WDL** cuando aumenta la capacidad de la SRAM.
- Básicamente distribuye las capacitancias de manera mas eficiente, ya que las cargas por Sub Global Wordline se mantienen bajas (en el esquema **DWL** la Wordline hubiese sido muy grande y la capacidad de carga inadmisible).
- A 256 Kbytes no se percibe diferencia entre **DWL** y **HWD**, pero para SRAMs de 4 Mbytes de capacidad se verifica disminución de delay del 20 % y 30 % menos de carga capacitiva.



Temario

Diagrama general de circuitos periféricos

Un sistema de celdas para construir una SRAM requiere de un conjunto de lógica adicional.

Diagrama general de circuitos periféricos

Un sistema de celdas para construir una SRAM requiere de un conjunto de lógica adicional.

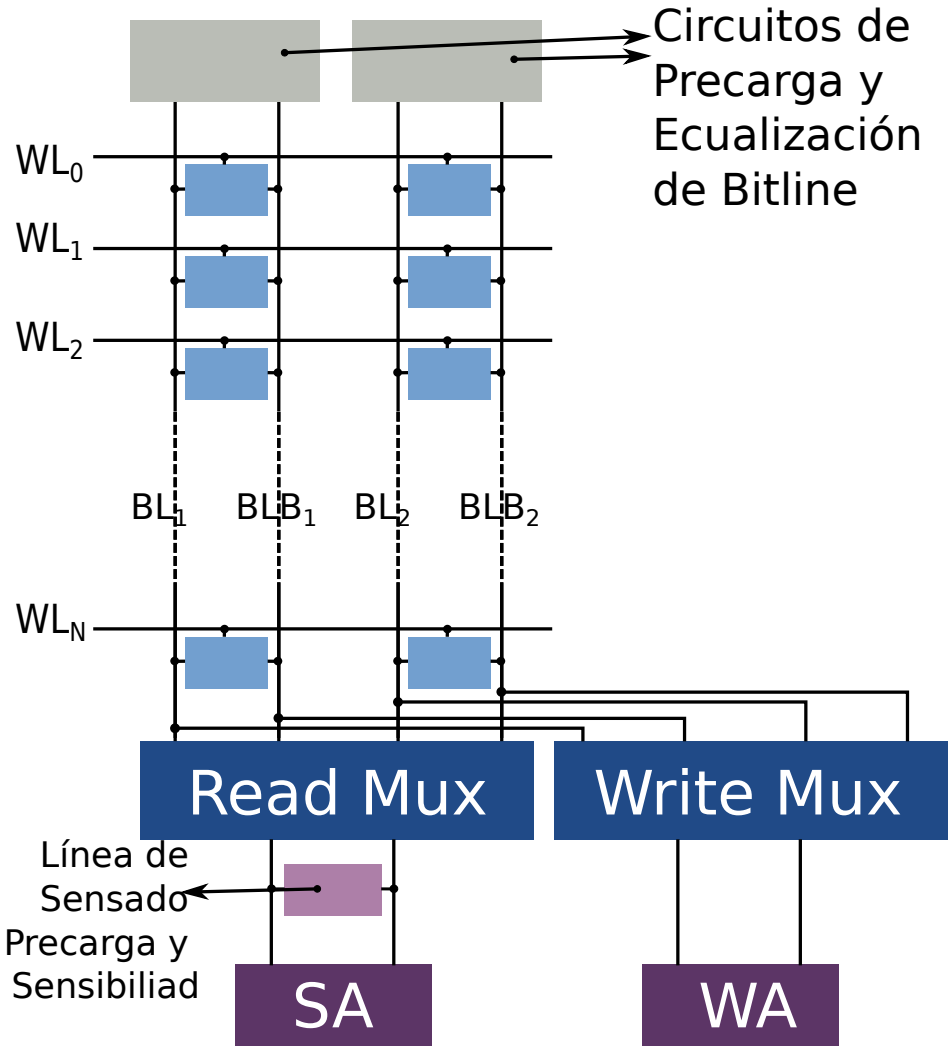
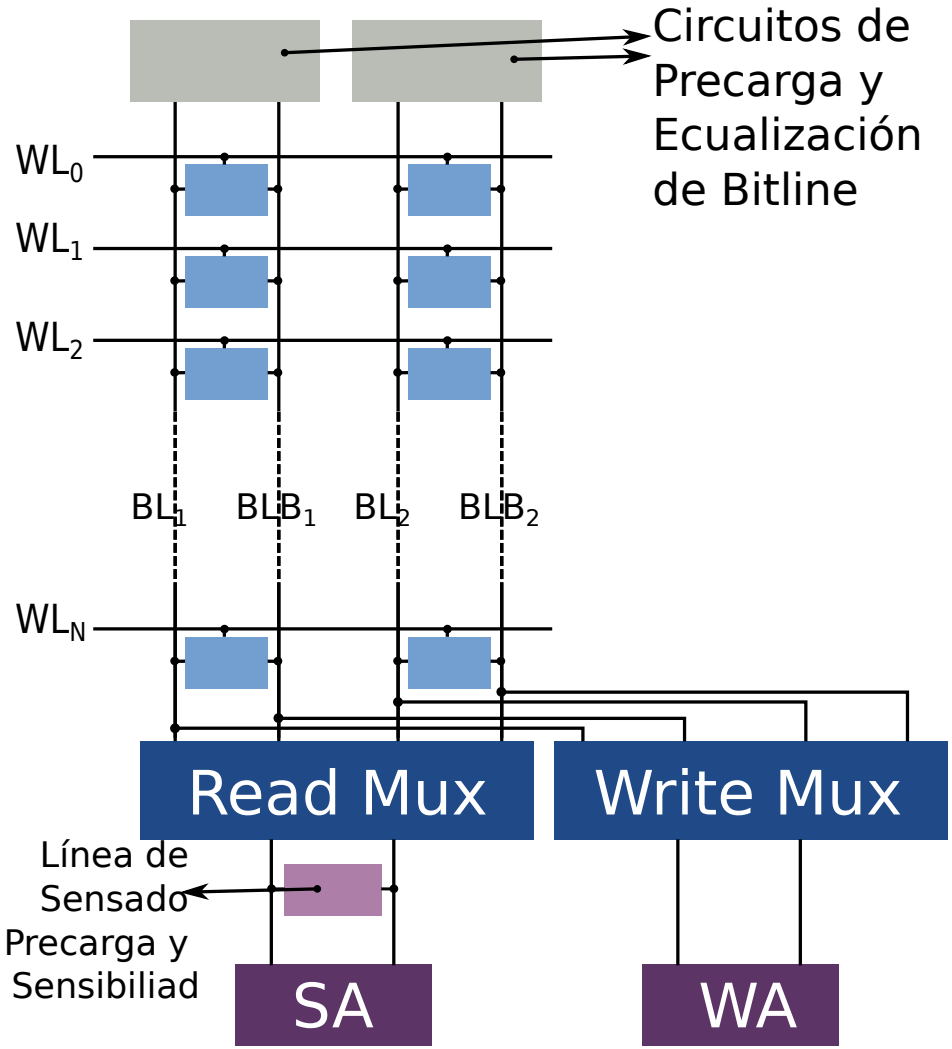


Diagrama general de circuitos periféricos

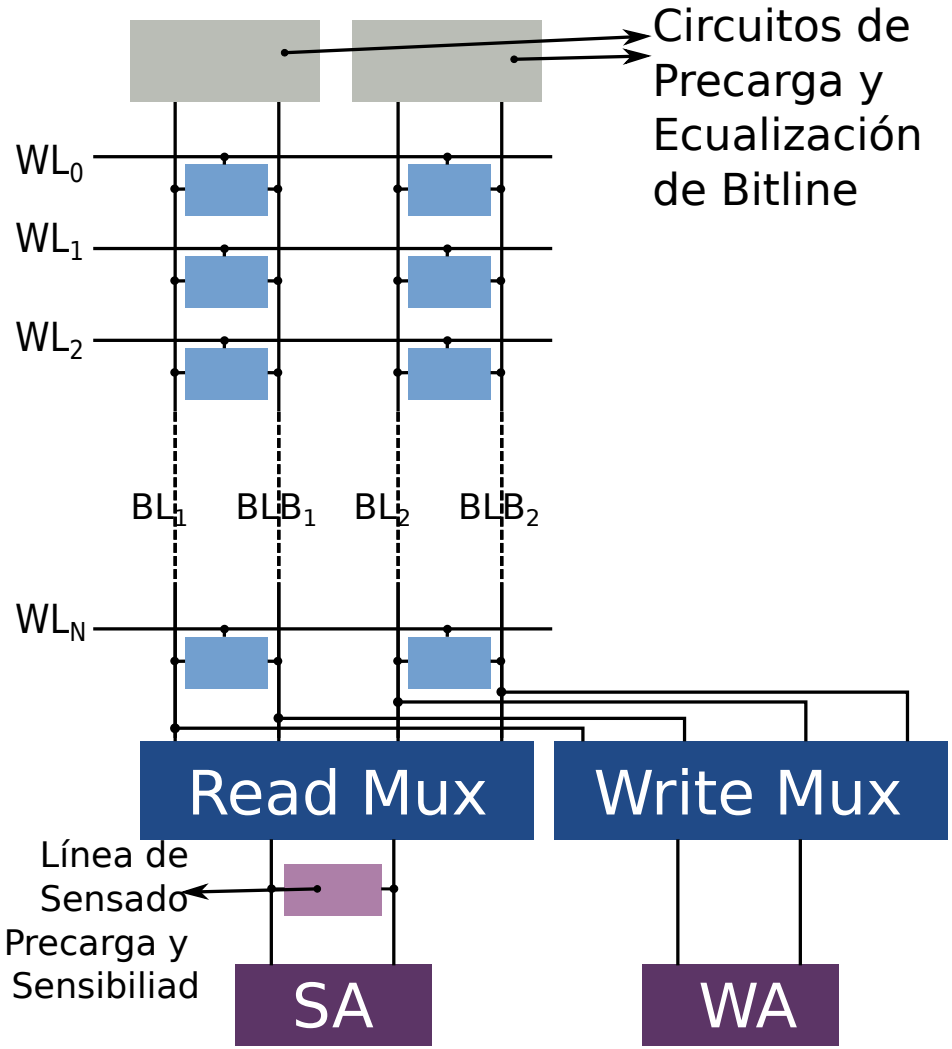
Un sistema de celdas para construir una SRAM requiere de un conjunto de lógica adicional.



- Se asume que todos los bitlines se precargan a una tensión predeterminada: V_{DD}

Diagrama general de circuitos periféricos

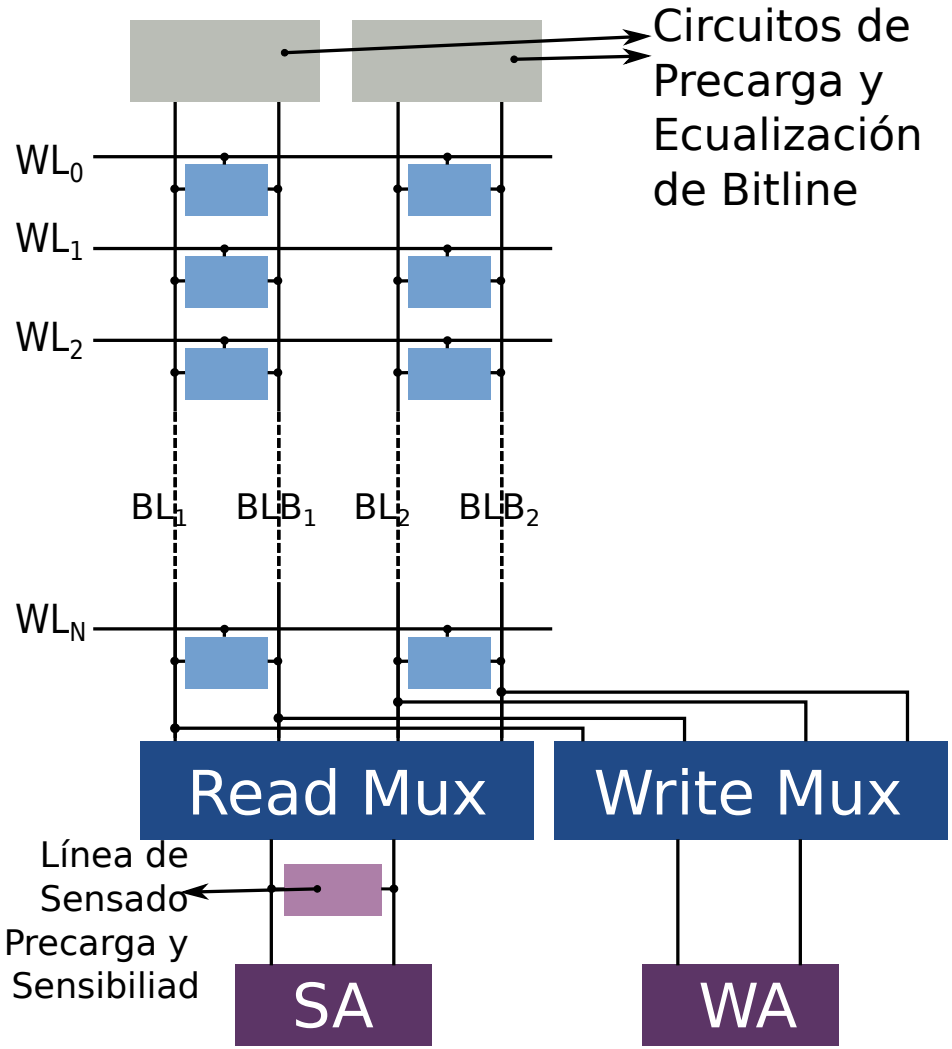
Un sistema de celdas para construir una SRAM requiere de un conjunto de lógica adicional.



- Se asume que todos los bitlines se precargan a una tensión predeterminada: V_{DD}
- Cuando se activa una **`WL`**, todas las celdas de memoria conectadas a ella tienen sus transistores de acceso habilitados.

Diagrama general de circuitos periféricos

Un sistema de celdas para construir una SRAM requiere de un conjunto de lógica adicional.



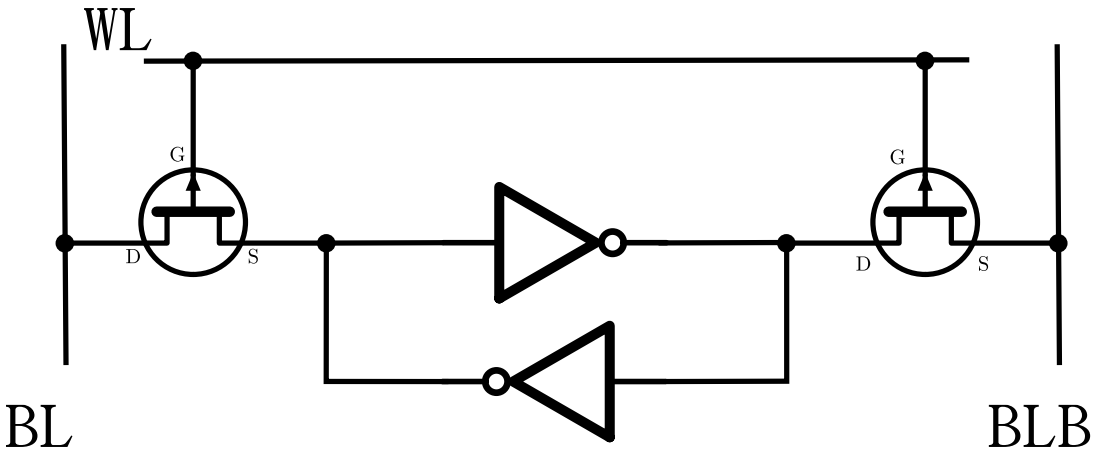
- Se asume que todos los bitlines se precargan a una tensión predeterminada: V_{DD}
- Cuando se activa una **WL**, todas las celdas de memoria conectadas a ella tienen sus transistores de acceso habilitados.
- Si la operación es de lectura por el funcionamiento propio del biestable que compone la celda una de las Bit Lines generará un pull up y la otra en cambio un pull down.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:



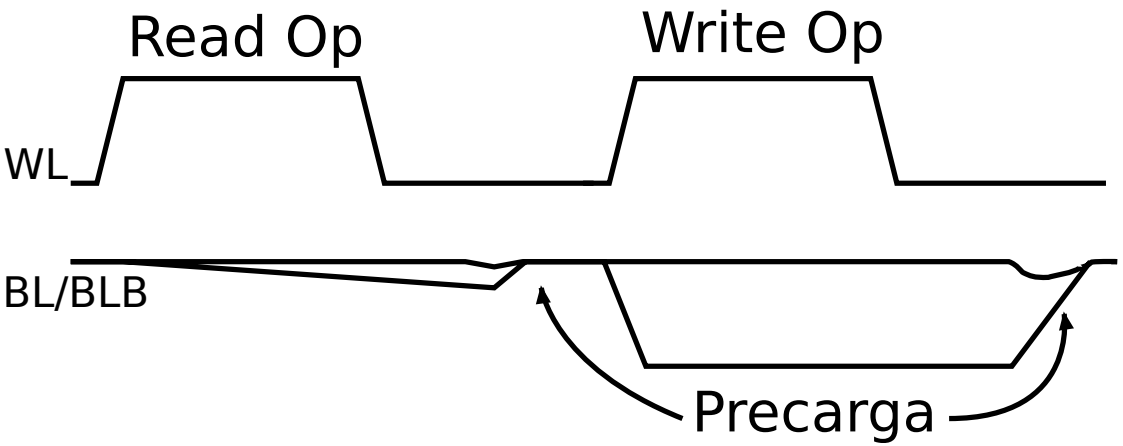
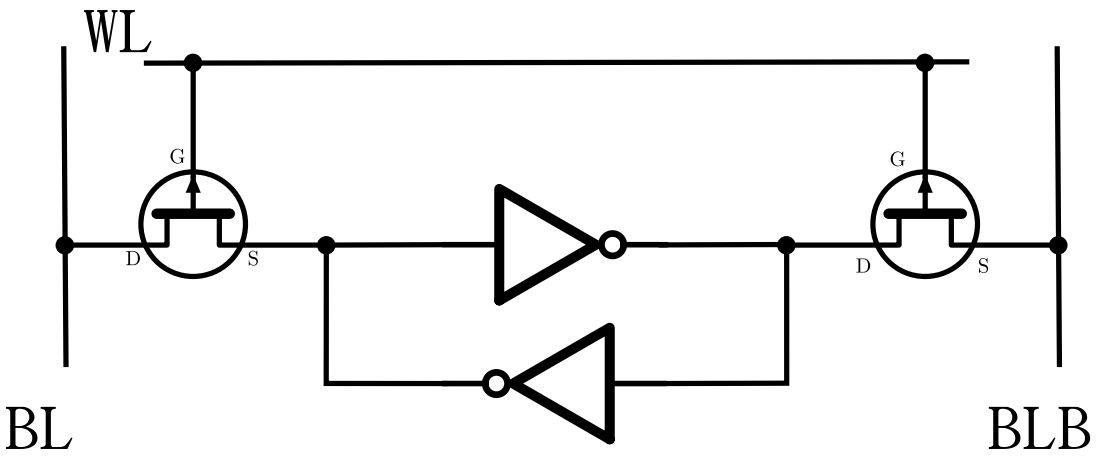
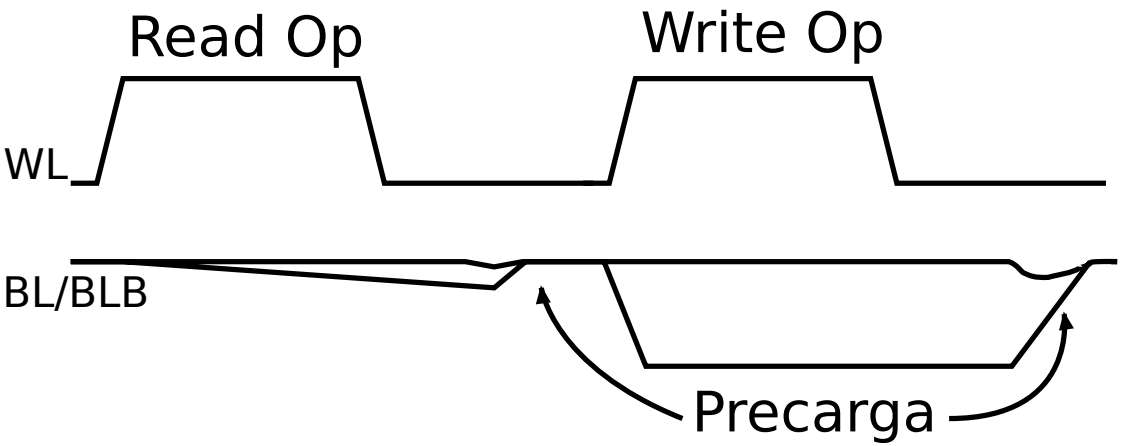


Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

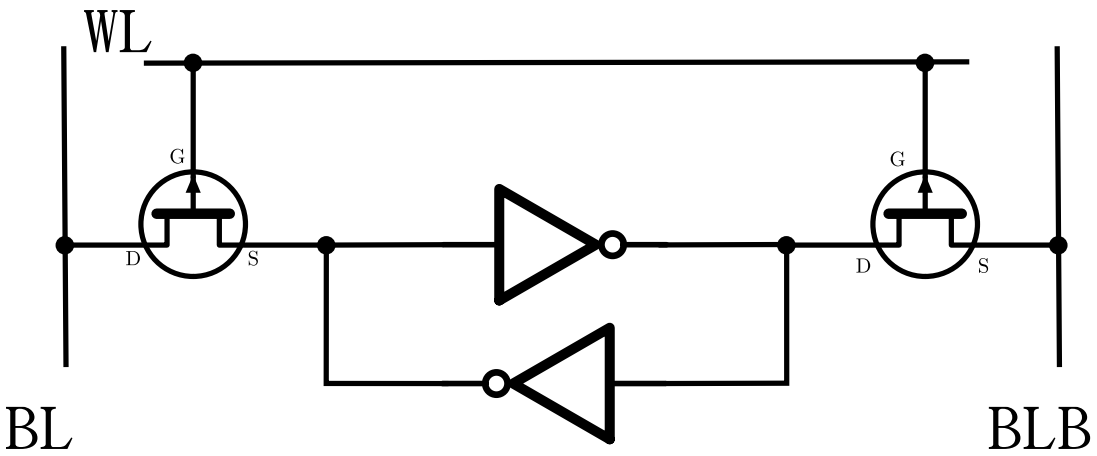


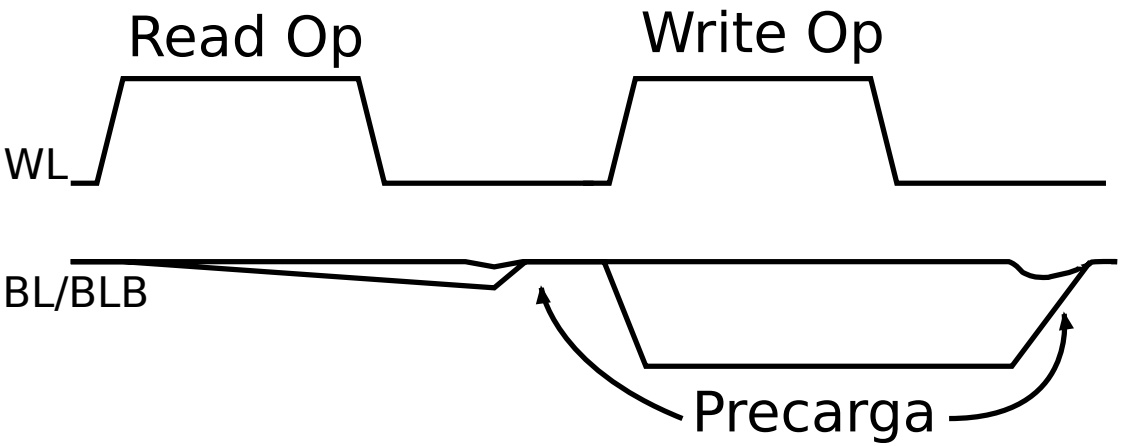


- Como el estado inicial de tensión en las Bit lines era alto, el pull up ya era un hecho de modo que interesa analizar el pull down en la Bit Line complementaria.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

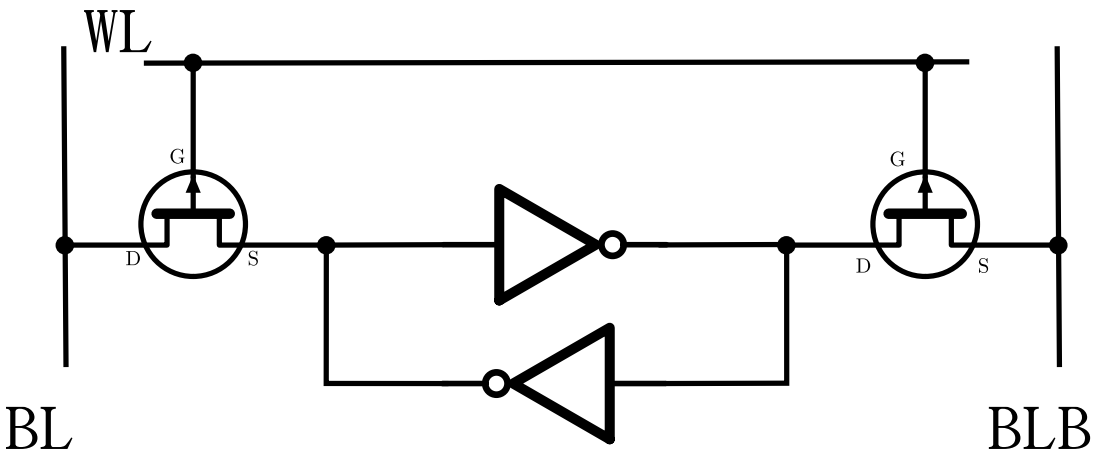


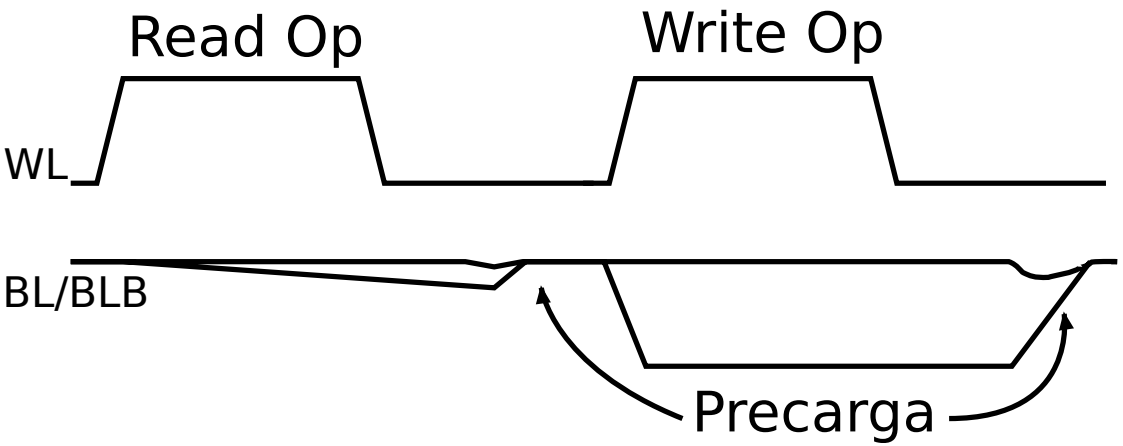


- Como el estado inicial de tensión en las Bit lines era alto, el pull up ya era un hecho de modo que interesa analizar el pull down en la Bit Line complementaria.
- En la Bit line que experimenta el pull down, la tensión cae de manera mas o menos débil (primer parte del gráfico)

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

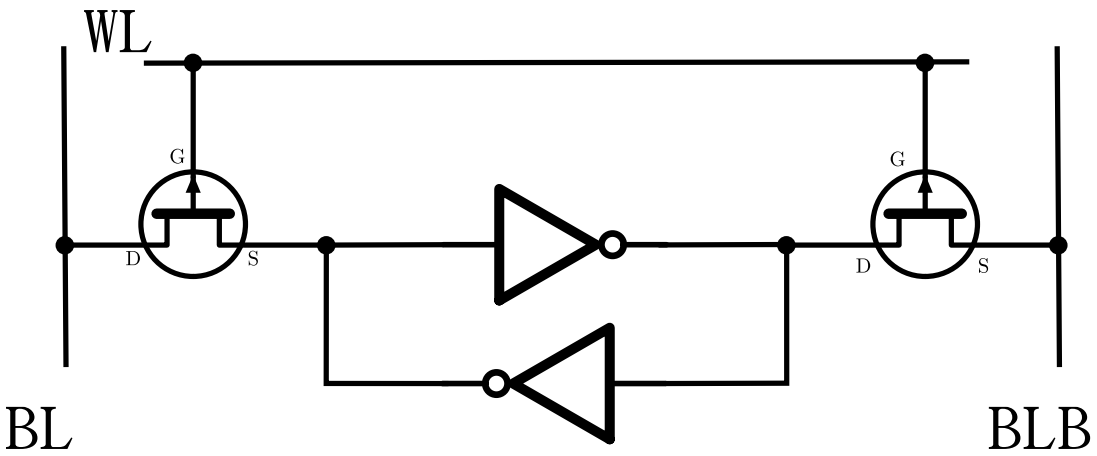


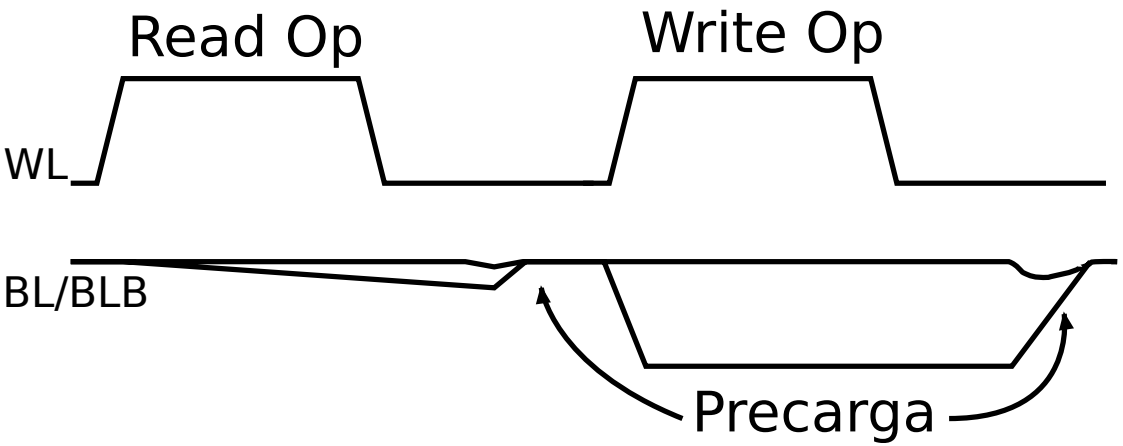


- Depende de la capacidad de difusión del transistor de acceso, y de los conectores empleados.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:



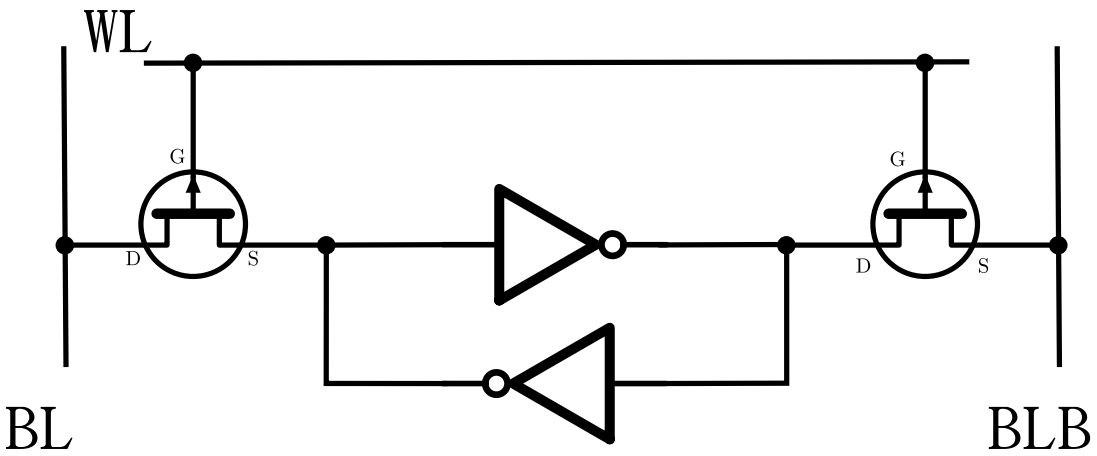


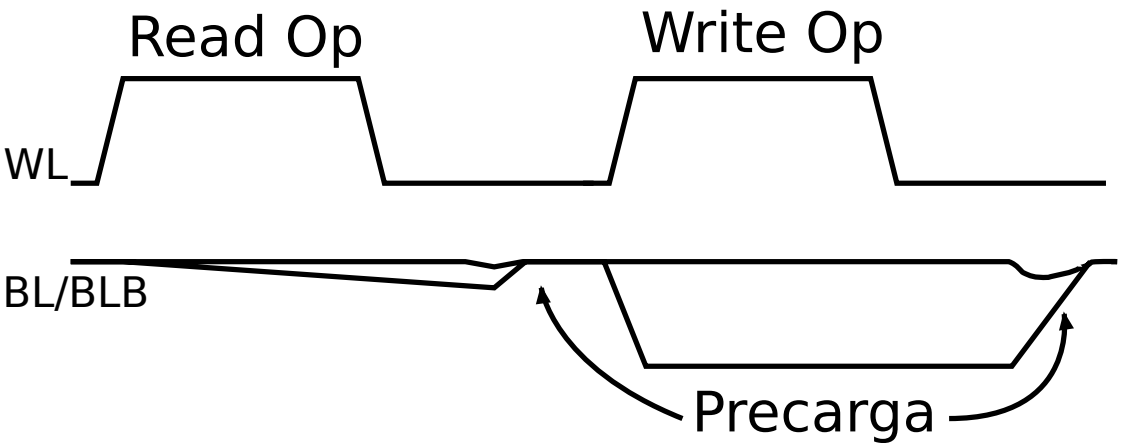
- Depende de la capacidad de difusión del transistor de acceso, y de los conectores empleados.
- El pull down continúa hasta que se desactive la **WL**

.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

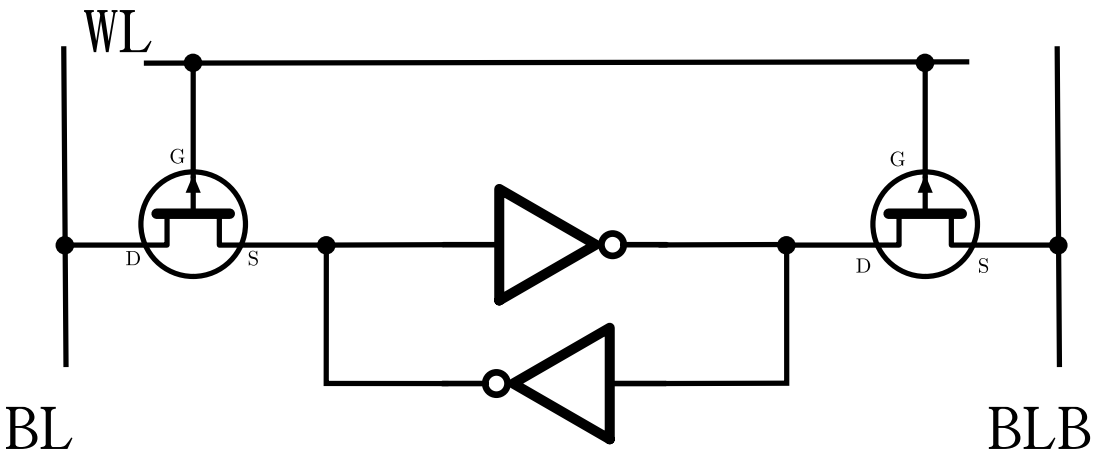


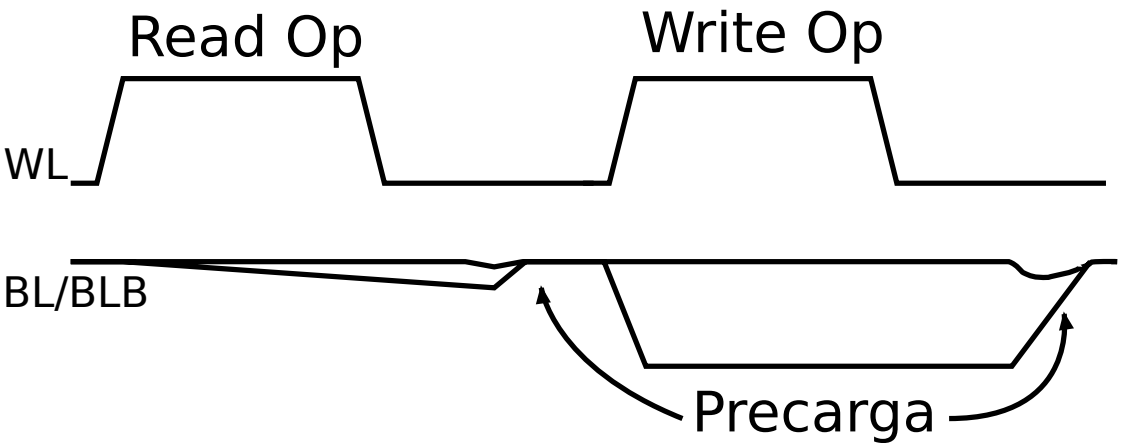


- Depende de la capacidad de difusión del transistor de acceso, y de los conectores empleados.
- El pull down continúa hasta que se desactive la **WL** .
- Luego dependiendo del circuito de precarga el bitline se precarga o permanece constante (en el diagrama se representa ésto último).

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

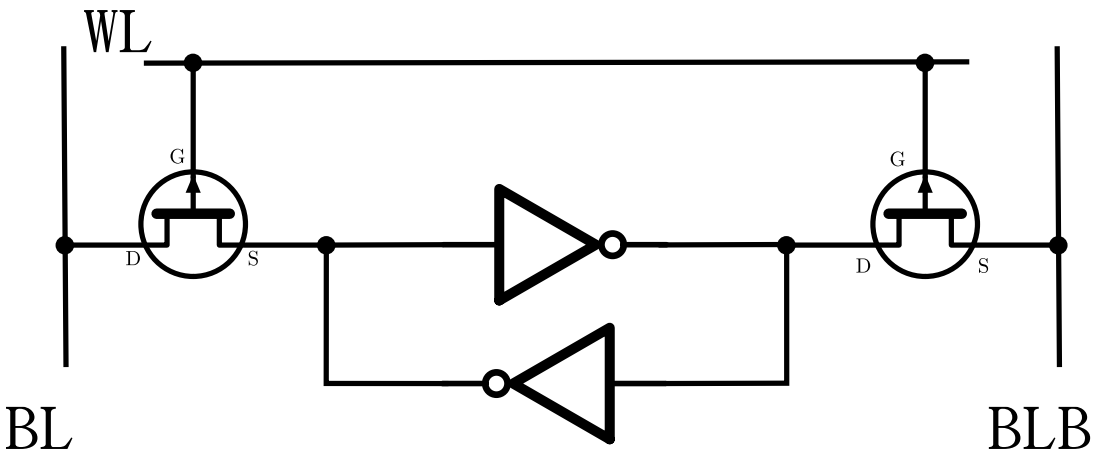


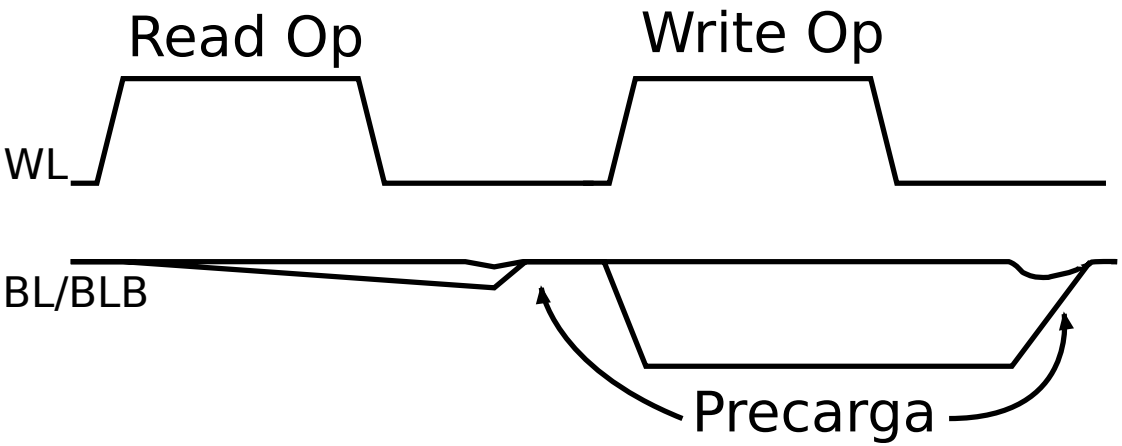


- Durante el pull down Read Mux selecciona la columna deseada y conecta las líneas **BL** y **BLB** en la entrada del Amplificador de Detección (Sense Amp)

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

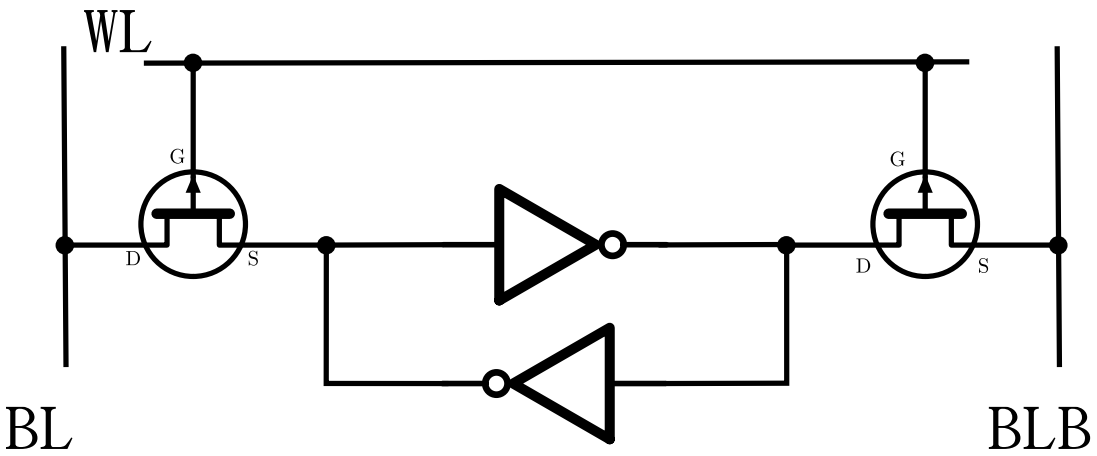


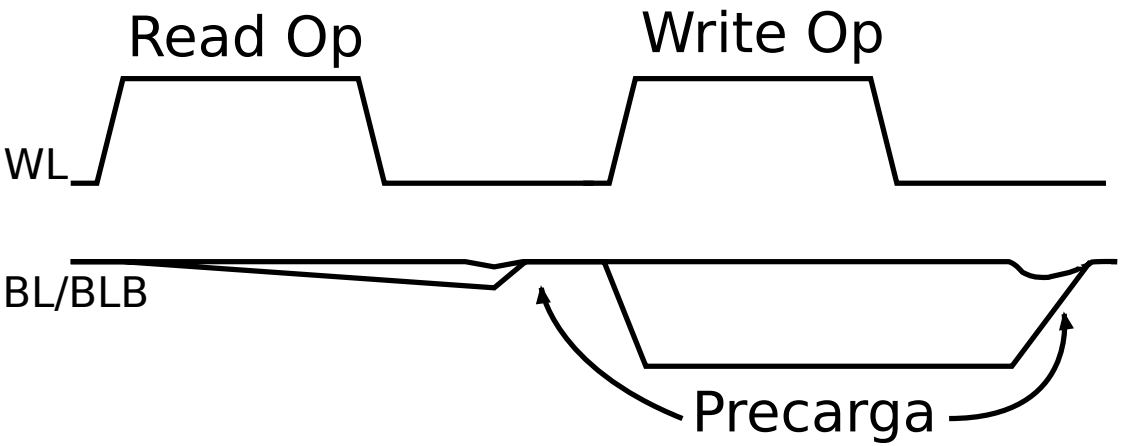


- Durante el pull down Read Mux selecciona la columna deseada y conecta las líneas **BL** y **BLB** en la entrada del Amplificador de Detección (Sense Amp)
- Así se detecta el estado lógico.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

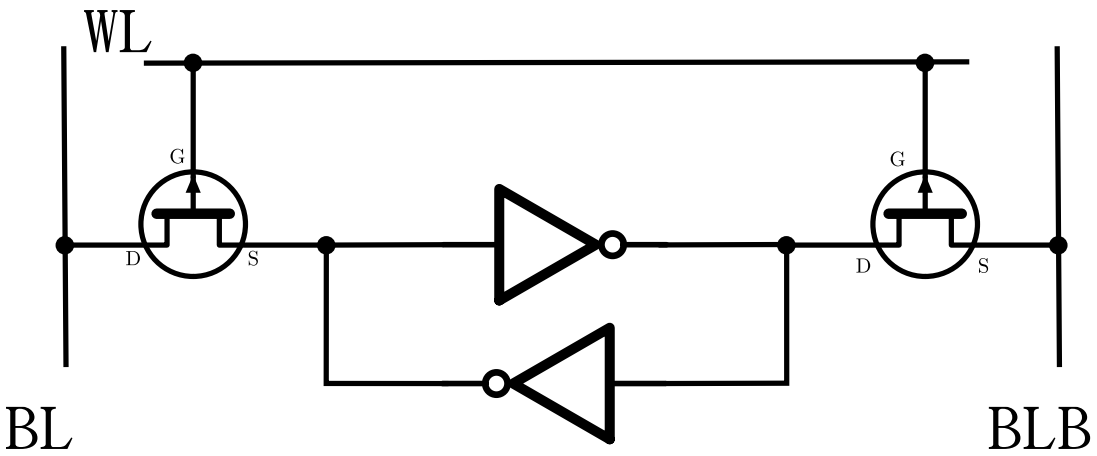


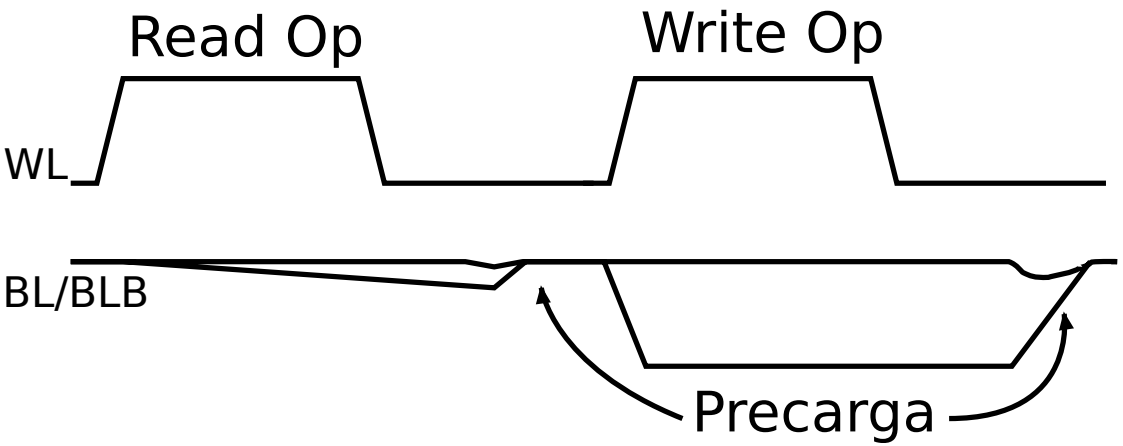


- Durante el pull down Read Mux selecciona la columna deseada y conecta las líneas **BL** y **BLB** en la entrada del Amplificador de Detección (Sense Amp)
- Así se detecta el estado lógico.
- Antes de la nueva operación el circuito de precarga y ecualización pone ambas líneas a V_{DD} para estar listo nuevamente.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

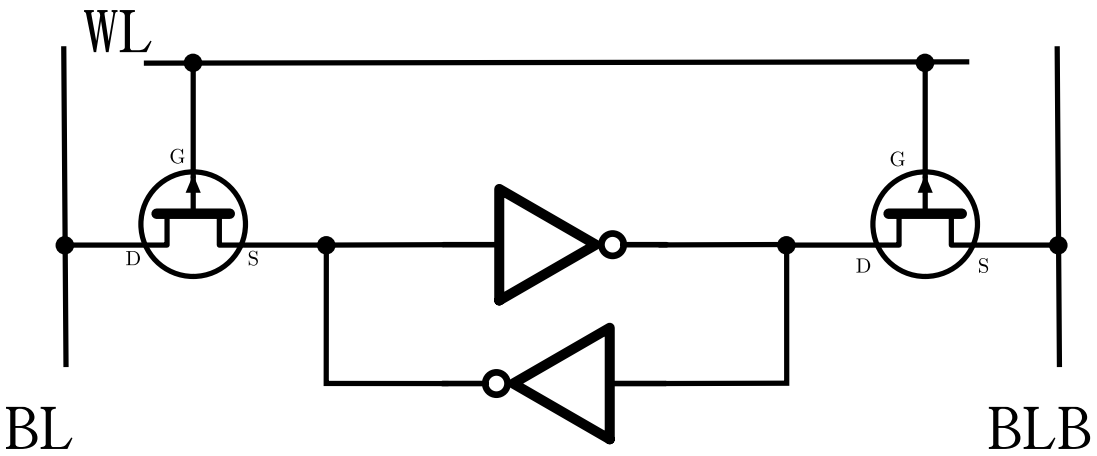


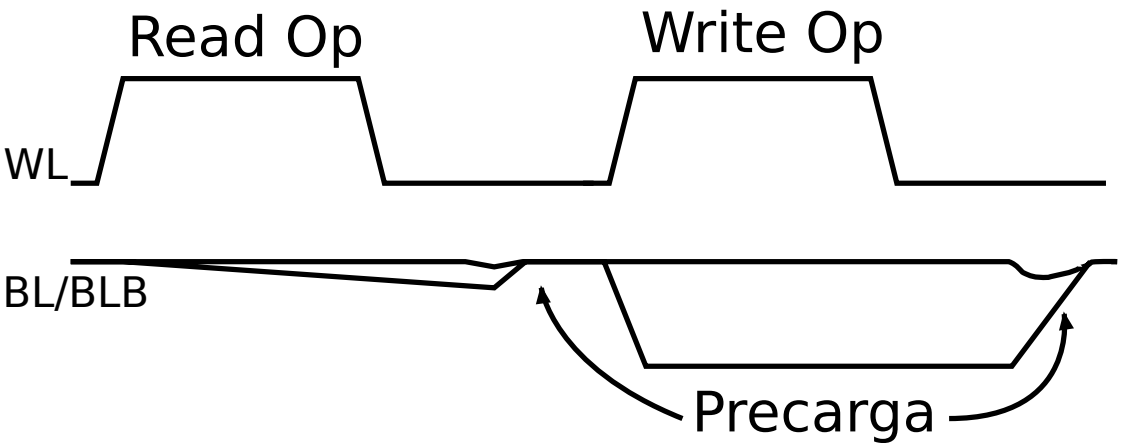


- En una operación de escritura, se habilitan nuevamente los transistores de acceso de todas las celdas conectas a la correspondiente **`WL`**.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

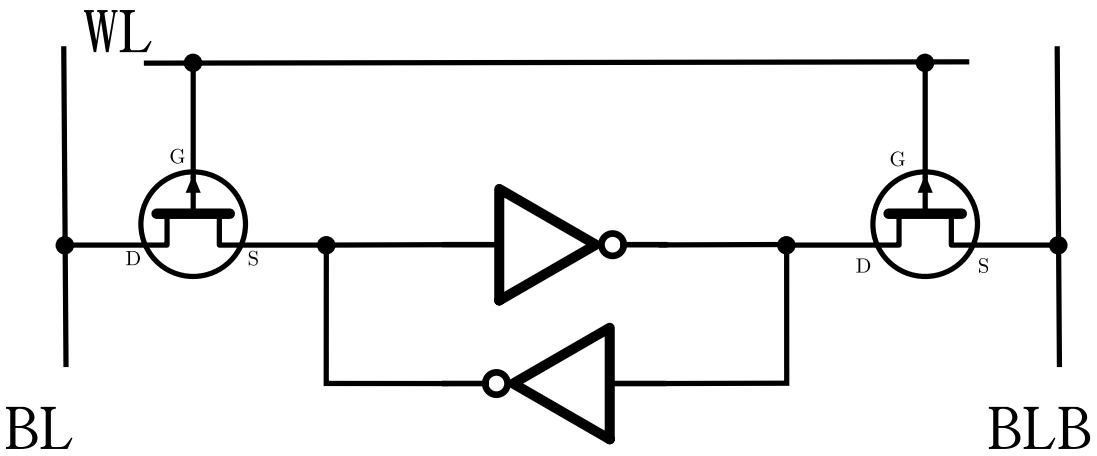


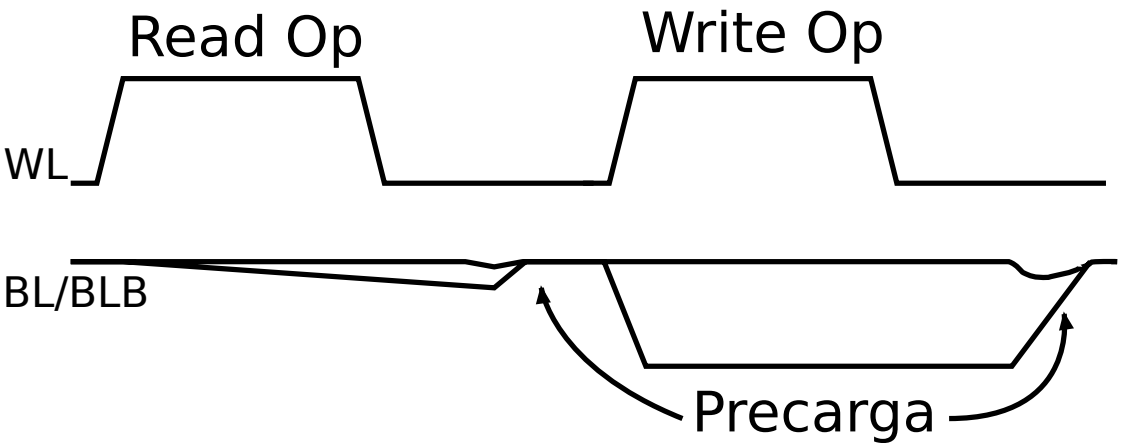


- En una operación de escritura, se habilitan nuevamente los transistores de acceso de todas las celdas conectas a la correspondiente **WL**.
- Esta vez el amplificador de escritura impone una tensión diferencial entre **BL** y **BLB**.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:

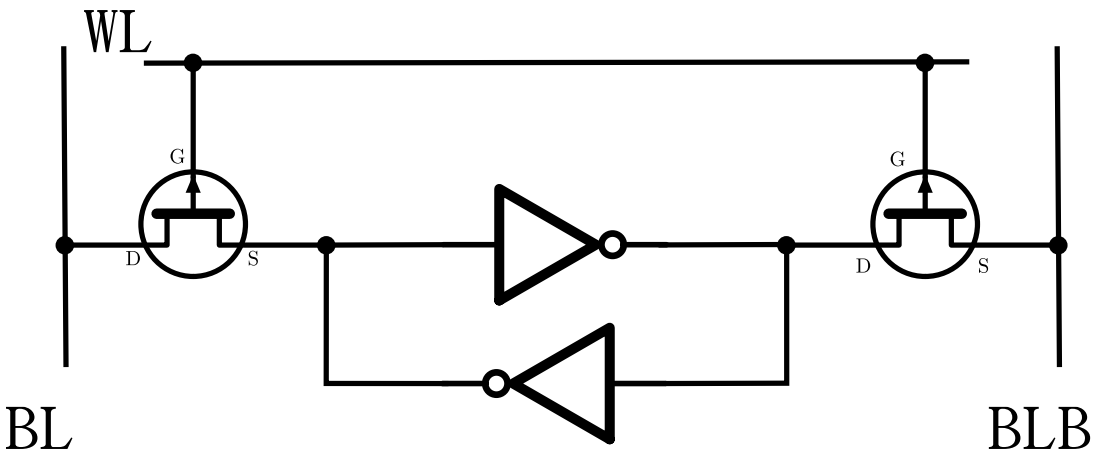


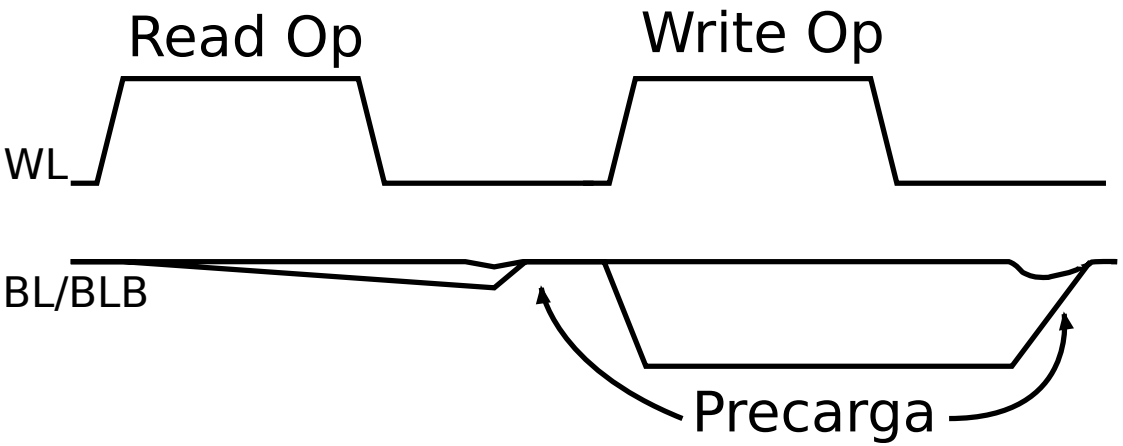


- La mayor capacitancia de las Bitlines junto con la mayor corriente de salida del amplificador de escritura fuerzan la tensión diferencial y por ende el estado lógico en la celda.

Diagrama general de circuitos periféricos

Lo anterior se termina de ver en el circuito equivalente de la celda a continuación:



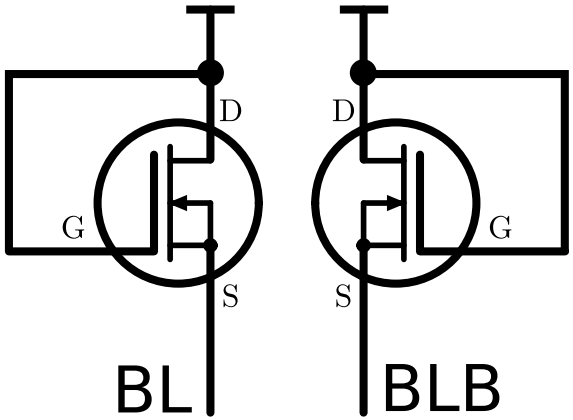


- La mayor capacitancia de las Bitlines junto con la mayor corriente de salida del amplificador de escritura fuerzan la tensión diferencial y por ende el estado lógico en la celda.
- Nuevamente se precargan las Bitlines antes de la siguiente operación.

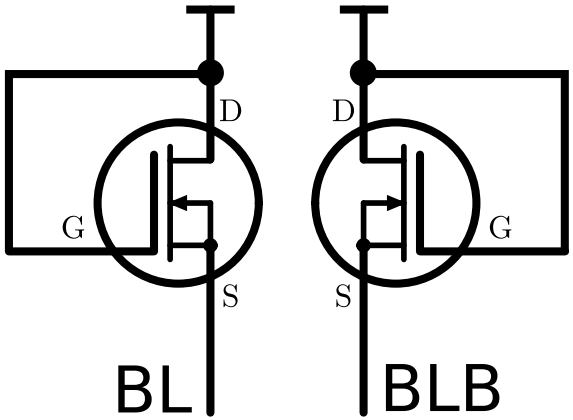
Circuitos de Precarga y Ecualización

Los circuitos de Precarga y Equalización son de lo mas diversos y responden a diferentes métodos de diseño.
Nos vamos a centrar en un par de alternativas de cada uno que permitirán cubrir la idea general de su función dentro de una memoria SRAM

Precarga y Ecualización #1 NMOS como diodo

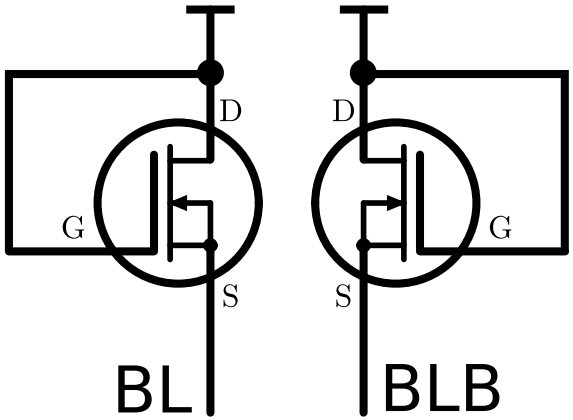


Precarga y Ecualización #1 NMOS como diodo



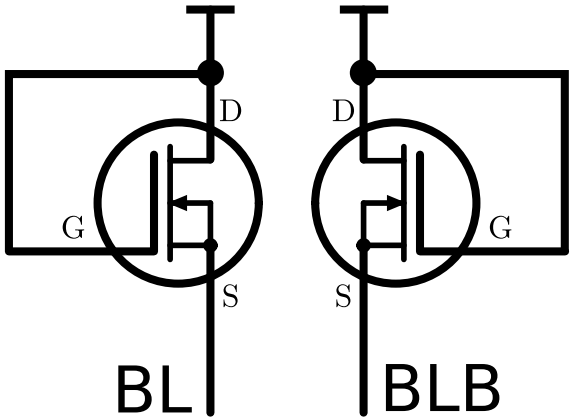
- Utiliza un par de NMOS en configuración diodo, cada uno conectado a cada bitline.

Precarga y Ecualización #1 NMOS como diodo



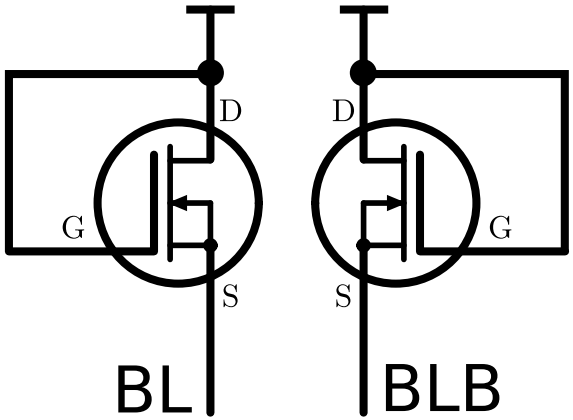
- Utiliza un par de NMOS en configuración diodo, cada uno conectado a cada bitline.
- Fuerza permanentemente un estado V_{DD}-V_t en las bitlines.

Precarga y Ecualización #1 NMOS como diodo



- Utiliza un par de NMOS en configuración diodo, cada uno conectado a cada bitline.
- Fuerza permanentemente un estado V_{DD}-V_t en las bitlines.
- Este pull up permanente es a la vez su fortaleza y su limitación.

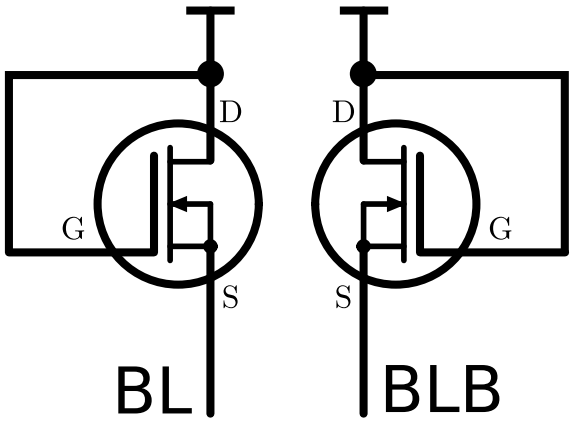
Precarga y Ecualización #1 NMOS como diodo



- Utiliza un par de NMOS en configuración diodo, cada uno conectado a cada bitline.
- Fuerza permanentemente un estado V_{DD}-V_t en las bitlines.
- Este pull up permanente es a la vez su fortaleza y su limitación.

- Por un lado provee una solución con mínima complejidad circuital. Ni siquiera requiere control de recarga

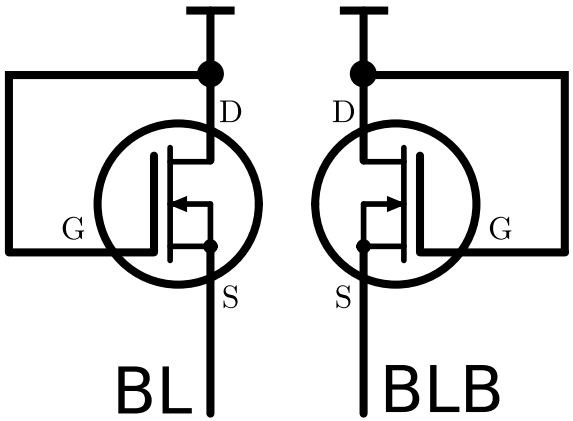
Precarga y Ecualización #1 NMOS como diodo



- Utiliza un par de NMOS en configuración diodo, cada uno conectado a cada bitline.
- Fuerza permanentemente un estado V_{DD}-V_t en las bitlines.
- Este pull up permanente es a la vez su fortaleza y su limitación.

- Por un lado provee una solución con mínima complejidad circuital. Ni siquiera requiere control de recarga
- Por otra parte al haber un camino de corriente constante desde V_{DD} hacia la bitline, tiende a generar demoras para el establecimiento de la tensión diferencial cuando se selecciona la `WL`.

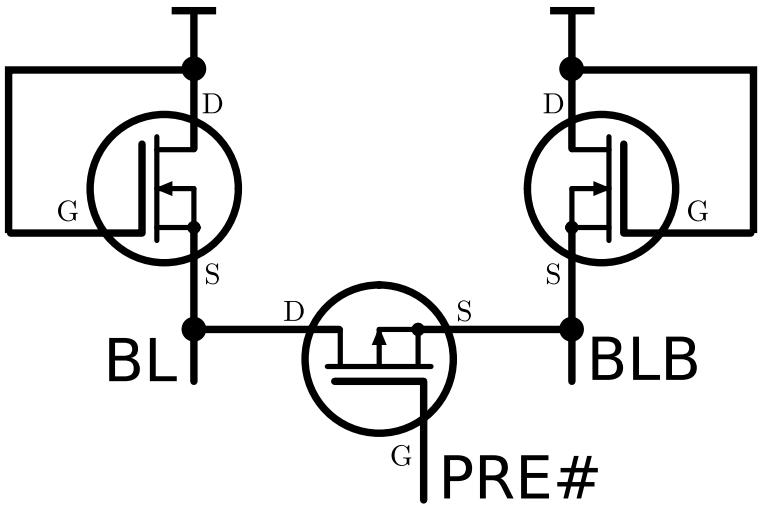
Precarga y Ecualización #1 NMOS como diodo



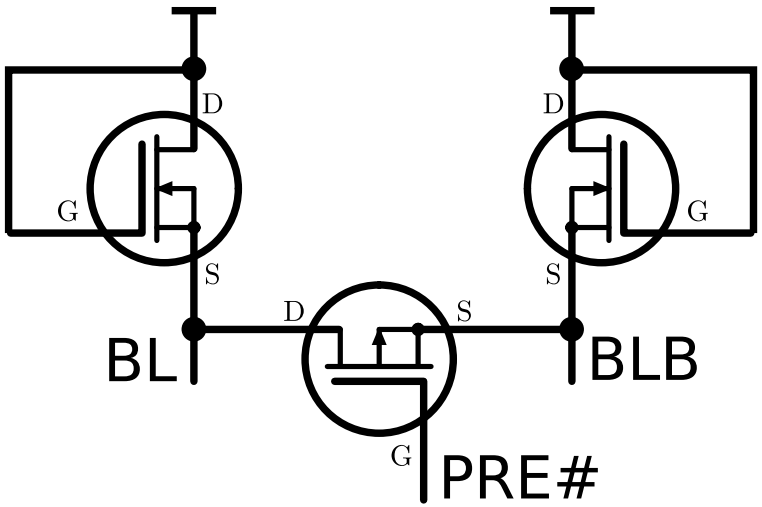
- Utiliza un par de NMOS en configuración diodo, cada uno conectado a cada bitline.
- Fuerza permanentemente un estado V_{DD}-V_t en las bitlines.
- Este pull up permanente es a la vez su fortaleza y su limitación.

- Por un lado provee una solución con mínima complejidad circuital. Ni siquiera requiere control de recarga
- Por otra parte al haber un camino de corriente constante desde V_{DD} hacia la bitline, tiende a generar demoras para el establecimiento de la tensión diferencial cuando se selecciona la `WL`.
- Introduce demoras en las lecturas y disipación de potencia en las escrituras ya que una de las bitlines resiste el pull up al que la fuerza el Amplificador de Detección (Sense Amp).

Precarga y Ecualización #2 PMOS en puente

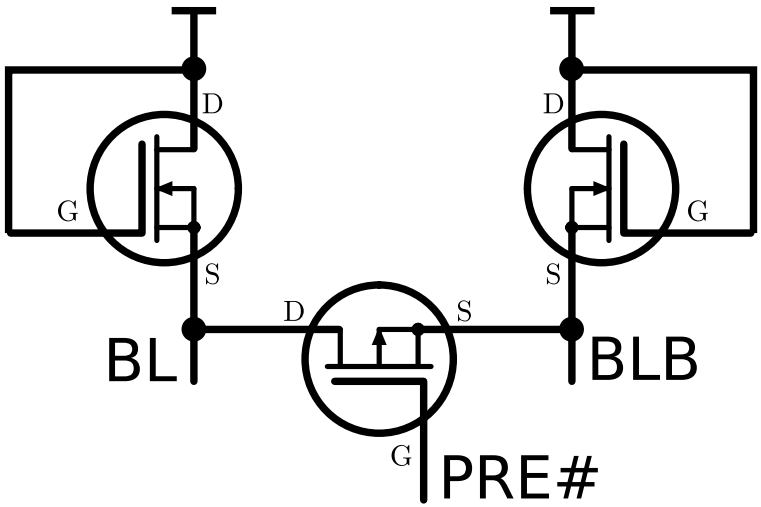


Precarga y Ecualización #2 PMOS en puente



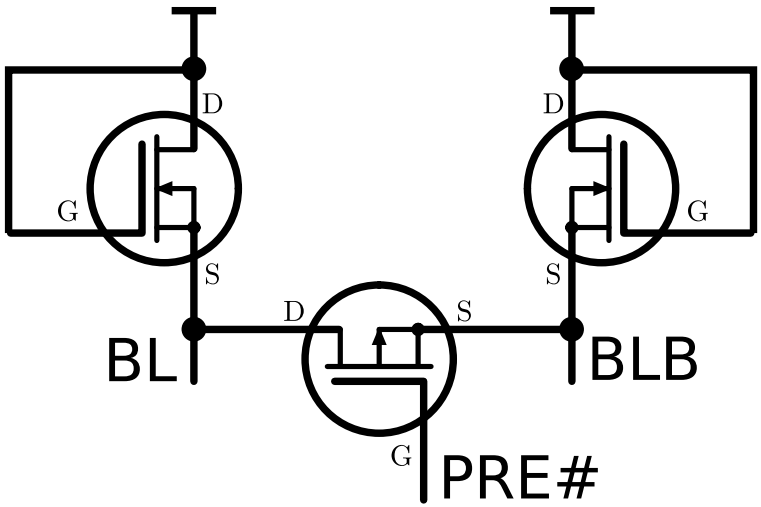
- La función del PMOS en puente es ecualizar la tensión entre las dos líneas siempre que se lo active.

Precarga y Ecualización #2 PMOS en puente



- La función del PMOS en puente es ecualizar la tensión entre las dos líneas siempre que se lo active.
- Importante si los NMOS tienen carga activa, para prevenir desbalances debidos a variaciones en la V_t de los NMOS.

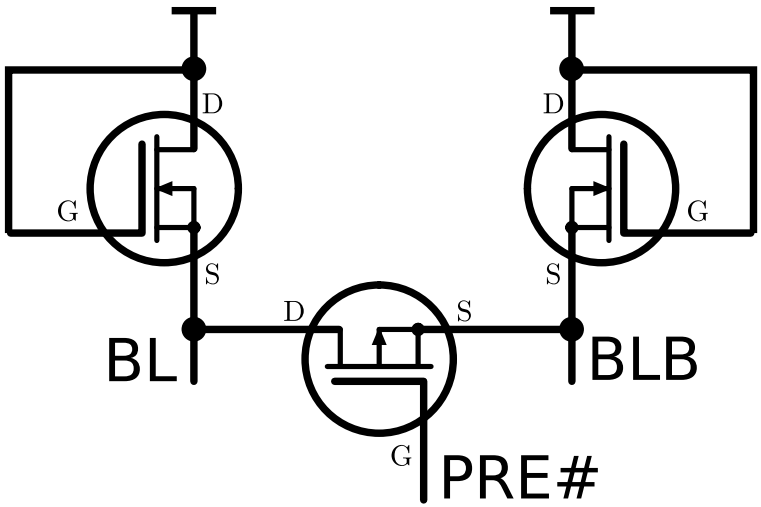
Precarga y Ecualización #2 PMOS en puente



- La función del PMOS en puente es ecualizar la tensión entre las dos líneas siempre que se lo active.
- Importante si los NMOS tienen carga activa, para prevenir desbalances debidos a variaciones en la V_t de los NMOS.

- El desbalance en las V_t causa diferencias en los niveles de precarga afectando a los Amplificadores de Detección (Sense Amp).

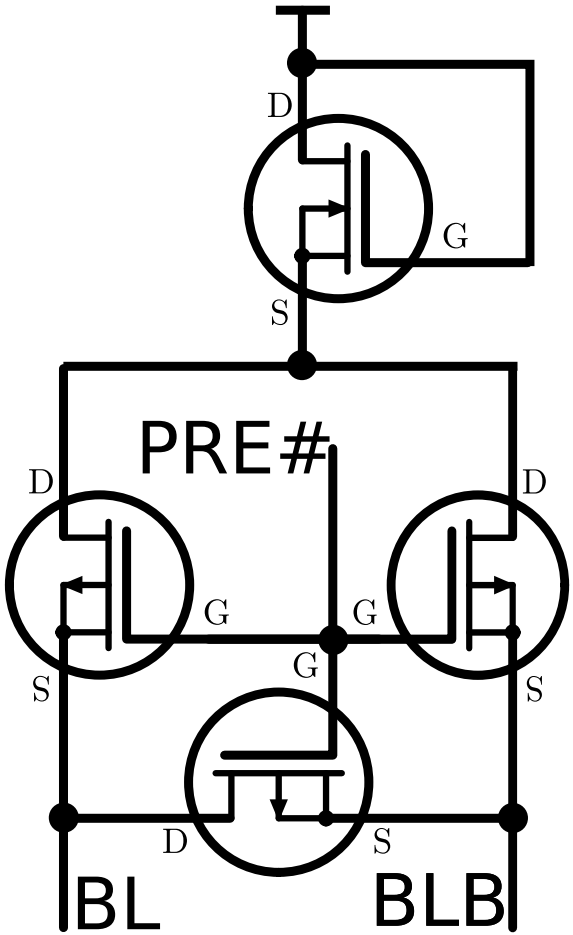
Precarga y Ecualización #2 PMOS en puente



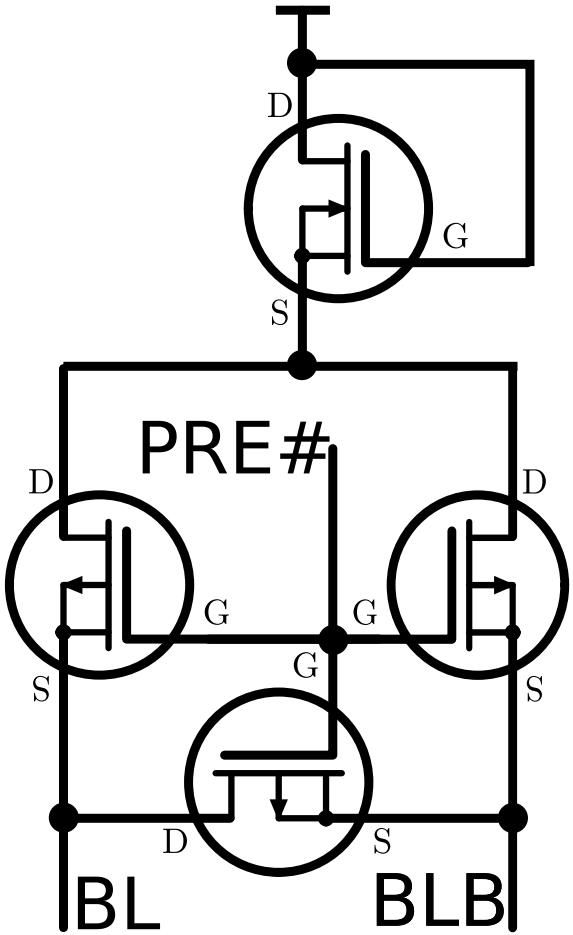
- La función del PMOS en puente es ecualizar la tensión entre las dos líneas siempre que se lo active.
- Importante si los NMOS tienen carga activa, para prevenir desbalances debidos a variaciones en la V_t de los NMOS.

- El desbalance en las V_t causa diferencias en los niveles de precarga afectando a los Amplificadores de Detección (Sense Amp).
- Consideremos que la configuración diodo puede reemplazarse por cargas activas PMOS si el desfasaje de V_t no es aceptable.

Precarga y Ecualización #3

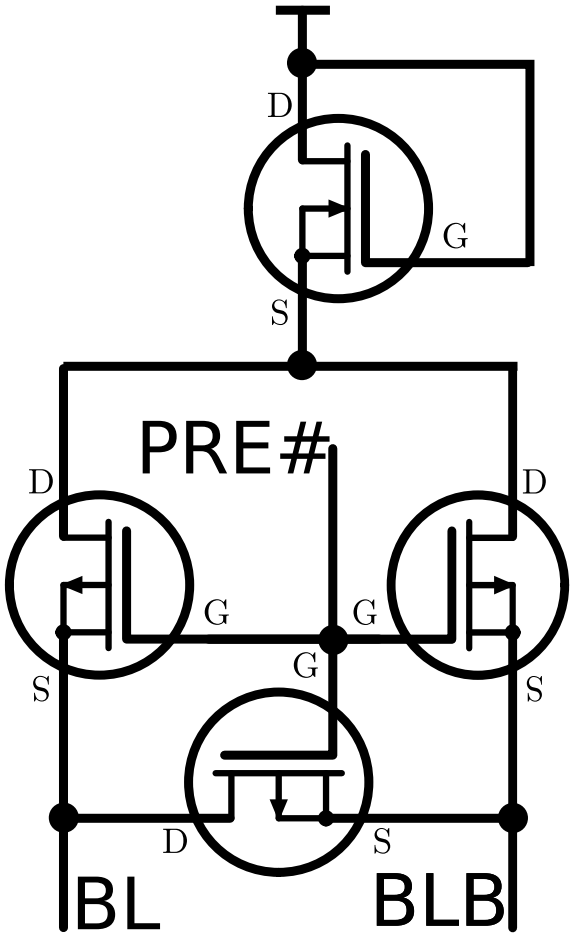


Precarga y Ecualización #3



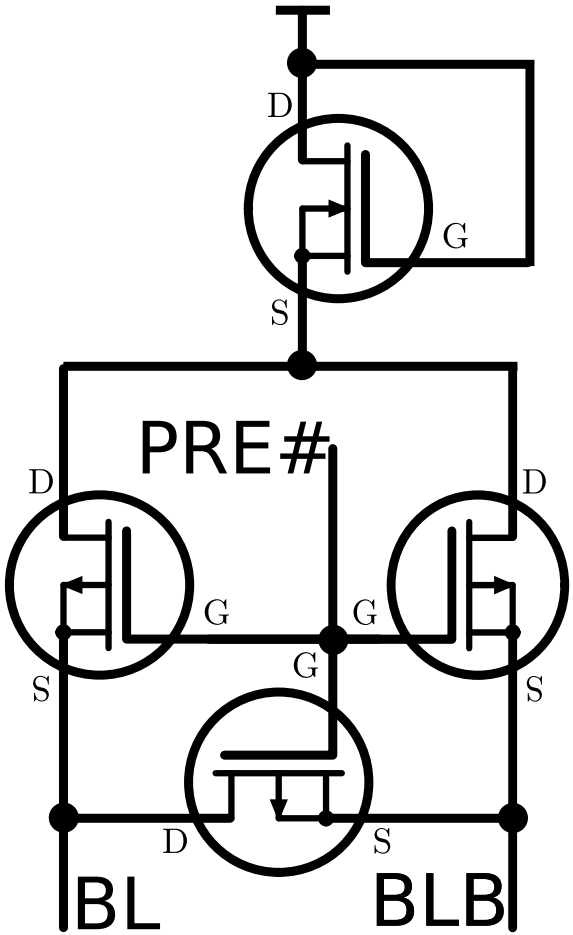
- Usa un NMOS con para establecer una precarga de V_{DD}-V_t, combinado con transistores PMOS de paso para conectar las Bitlines solo durante la precarga.

Precarga y Ecualización #3



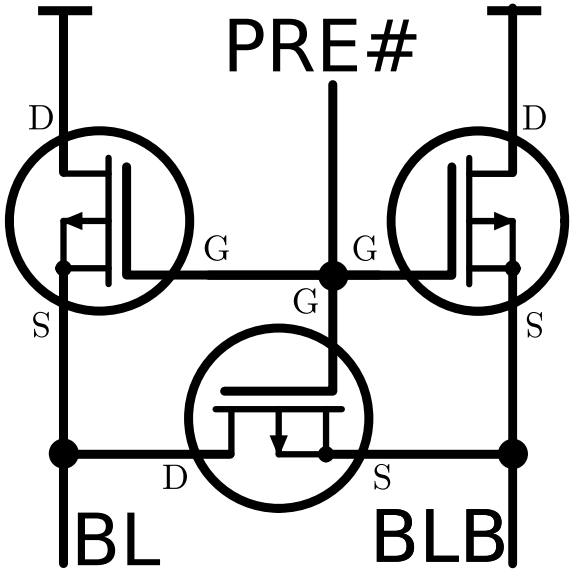
- Usa un NMOS con para establecer una precarga de V_{DD}-V_t, combinado con transistores PMOS de paso para conectar las Bitlines solo durante la precarga.
- Configuración apta para tensiones de alimentación moderadas (p.ej. 3.3.V), o cuando los amplificadores de Detección operan con tensiones de modo común por debajo de las de alimentación.

Precarga y Ecualización #3

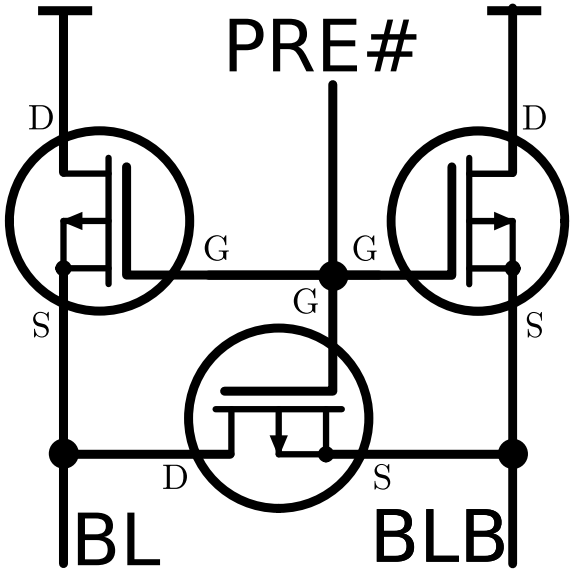


- Usa un NMOS con para establecer una precarga de V_{DD}-V_t, combinado con transistores PMOS de paso para conectar las Bitlines solo durante la precarga.
- Configuración apta para tensiones de alimentación moderadas (p.ej. 3.3.V), o cuando los amplificadores de Detección operan con tensiones de modo común por debajo de las de alimentación.
- Debido al proceso de fabricación, a medida que la tensión de alimentación disminuye, las dispersiones en las V_t se tornan mas evidentes y hacen a este circuito poco propiado.

Precarga y Ecualización #4

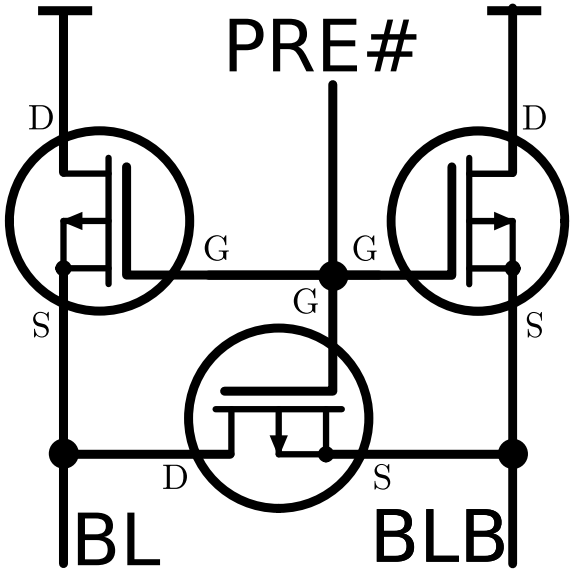


Precarga y Ecualización #4



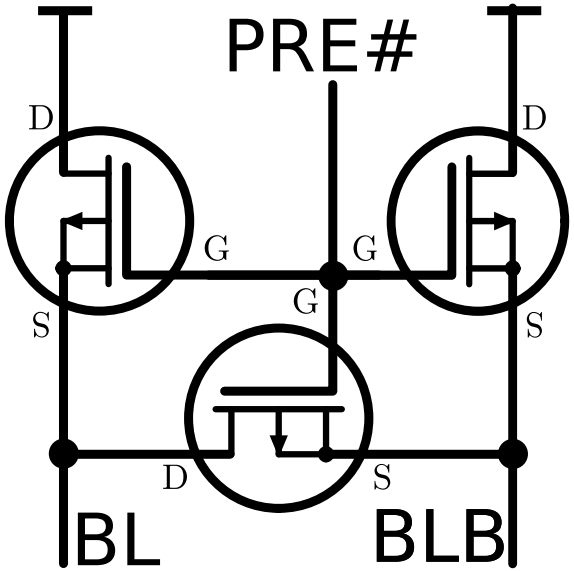
- Es la implementación mas popular.

Precarga y Ecualización #4



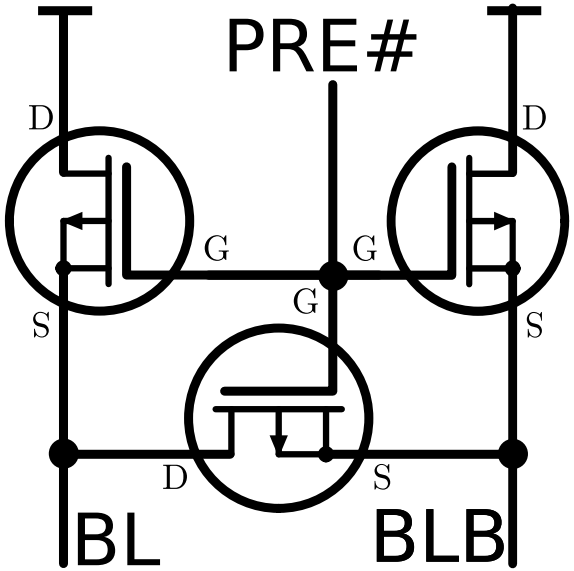
- Es la implementación mas popular.
- Durante las precargas y ecualizaciones los tres PMOS se encuentran habilitados, presentando baja impedancia para ambas bitlines tanto a V_{DD} como entre sí.

Precarga y Ecualización #4



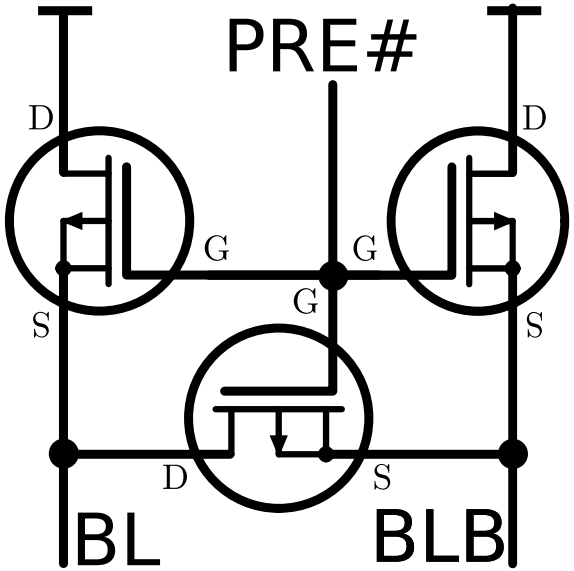
- Es la implementación mas popular.
- Durante las precargas y ecualizaciones los tres PMOS se encuentran habilitados, presentando baja impedancia para ambas bitlines tanto a V_{DD} como entre sí.
- Una vez precargadas las líneas los PMOS se apagan presentando alta impedancia, aislando a ambas respecto de V_{DD}.

Precarga y Ecualización #4



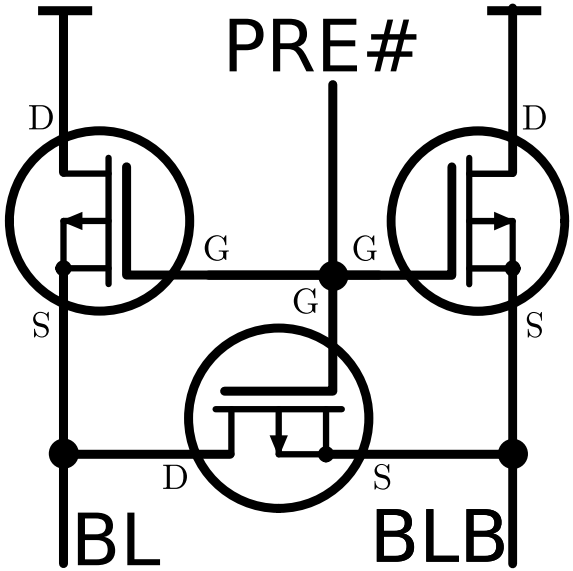
- Es la implementación mas popular.
- Durante las precargas y ecualizaciones los tres PMOS se encuentran habilitados, presentando baja impedancia para ambas bitlines tanto a V_{DD} como entre sí.
- Una vez precargadas las líneas los PMOS se apagan presentando alta impedancia, aislando a ambas respecto de V_{DD}.
- Acelera la obtención de la tensión diferencial ya que el pull down solo tiene que descargar la capacitancia existente en la bitline.

Precarga y Ecualización #4



- Es la implementación mas popular.
- Durante las precargas y ecualizaciones los tres PMOS se encuentran habilitados, presentando baja impedancia para ambas bitlines tanto a V_{DD} como entre sí.
- Una vez precargadas las líneas los PMOS se apagan presentando alta impedancia, aislando a ambas respecto de V_{DD}.
- Acelera la obtención de la tensión diferencial ya que el pull down solo tiene que descargar la capacitancia existente en la bitline.
- Durante las escrituras no se malgasta energía para precargar este circuito ya que no afecta la operación del amplificador de escritura.

Precarga y Ecualización #4

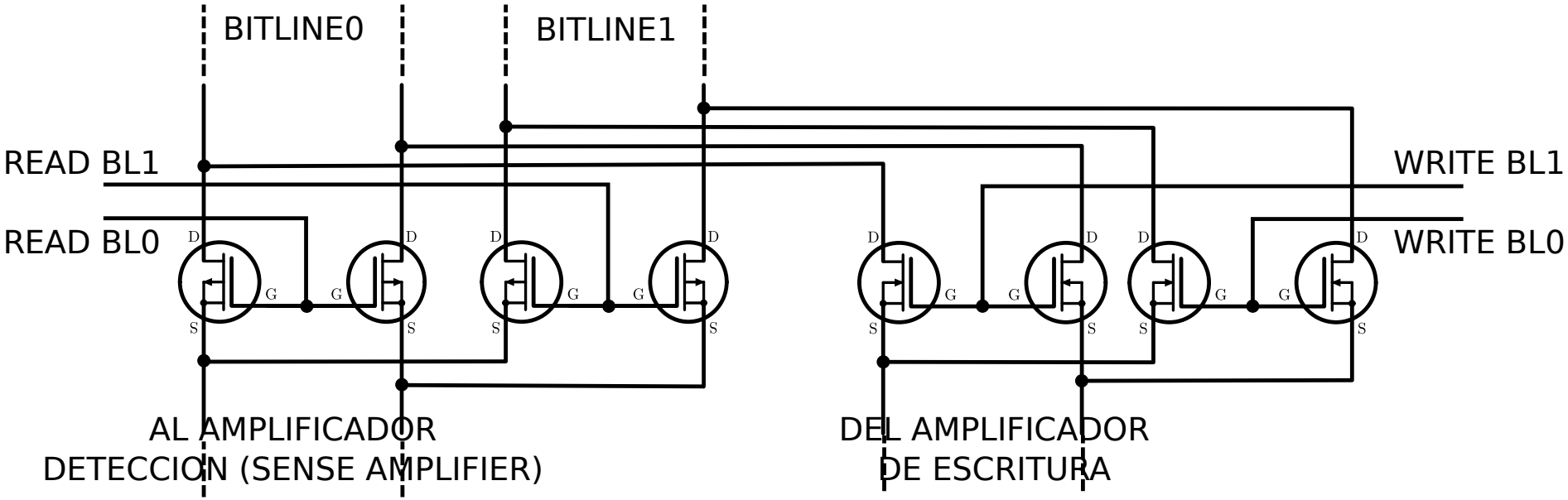


- Es la implementación mas popular.
- Durante las precargas y ecualizaciones los tres PMOS se encuentran habilitados, presentando baja impedancia para ambas bitlines tanto a V_{DD} como entre sí.
- Una vez precargadas las líneas los PMOS se apagan presentando alta impedancia, aislando a ambas respecto de V_{DD}.

- Acelera la obtención de la tensión diferencial ya que el pull down solo tiene que descargar la capacitancia existente en la bitline.
- Durante las escrituras no se malgasta energía para precargar este circuito ya que no afecta la operación del amplificador de escritura.
- Desvetajas: Disipa potencia al conmutar los transistores lo cual está nivelado por la que se ahorra en la precarga. Además agrega complejidad al control de disparo de la precarga.

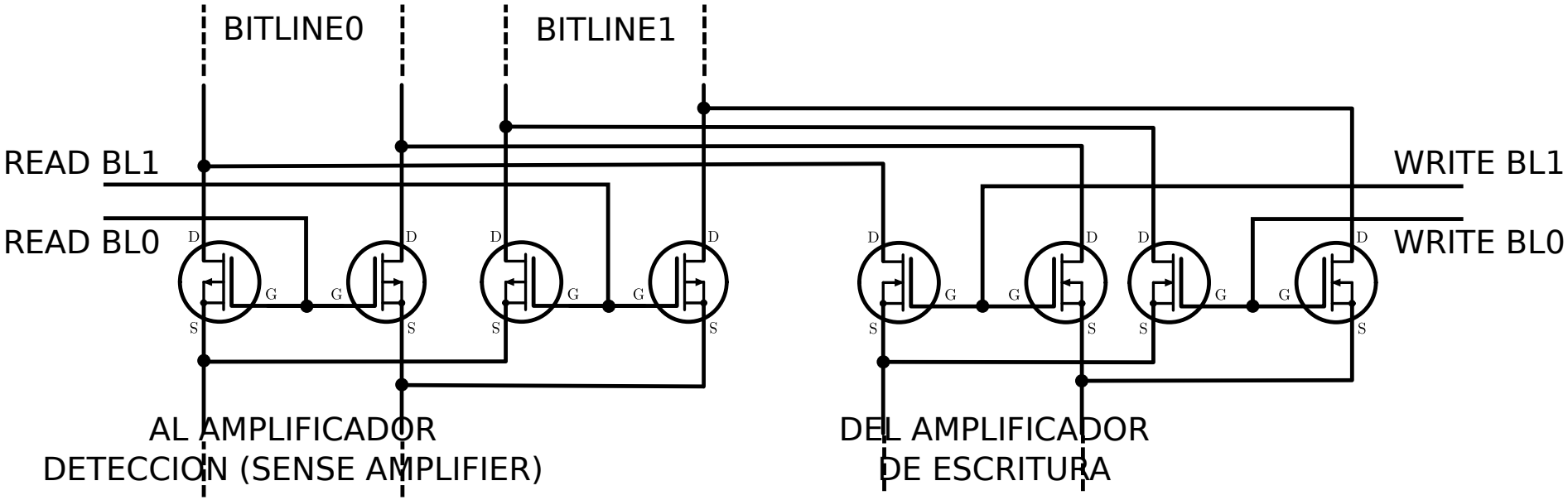
Multiplexores de Lectura/Escritura

Multiplexores de Lectura/Escritura



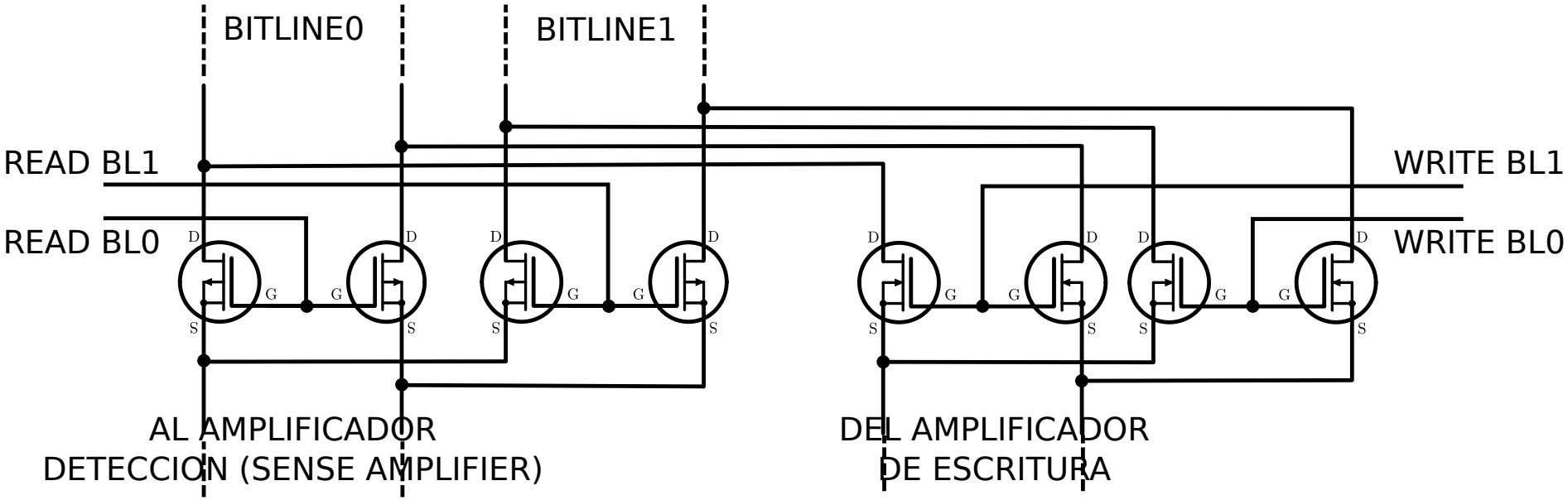
- Permiten compartir un Amplificador de Detección (Sense Amp), o uno de Escritura, entre múltiples pares de Bitlines.

Multiplexores de Lectura/Escritura



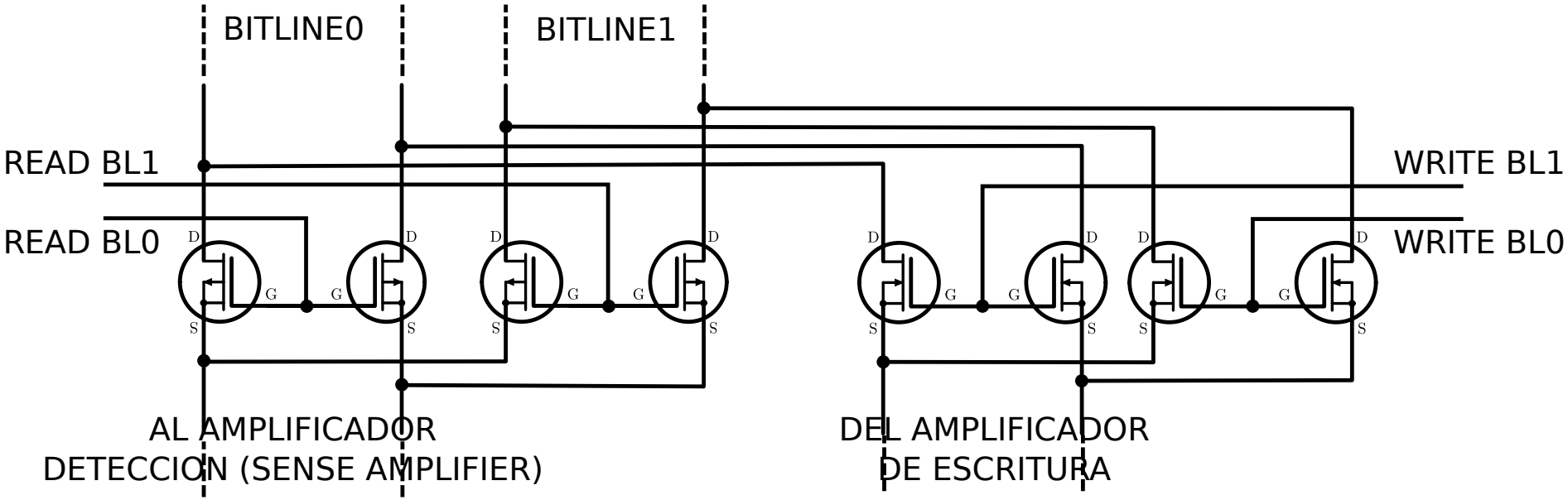
- Permiten compartir un Amplificador de Detección (Sense Amp), o uno de Escritura, entre múltiples pares de Bitlines.
- Se pueden organizar para compartir múltiples pares de Bitlines con un MUX.

Multiplexores de Lectura/Escritura



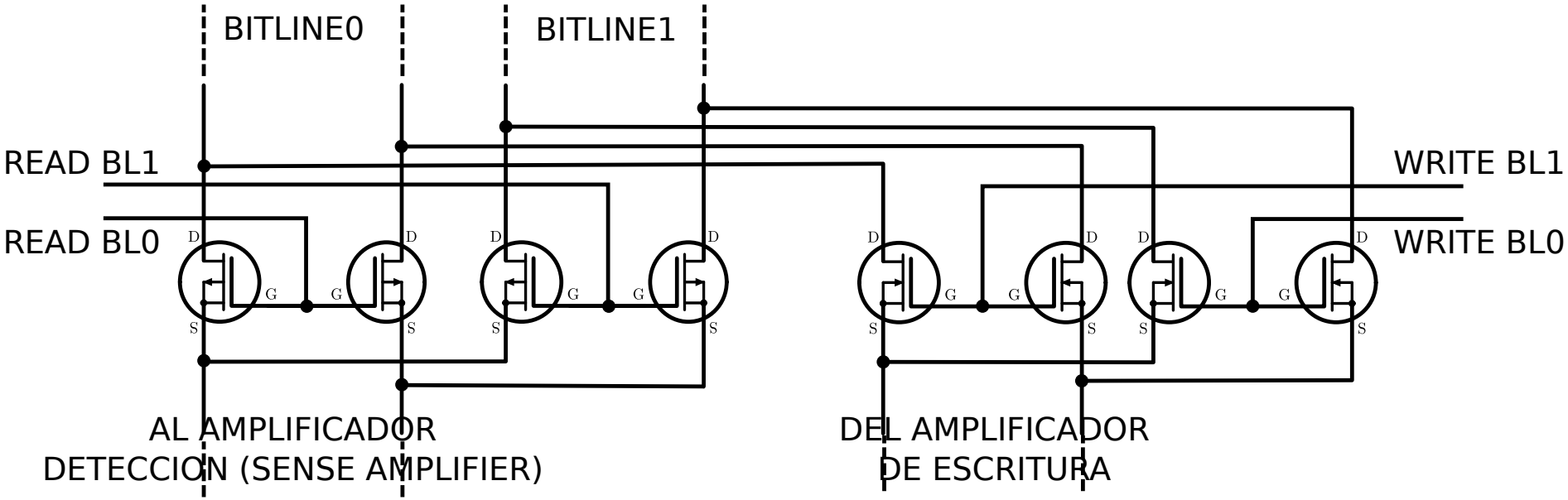
- Permiten compartir un Amplificador de Detección (Sense Amp), o uno de Escritura, entre múltiples pares de Bitlines.
- Se pueden organizar para compartir múltiples pares de Bitlines con un MUX.
- Para los de lectura se usan PMOS porque sus tensiones de modo común son próximas a la tensión correspondiente al estado alto, lo cual resulta en una mejor transconductancia del dispositivo.

Multiplexores de Lectura/Escritura



- Para las escrituras una de las bitlines será descargada. Para esto se utilizan transistores NMOS Ya que al activarse pondrán en su salida un valor de tensión muy bajo lo cual significará un fuerte pull down.

Multiplexores de Lectura/Escritura



- Para las escrituras una de las bitlines será descargada. Para esto se utilizan transistores NMOS Ya que al activarse pondrán en su salida un valor de tensión muy bajo lo cual significará un fuerte pull down.
- En ambos casos solo se utiliza un solo transistor ya que su complementario no se utilizará de manera efectiva.

Amplificadores de Detección

Amplificadores de Detección

La generación de una tensión diferencial entre un par de bitlines fuerza al dispositivo que integra la celda de memoria a descargarse a través de una capacidad cuyo valor generará una demora que perjudicaría el tiempo de acceso para leer el dispositivo.

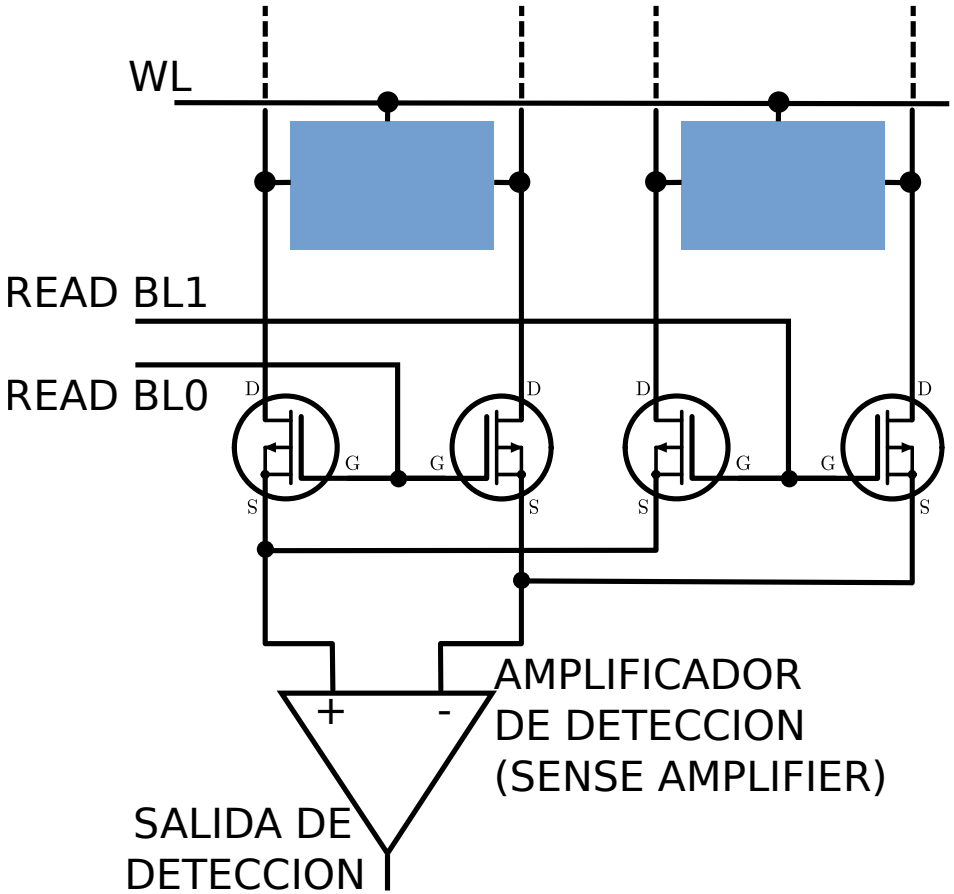
Amplificadores de Detección

La generación de una tensión diferencial entre un par de bitlines fuerza al dispositivo que integra la celda de memoria a descargarse a través de una capacidad cuyo valor generará una demora que perjudicaría el tiempo de acceso para leer el dispositivo.
El Amplificador de Detección tiene la función de resolver este problema acelerando el proceso de obtención del valor que se está leyendo.

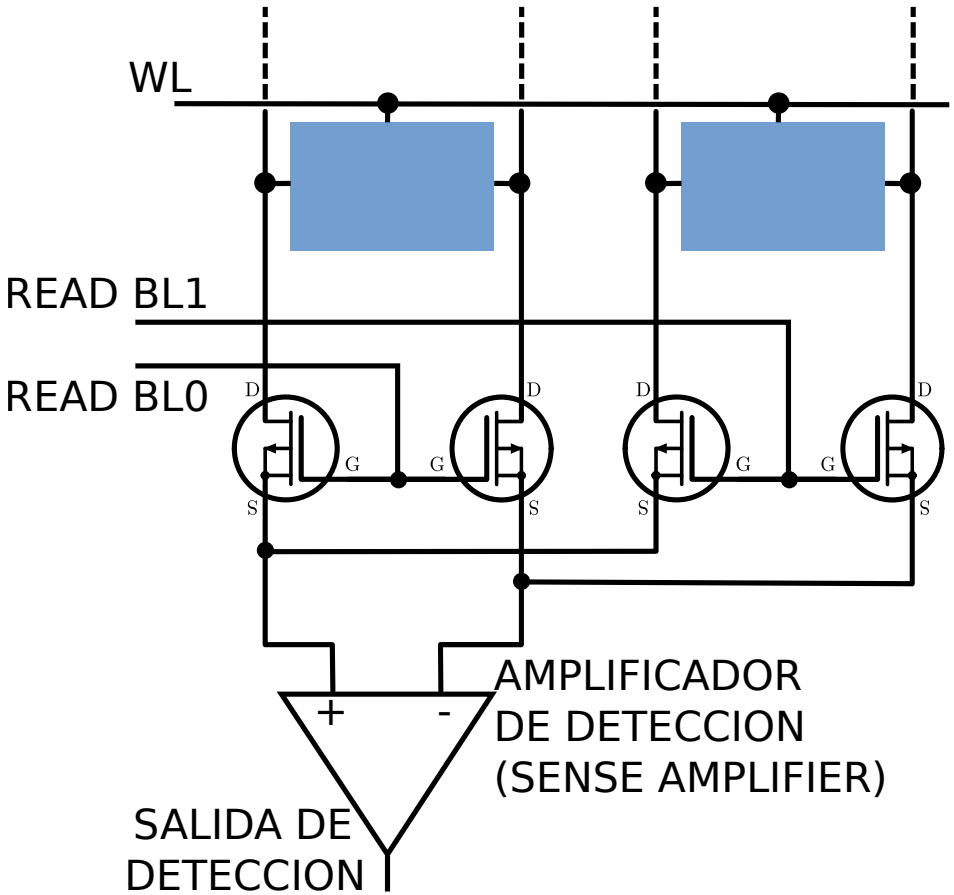
Amplificadores de Detección

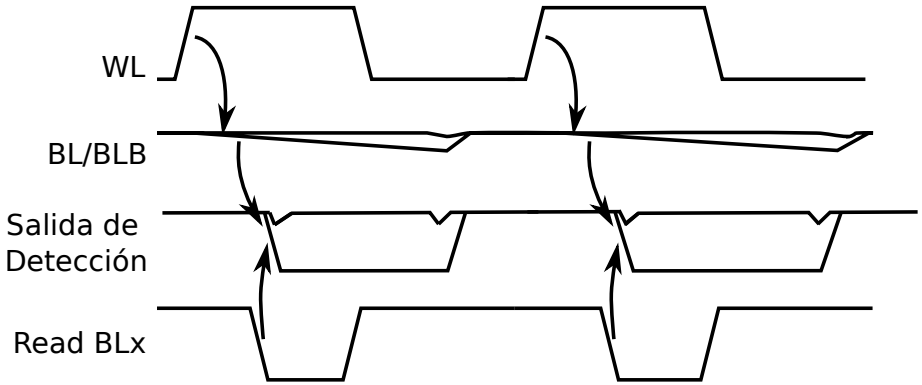
La generación de una tensión diferencial entre un par de bitlines fuerza al dispositivo que integra la celda de memoria a descargarse a través de una capacidad cuyo valor generará una demora que perjudicaría el tiempo de acceso para leer el dispositivo.
El Amplificador de Detección tiene la función de resolver este problema acelerando el proceso de obtención del valor que se está leyendo.
Su nombre, ***Sense Amplifier***, proviene de los circuitos de lectura de las viejas memorias magnéticas. El desafío por entonces era lograr un amplificador "sensitivo" de la magnetización de la celda.

Amplificador de Detección

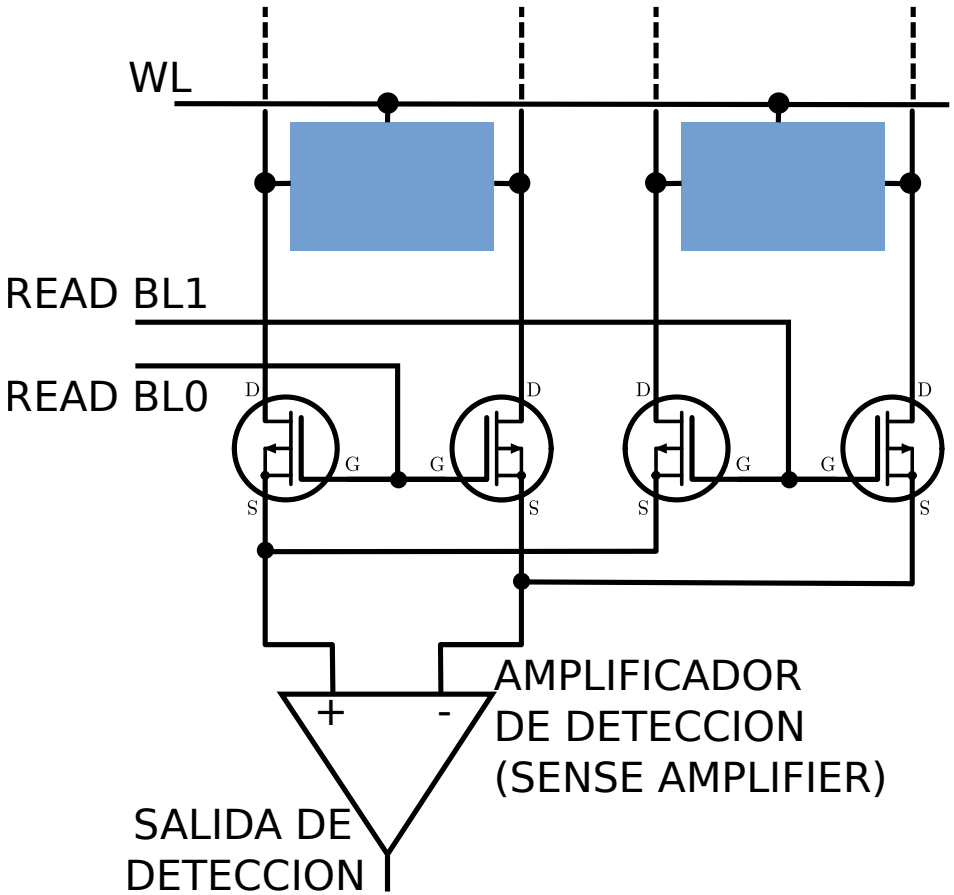


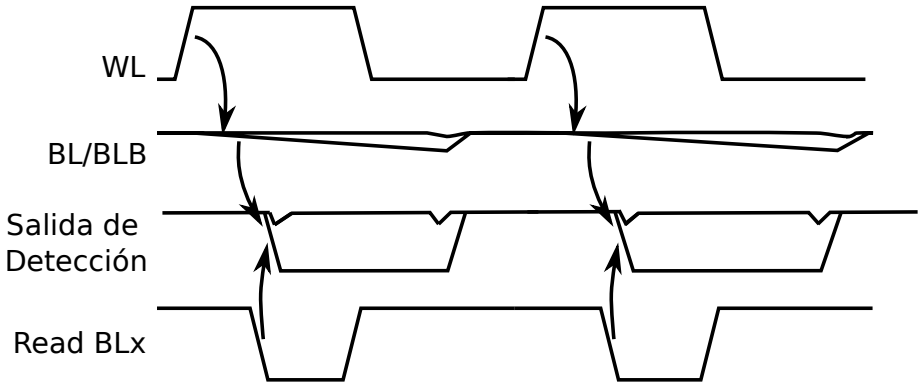
Amplificador de Detección





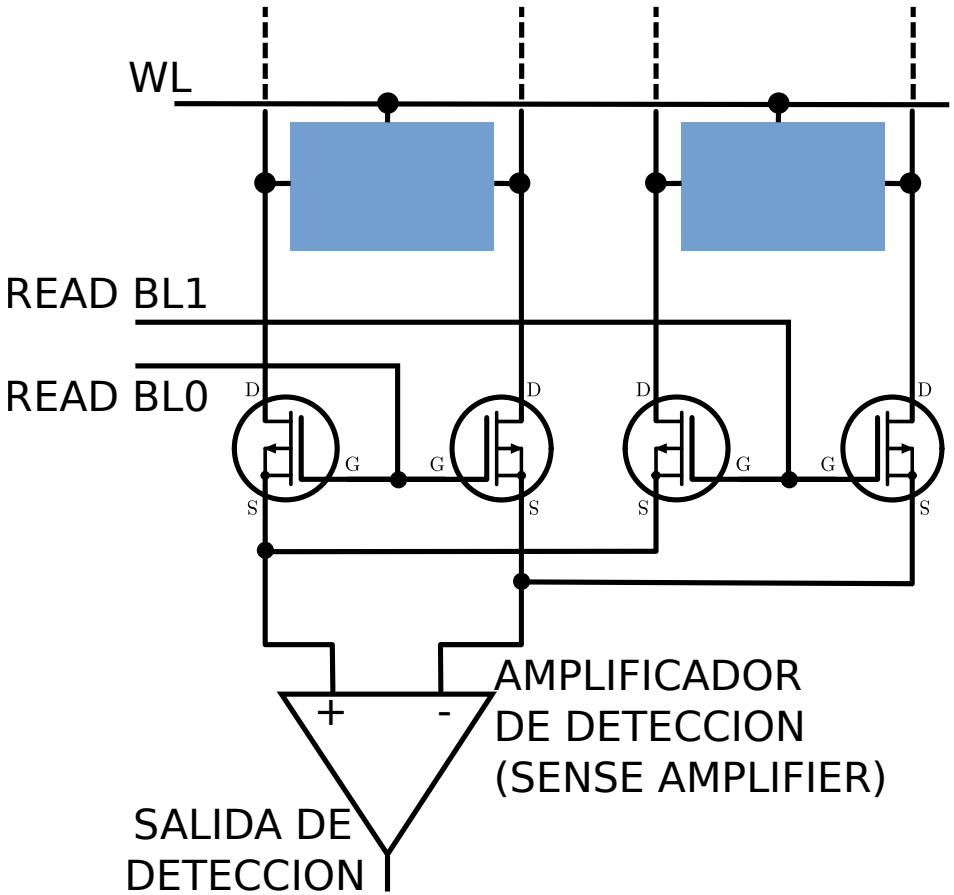
Amplificador de Detección

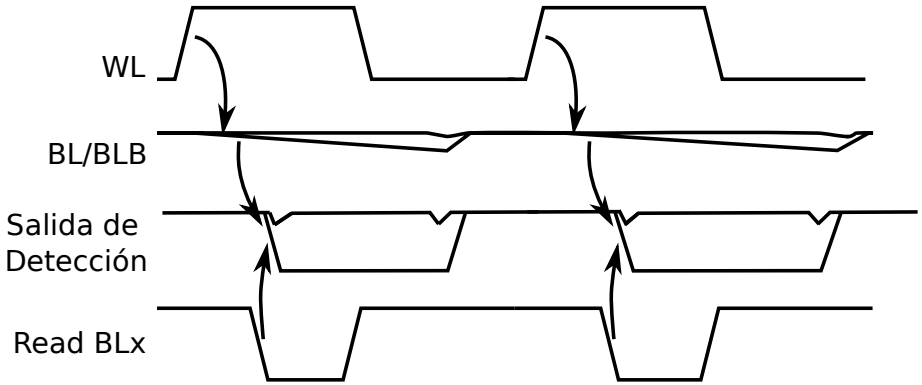




- Al activarse la **WL** empieza a aparecer una tensión diferencial entre **BL** y **BLB**.

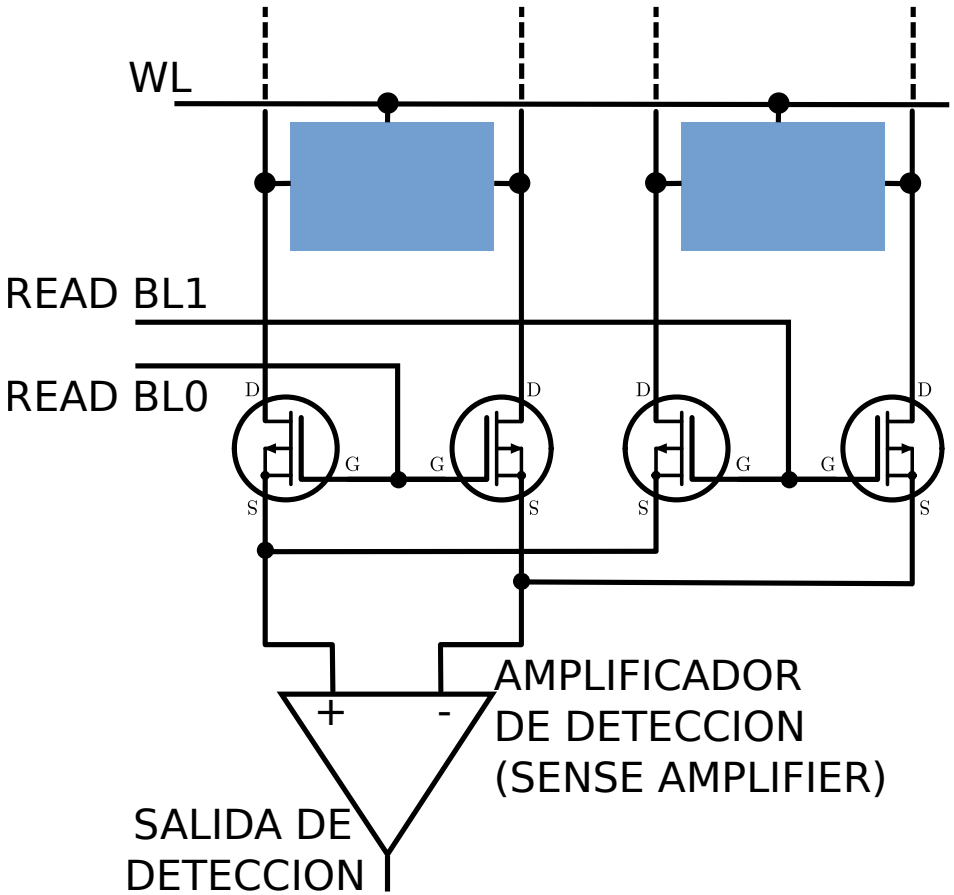
Amplificador de Detección

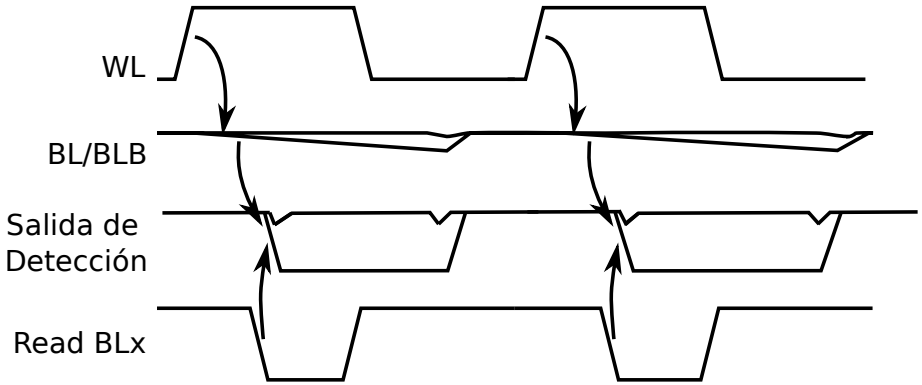




- Al activarse la **WL** empieza a aparecer una tensión diferencial entre **BL** y **BLB**.
- Luego del tiempo necesario para que se establezca una mínima tensión diferencial, se habilita la compuerta del PMOS para atacar las entradas del Amplificador de Detección.

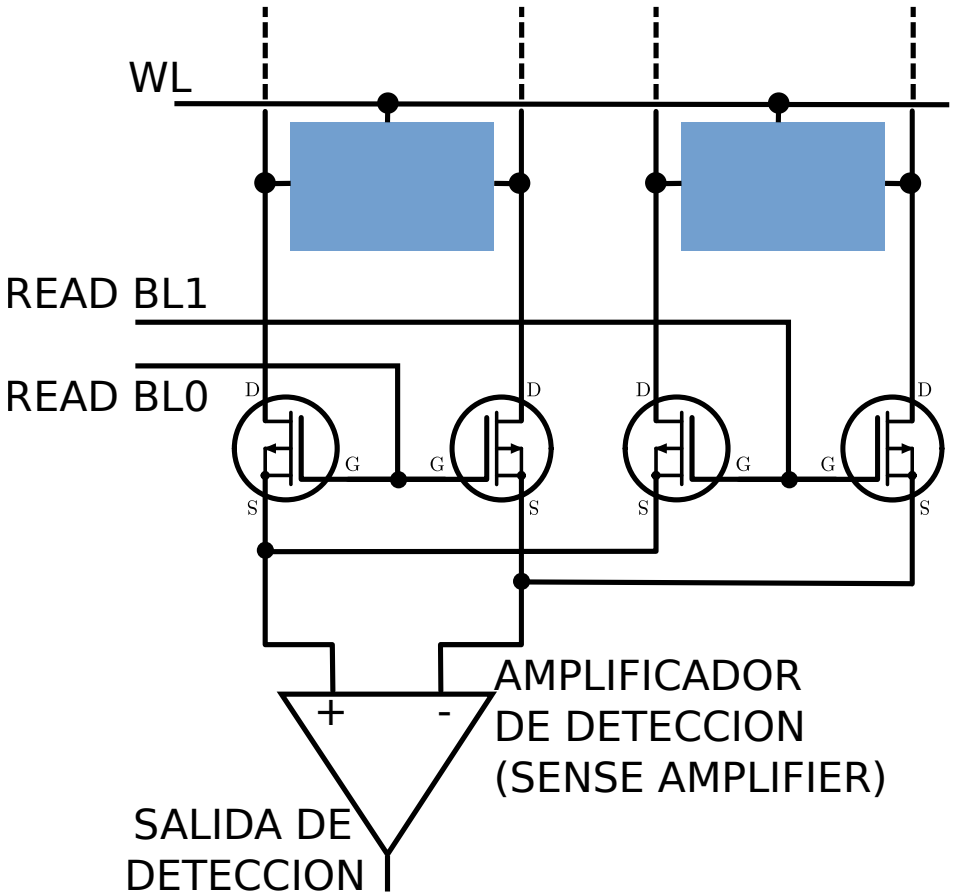
Amplificador de Detección

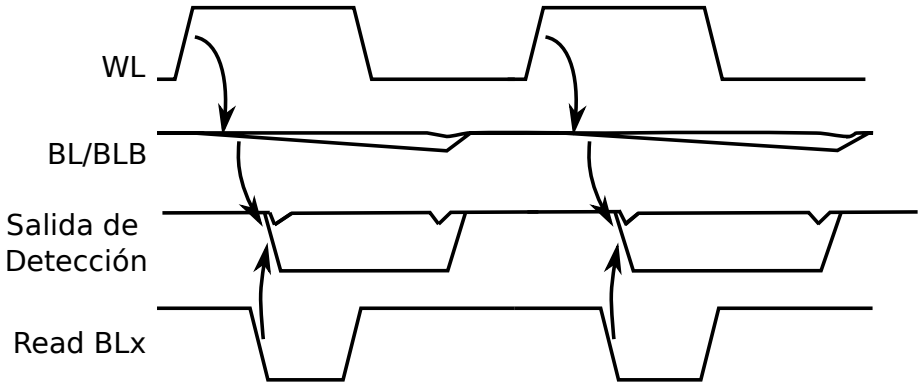




- El Amplificador de Detección convierte una diferencia débil de potencial en sus entradas, en un valor V_{DD} o 0v a su salida, dependiendo del sentido de la tensión diferencial.

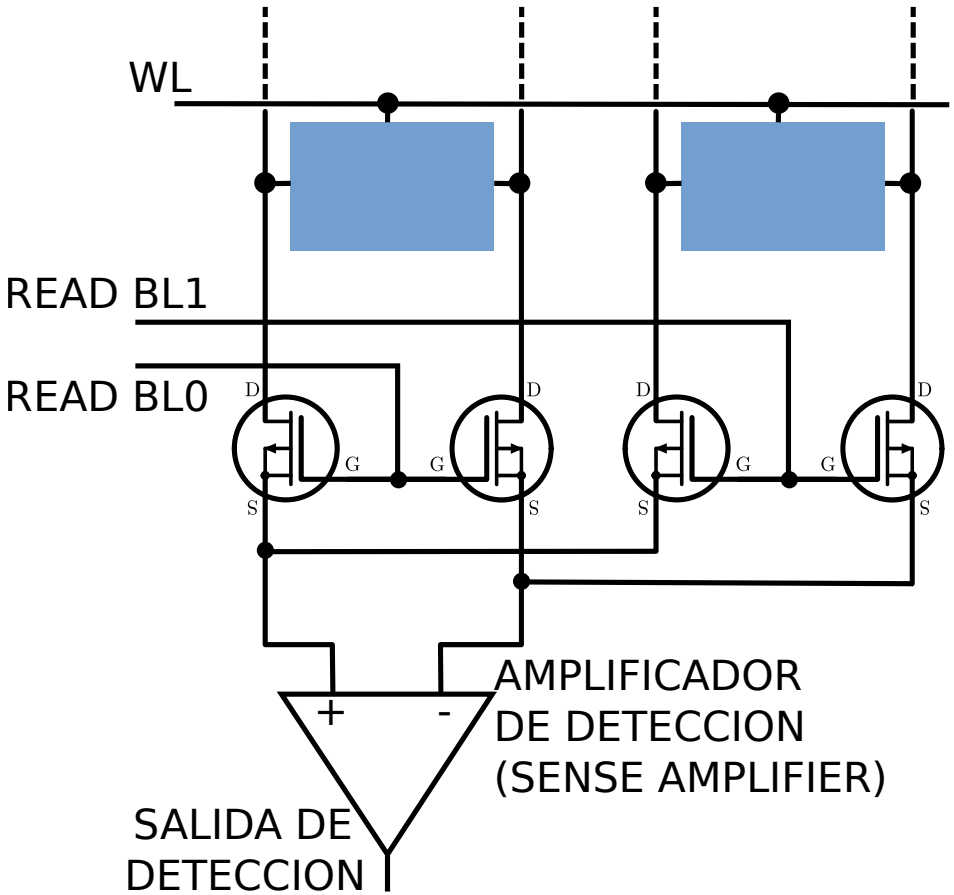
Amplificador de Detección

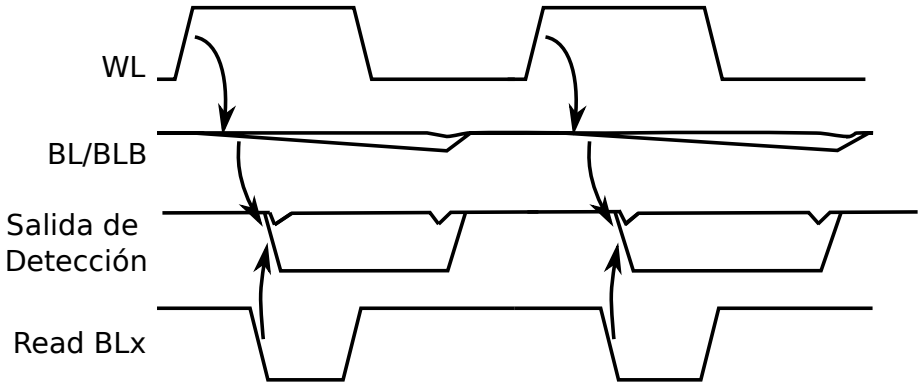




- Notar que es innecesario aumentar mas la señal diferencial. Se disiparía mas potencia en la recarga de las bitlines.

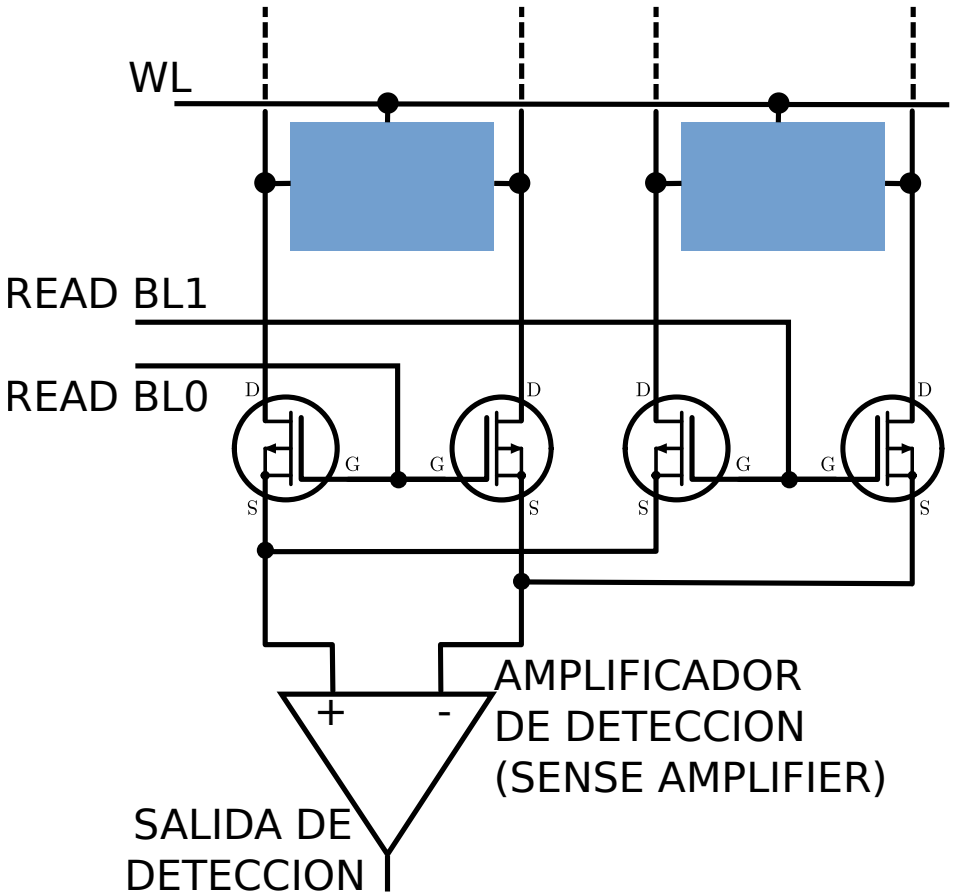
Amplificador de Detección

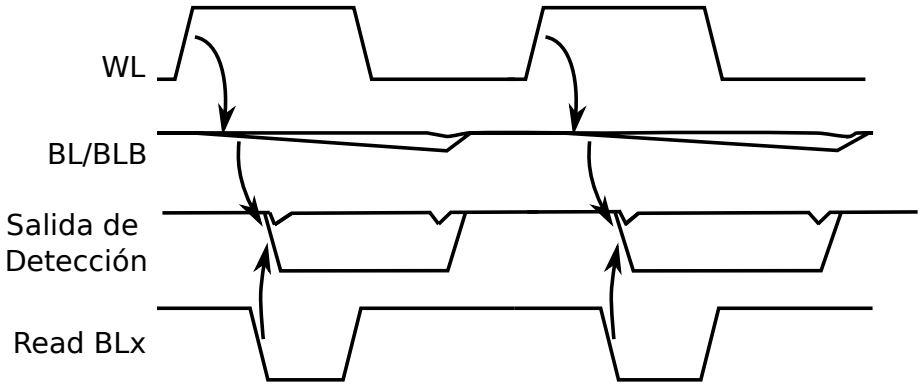




- Notar que es innecesario aumentar mas la señal diferencial. Se disiparía mas potencia en la recarga de las bitlines.
- El manejo de Pulsed Word Lines (**PWL**), minimiza la potencia disipada y la velocidad de detección. Se nota en el decremento del final de la tensión diferencial en la entrada.

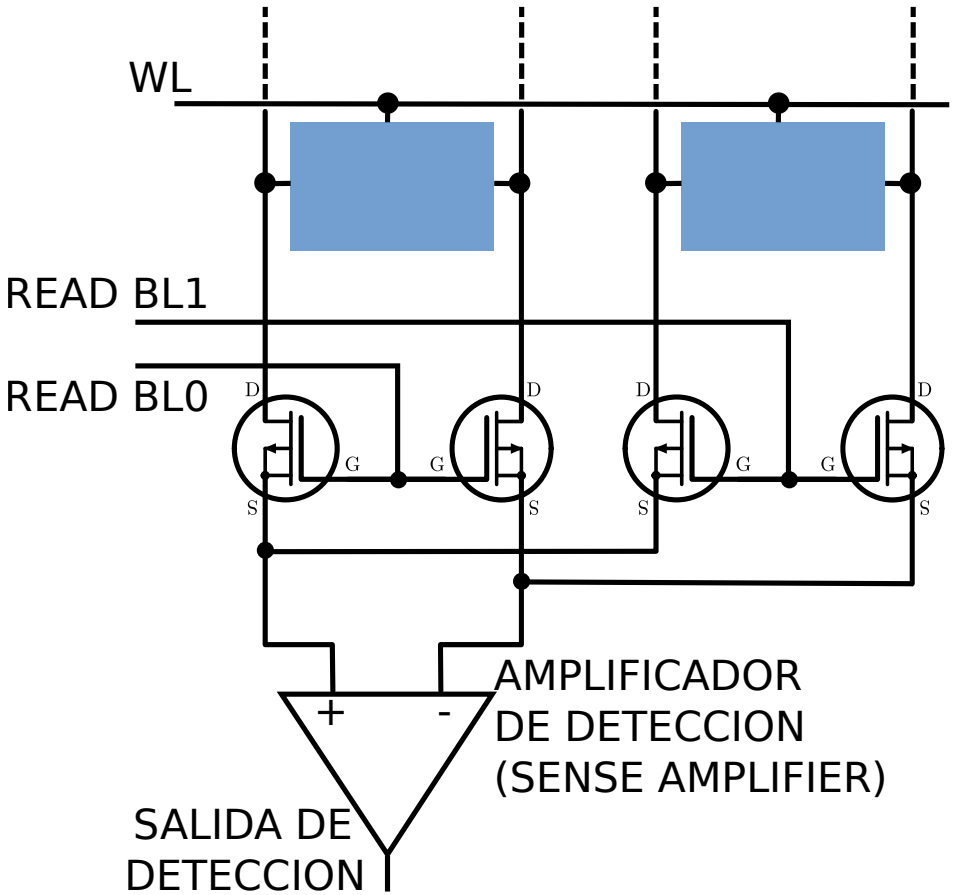
Amplificador de Detección

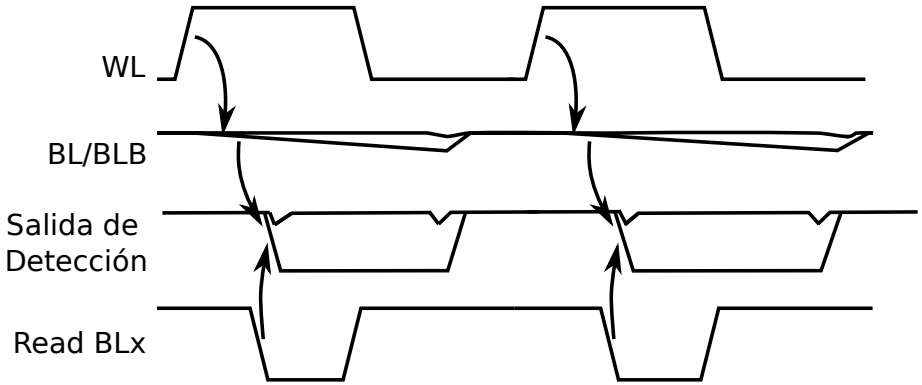




- El Amplificador de Detección acelera el acceso al contenido de la celda de memoria. A cambio de mas disipación de potencia.

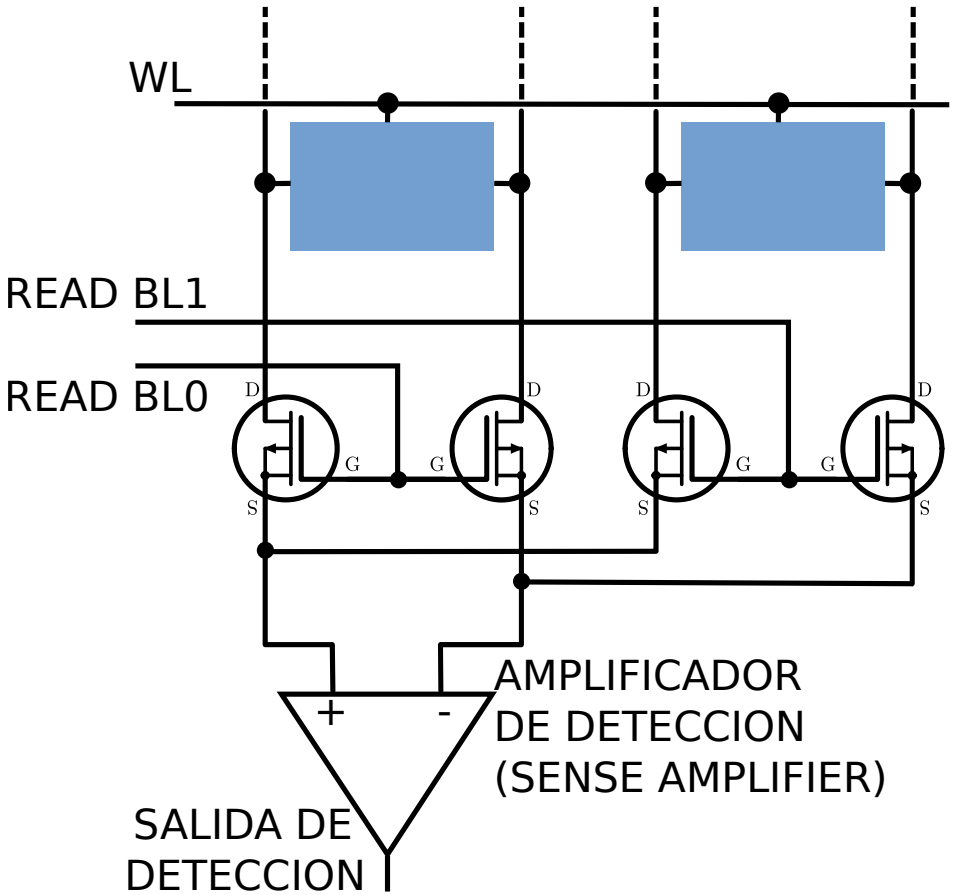
Amplificador de Detección

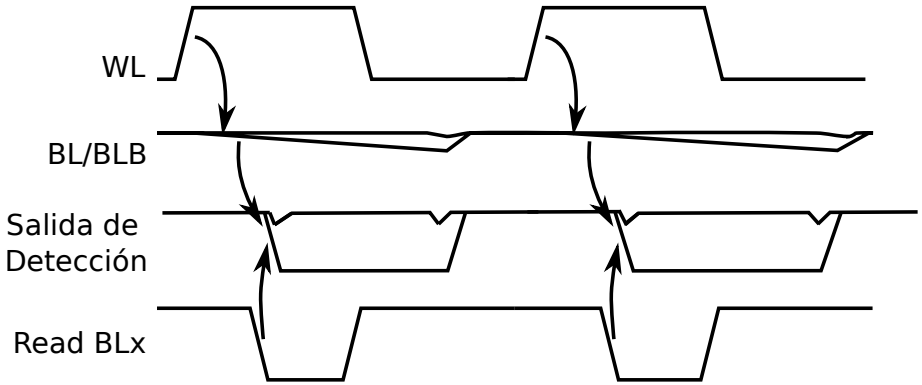




- El Amplificador de Detección acelera el acceso al contenido de la celda de memoria. A cambio de mas disipación de potencia.
- En un bit de DRAM, la corriente es cedida por la capacidad de la celda y se descarga con la lectura. Es indispensable un Amplificador de Detección ya que la corriente es efímera.

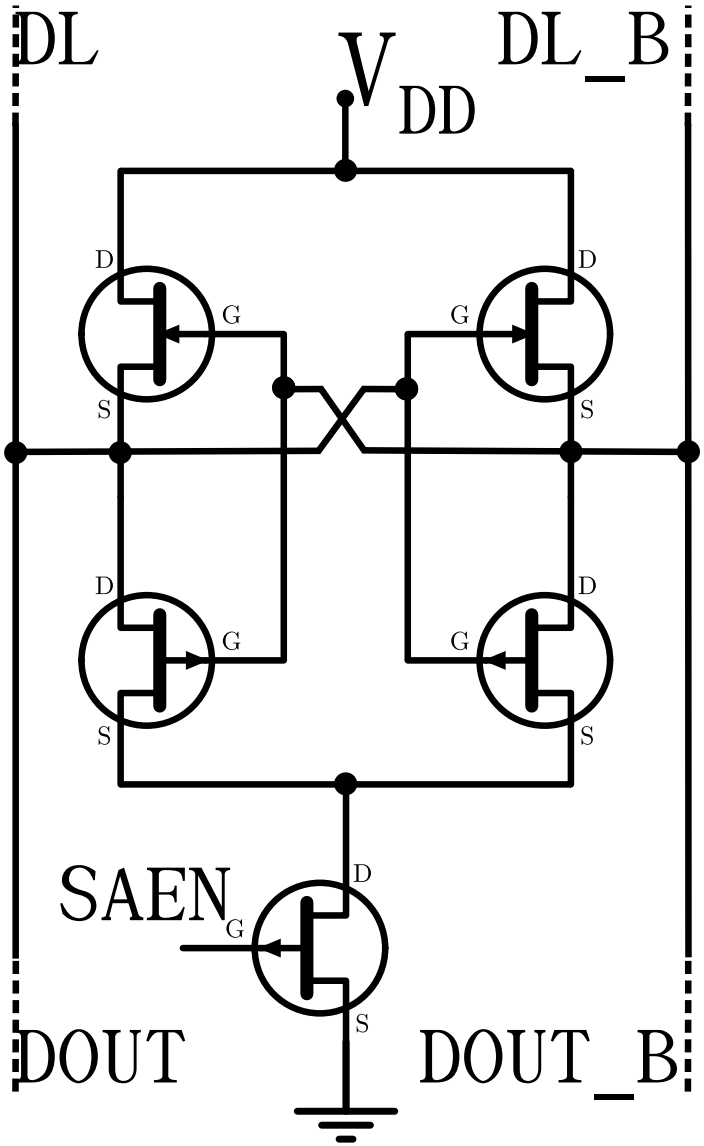
Amplificador de Detección



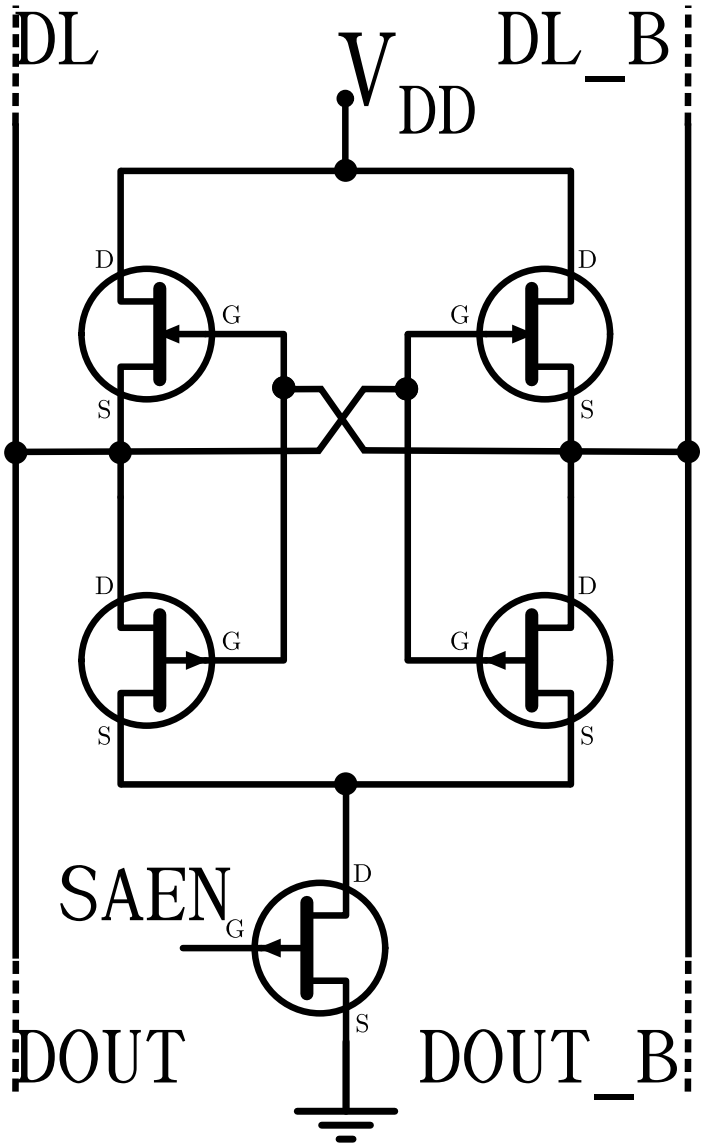


- En un bit de SRAM en cambio puede evitarse el Amplificador de Detección, cuando las capacidades de la bitline son bajas (scaling), en cuyo caso se descargan per sé a gran velocidad sin afectar los requisitos de tiempo de acceso a la celda, y ahorrando disipación de potencia.

Amplificador de Detección Latch Simple

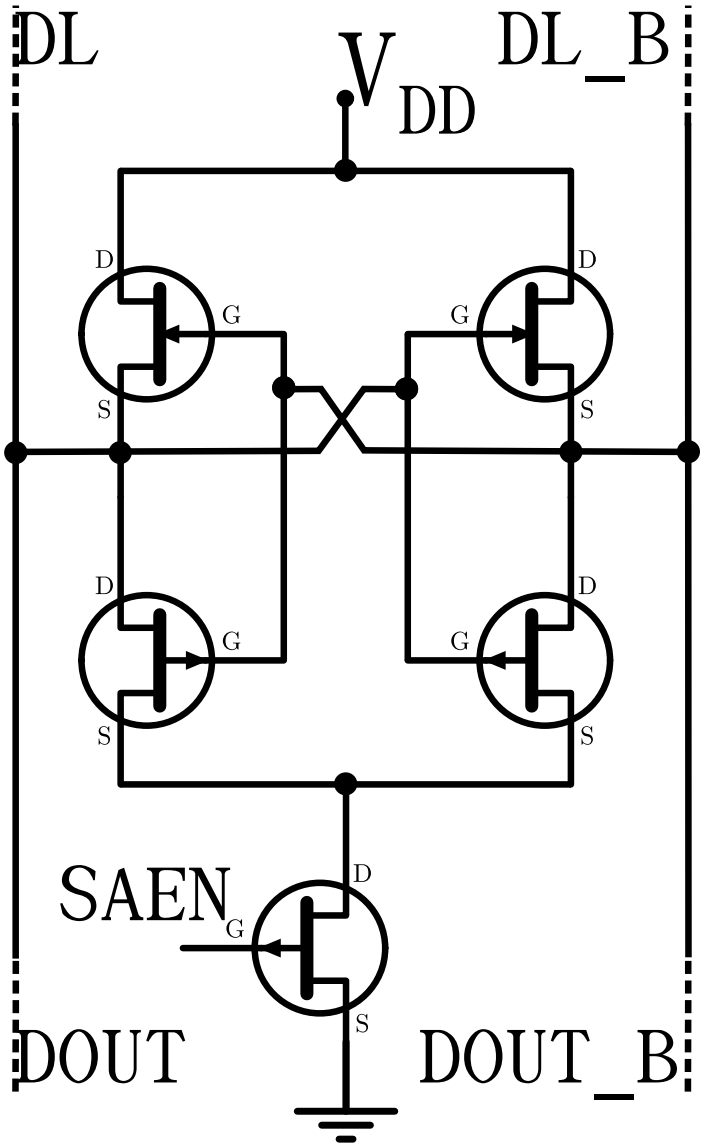


Amplificador de Detección Latch Simple



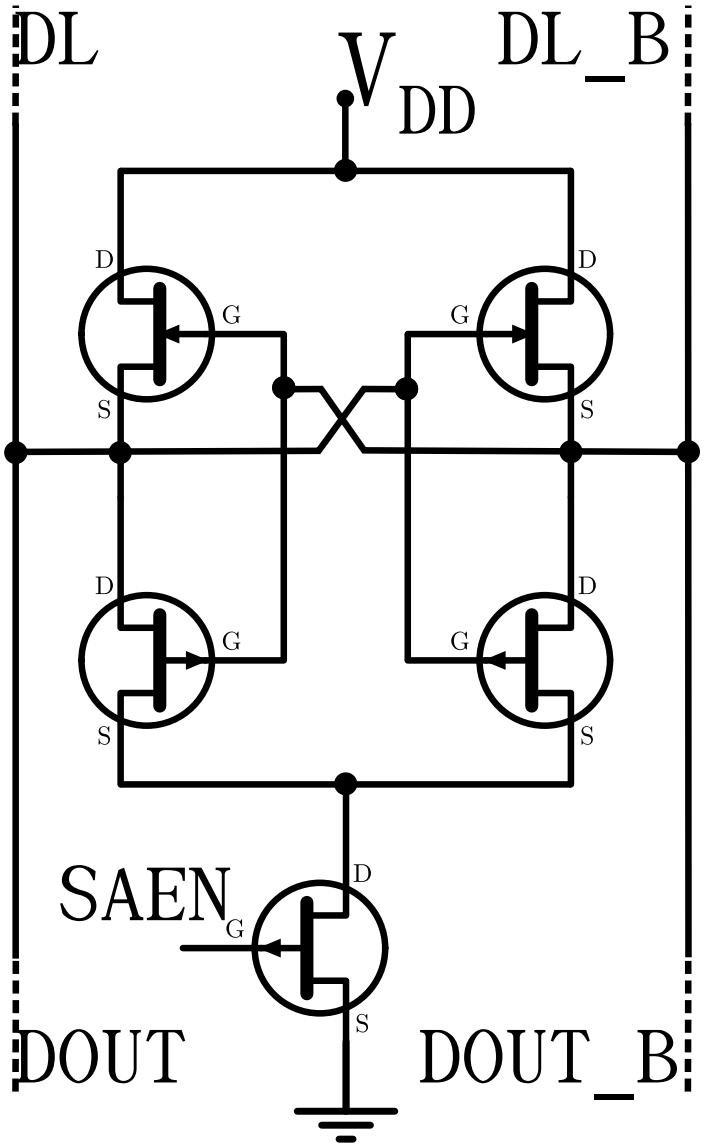
- Cuando está habilitado (SAEN=1) es un par de inversores cross acoplados.

Amplificador de Detección Latch Simple



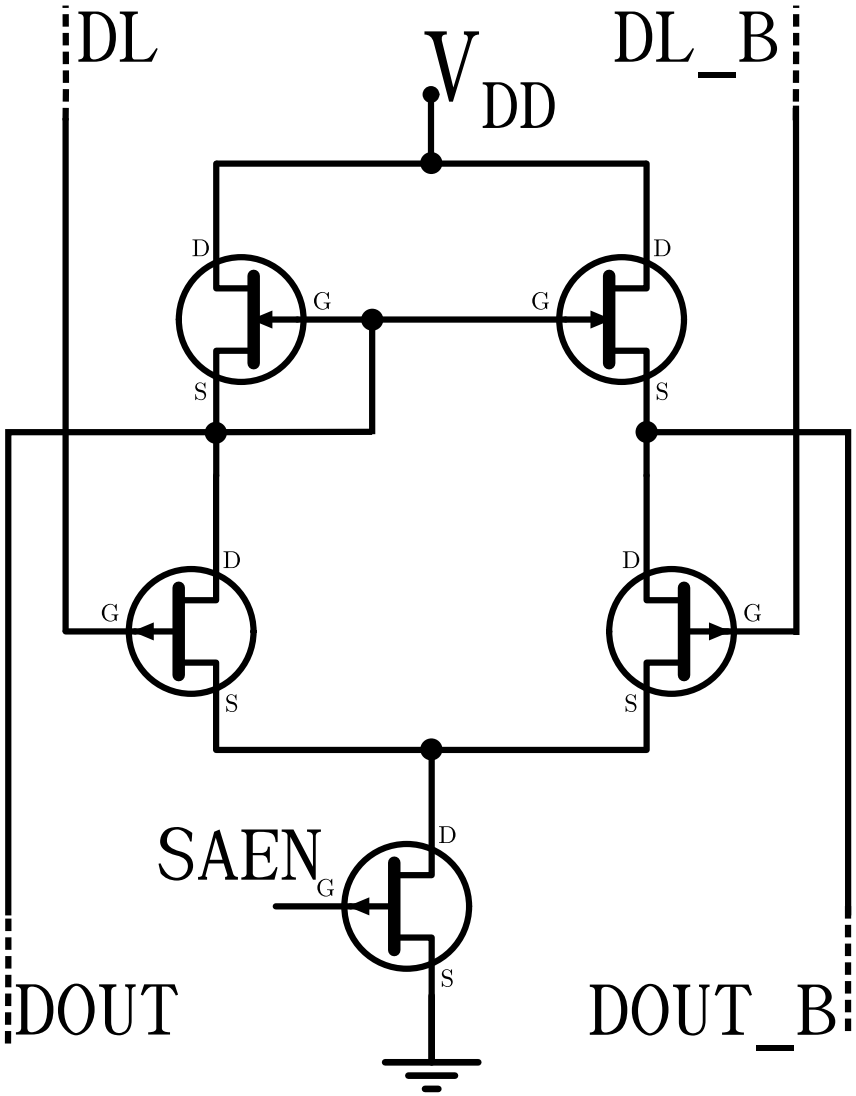
- Cuando está habilitado (SAEN=1) es un par de inversores cross acoplados.
- Es sencillo. Permite lograr una relación de compromiso aceptable entre área de *Si* requerida, disipación moderada de potencia y suficiente velocidad de acceso. Apto para SRAMs grandes que requieren muchos Amplificadores de Detección.

Amplificador de Detección Latch Simple

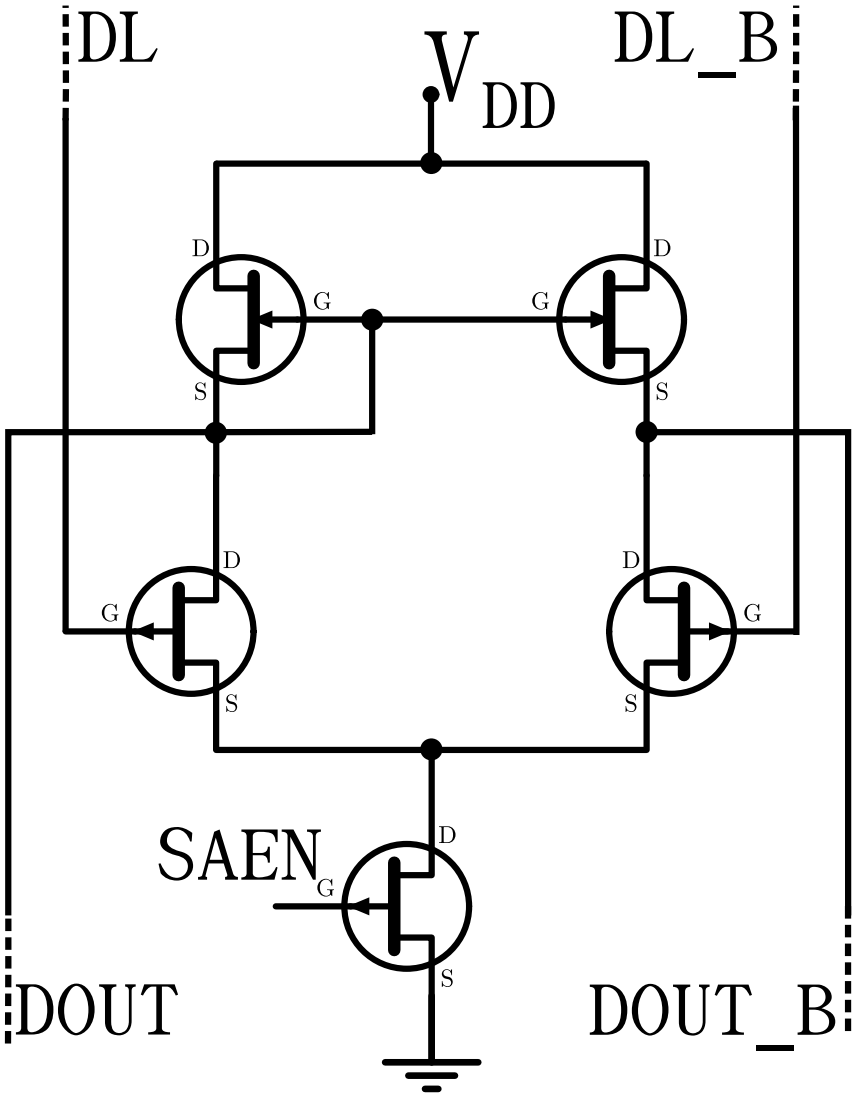


- Cuando está habilitado (SAEN=1) es un par de inversores cross acoplados.
- Es sencillo. Permite lograr una relación de compromiso aceptable entre área de *Si* requerida, disipación moderada de potencia y suficiente velocidad de acceso. Apto para SRAMs grandes que requieren muchos Amplificadores de Detección.
- Desventaja: Requiere habilitarlo solo cuando se haya generado una mínima tensión diferencial. De otro modo puede invertir el estado y sobrecargar el bitline diferencial.

Amplificador de Detección Espejo de Corriente

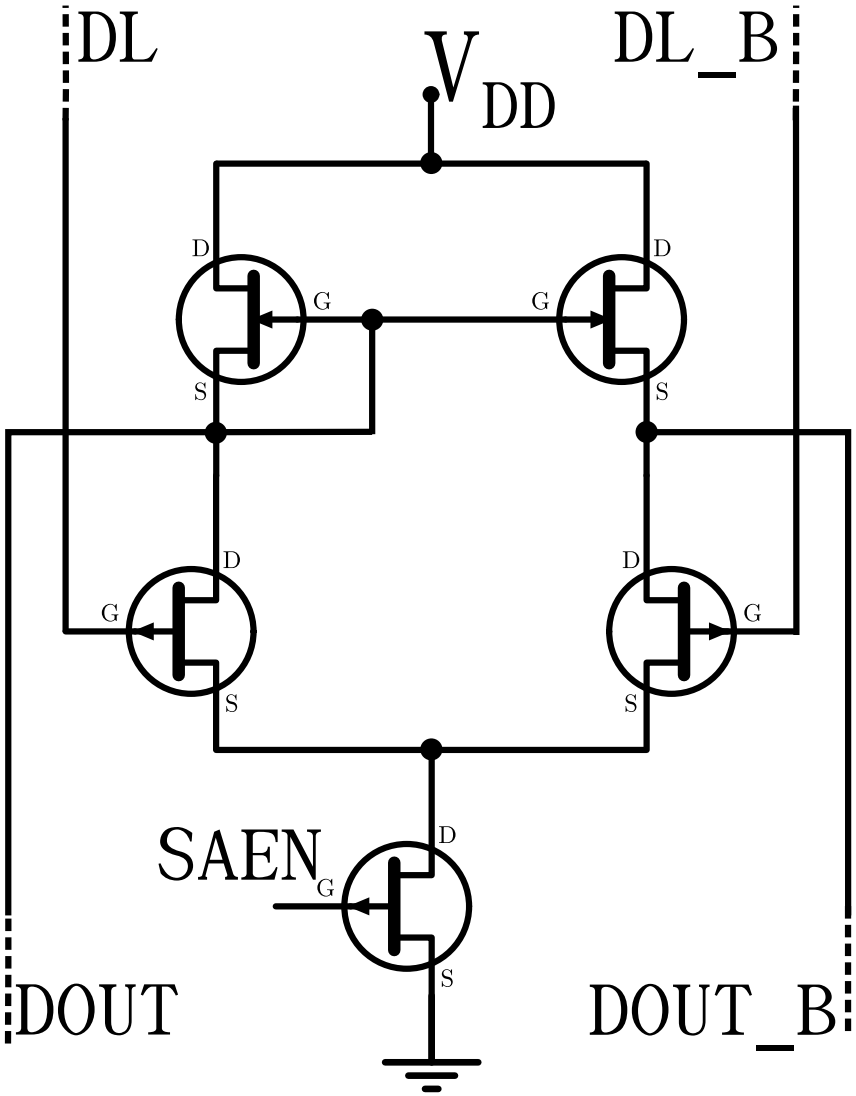


Amplificador de Detección Espejo de Corriente



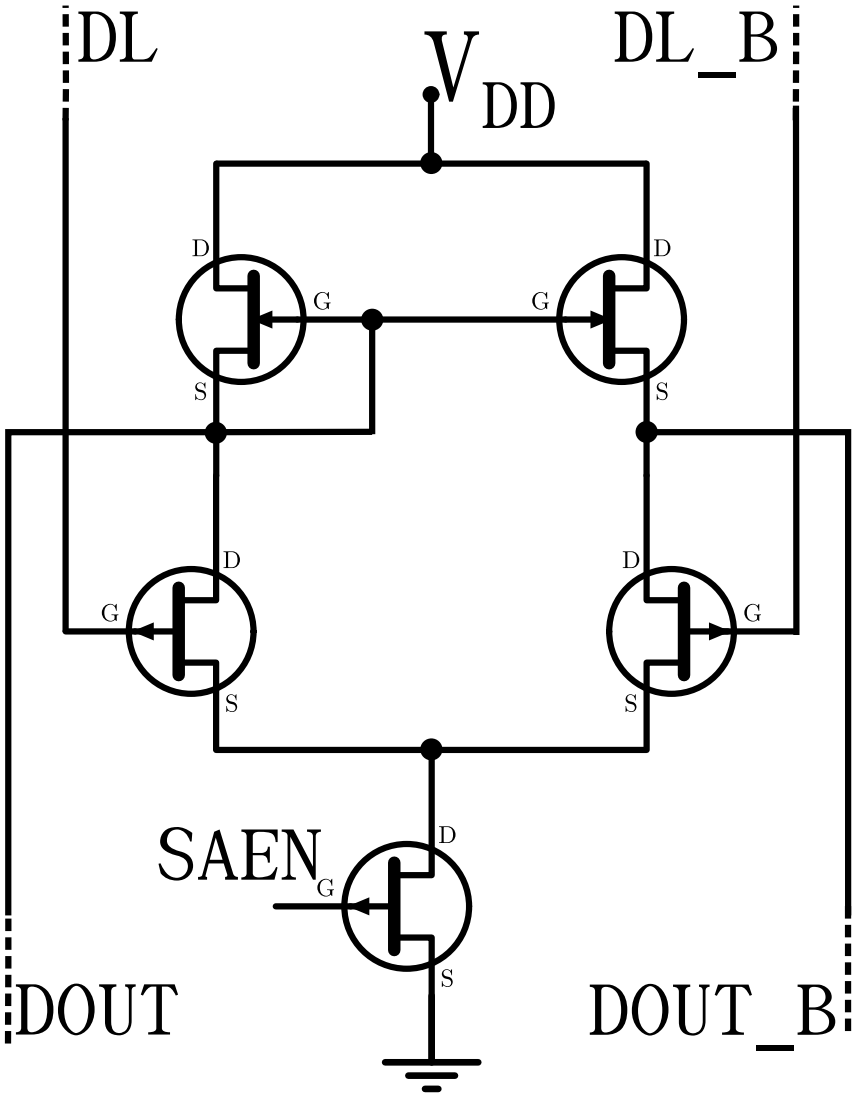
- En general los amplificadores espejo de corriente ofrecen una salida muy estable, velocidad de detección y una alta ganancia de tensión.

Amplificador de Detección Espejo de Corriente



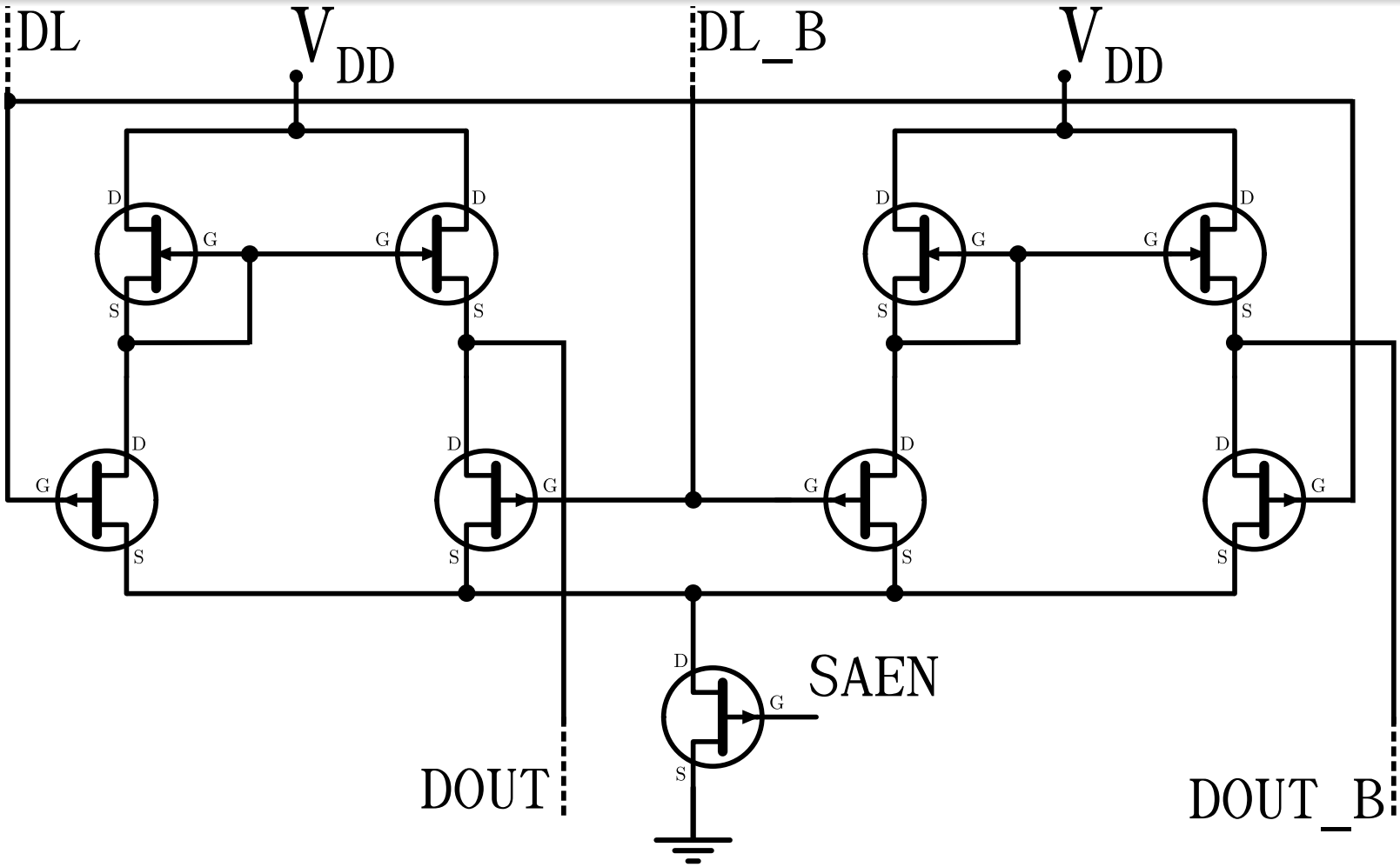
- En general los amplificadores espejo de corriente ofrecen una salida muy estable, velocidad de detección y una alta ganancia de tensión.
- El costo que apareja es una disipación de potencia mas alta ya que hay una vía de corriente estática entre V_{DD} y tierra si el amplificador está habilitado.

Amplificador de Detección Espejo de Corriente

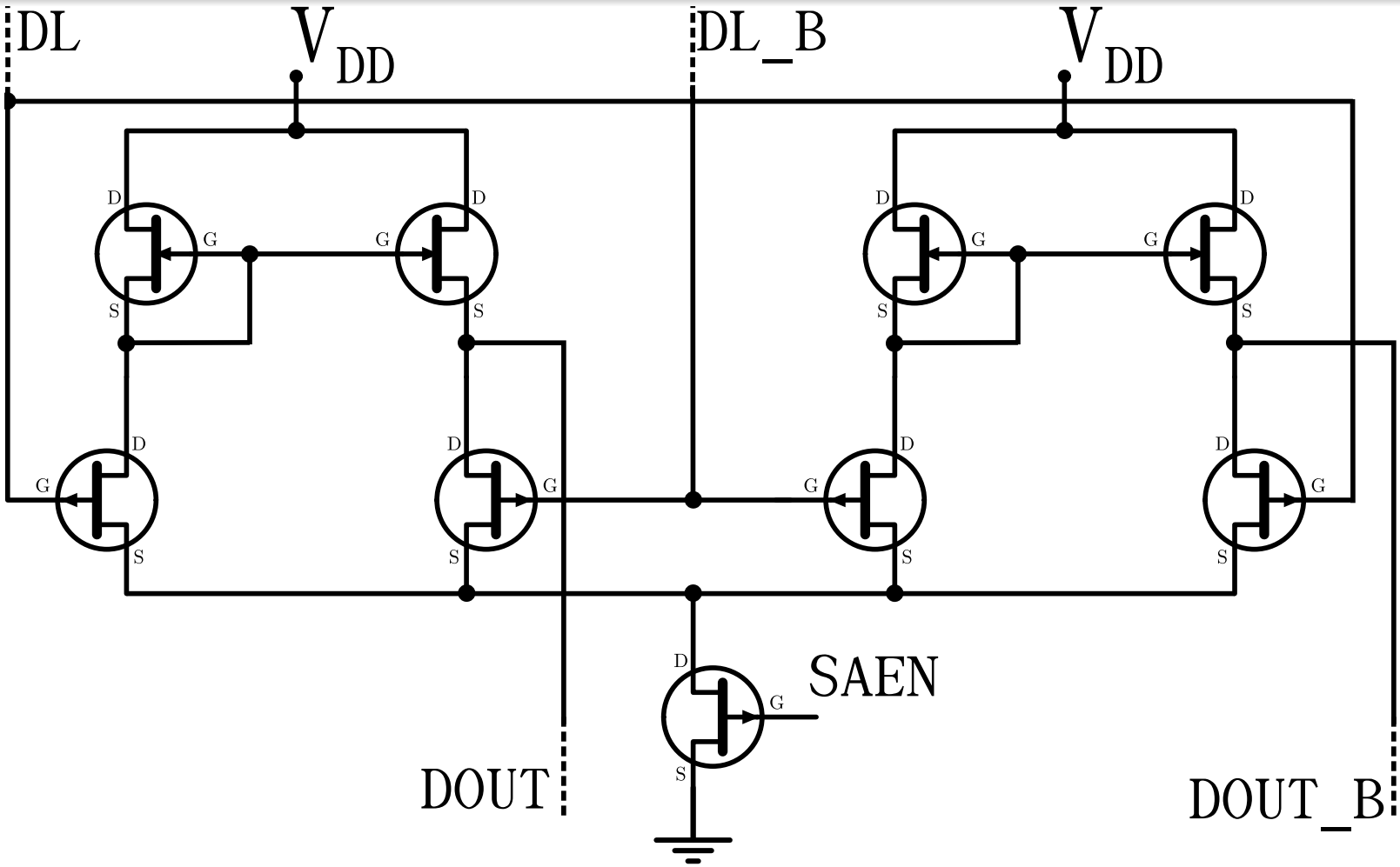


- En general los amplificadores espejo de corriente ofrecen una salida muy estable, velocidad de detección y una alta ganancia de tensión.
- El costo que apareja es una disipación de potencia mas alta ya que hay una vía de corriente estática entre V_{DD} y tierra si el amplificador está habilitado.
- Esta corriente aumenta cuanta mas velocidad de detección se busque proporcionarle al amplificador.

Amplificador de Detección PCMA

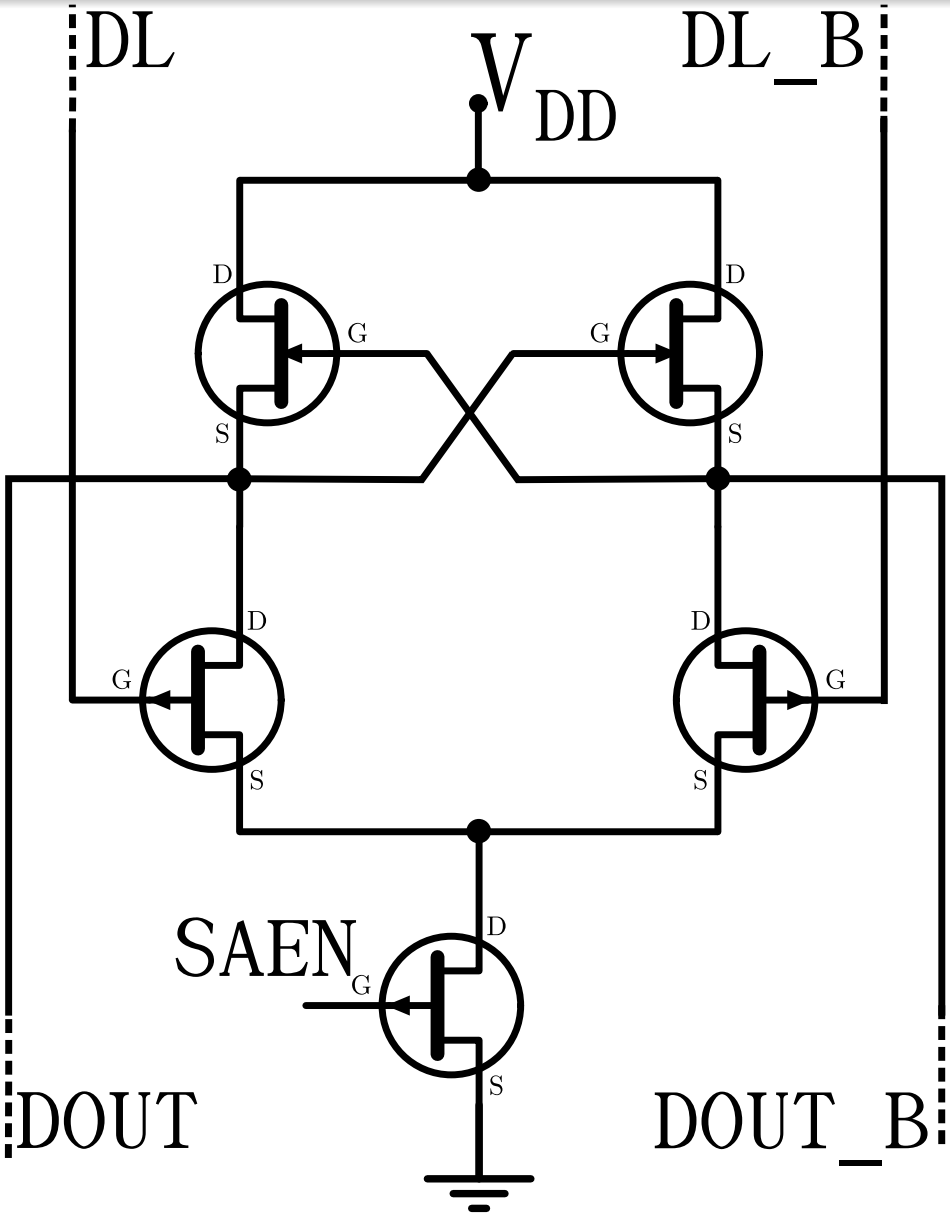


Amplificador de Detección PCMA

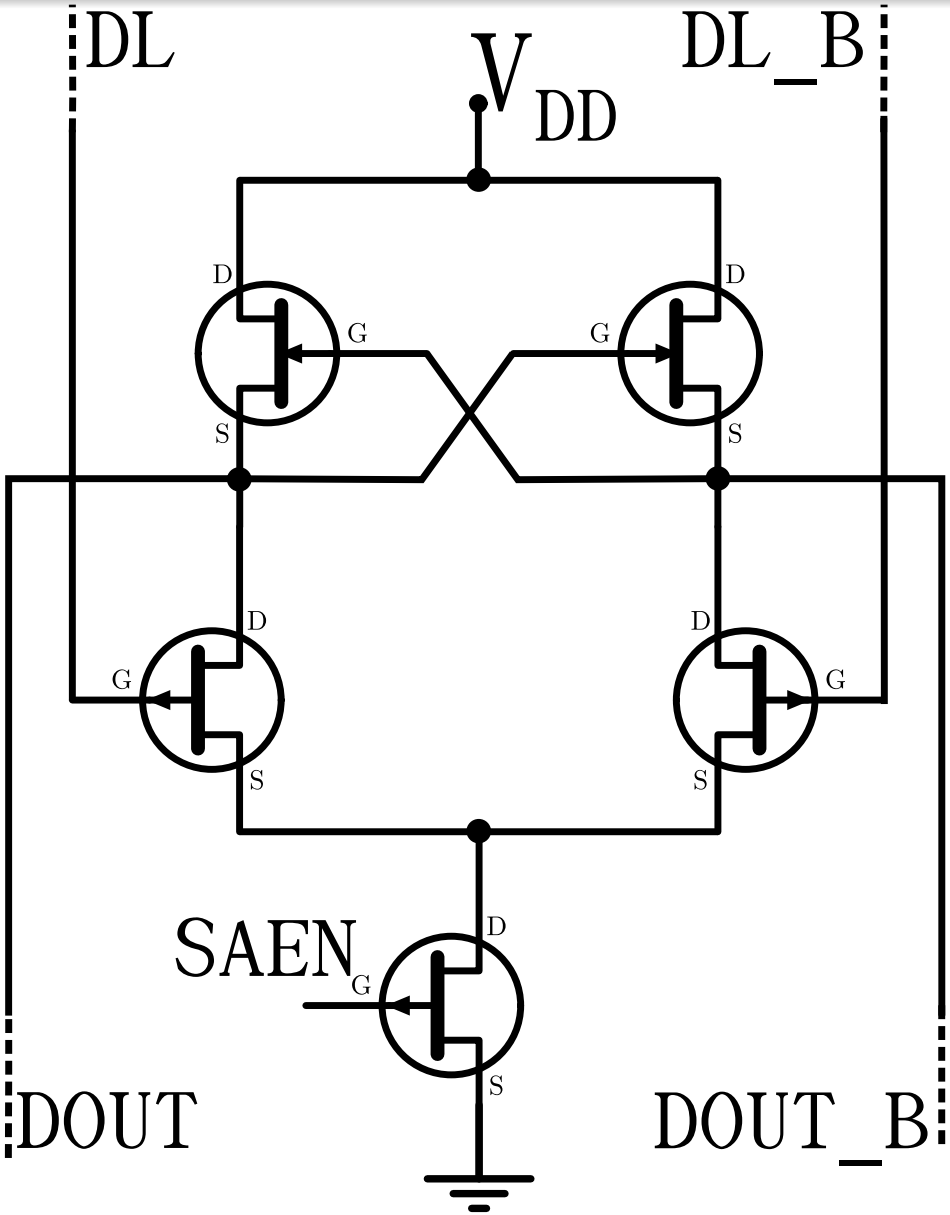


- **P**air **C**urrent **M**irror **A**mplifier. Es un Amplificador espejo de corriente doble. Es una alternativa para duplicar las ventajas y las desventajas del anterior. No es apto para SRAMs de alta densidad.

Amplificador de Detección PCCA

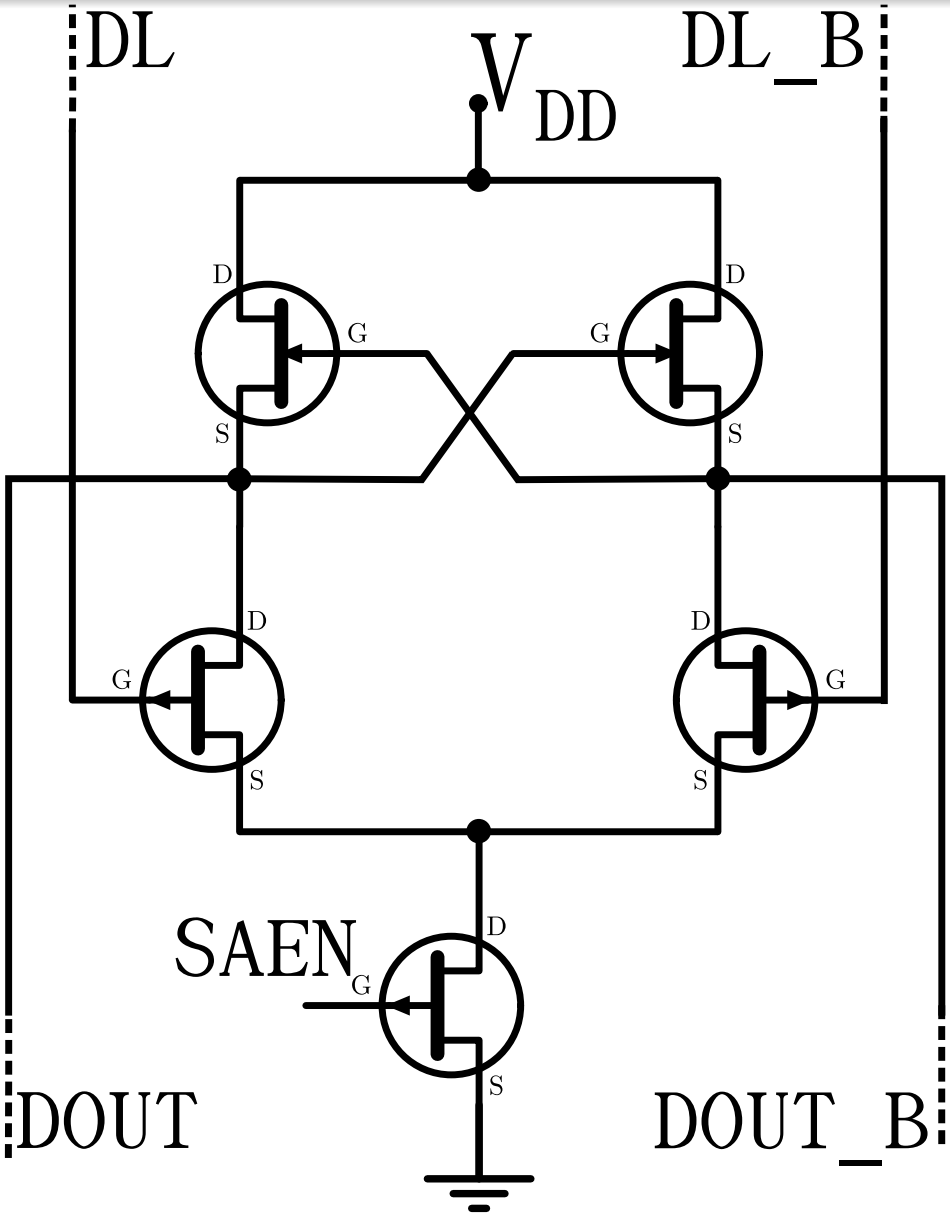


Amplificador de Detección PCCA



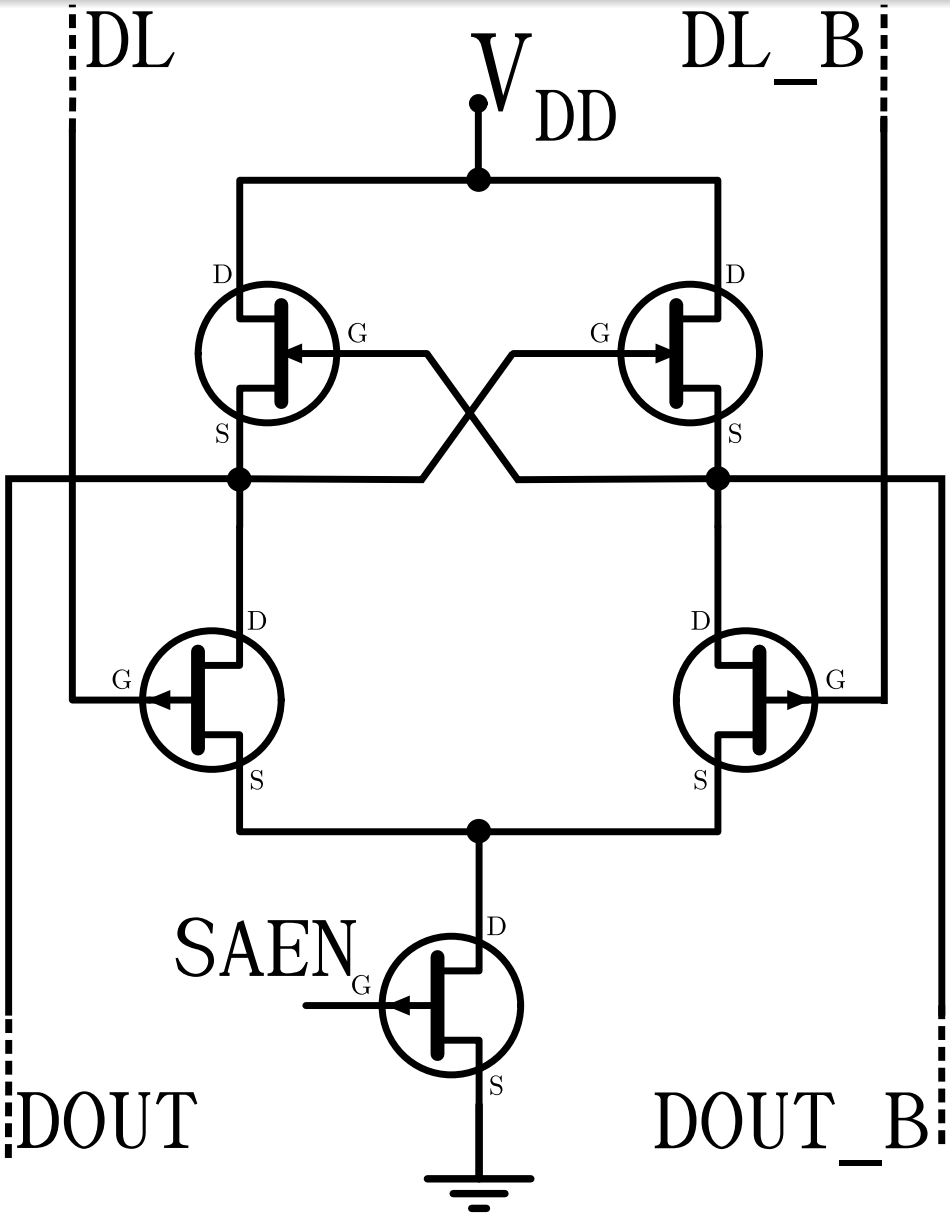
- **P**MOS **C**ross **C**oupled **A**mplifier, provee alta velocidad de detección sin la corriente estática del Espejo de Corriente, (menor disipación de potencia estática).

Amplificador de Detección PCCA



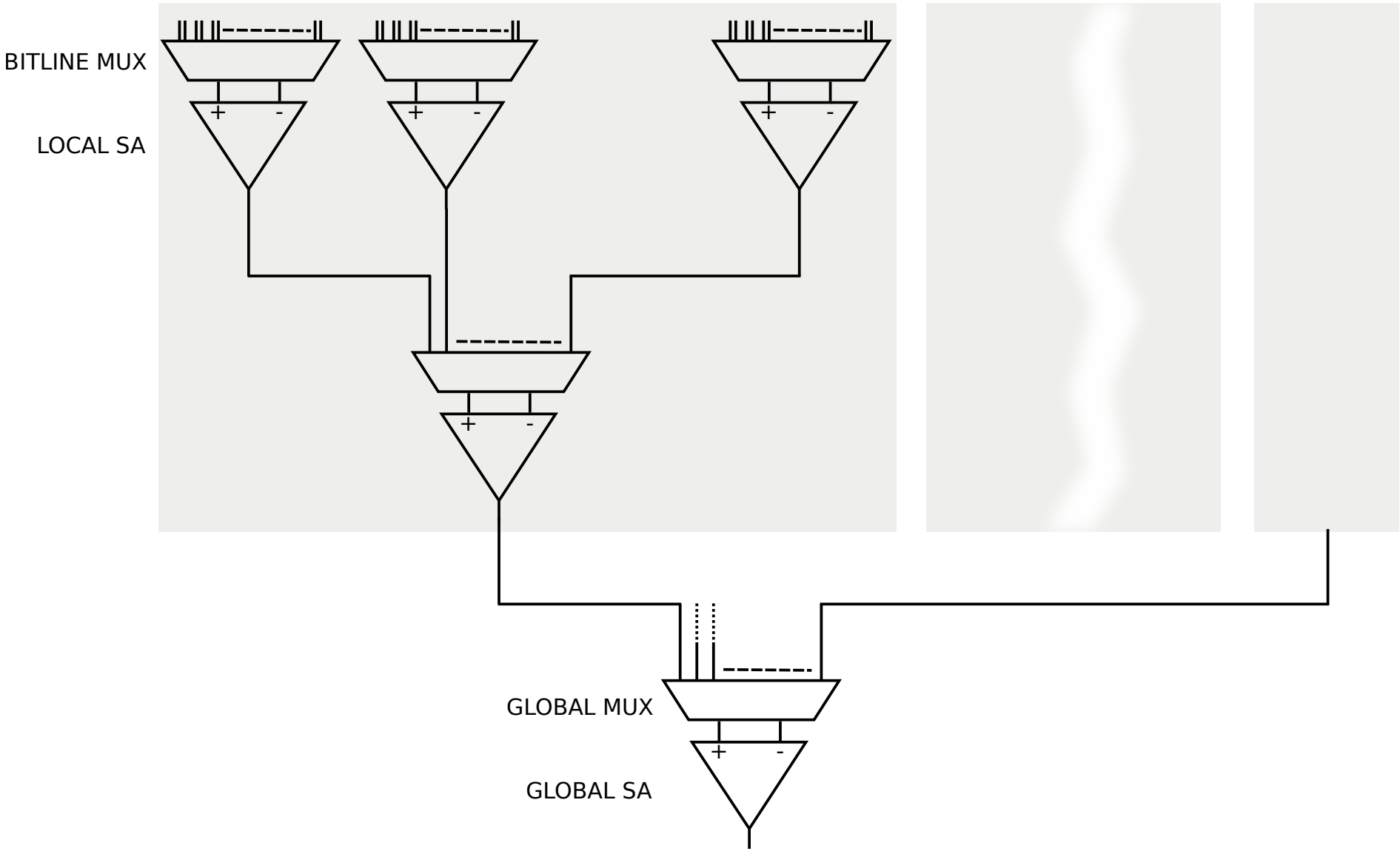
- **P**MOS **C**ross **C**oupled **A**mplifier, provee alta velocidad de detección sin la corriente estática del Espejo de Corriente, (menor disipación de potencia estática).
- Para muy alta velocidad de detección requiere un pre amplificador. Normalmente se coloca un PCMA o un tipo Latch Simple en esta etapa.

Amplificador de Detección PCCA



- **P**MOS **C**ross **C**oupled **A**mplifier, provee alta velocidad de detección sin la corriente estática del Espejo de Corriente, (menor disipación de potencia estática).
- Para muy alta velocidad de detección requiere un pre amplificador. Normalmente se coloca un PCMA o un tipo Latch Simple en esta etapa.
- Se prefiere combinarlo con el Latch Simple en virtud de su menor corriente estática y ser menos sensible a desadaptaciones entre pares de transistores acoplados.

Estructuras jerárquicas de Detección



Estructuras jerárquicas de Detección

- Las estructuras jerárquicas tienen aplicación en memoria de capacidades altas

Estructuras jerárquicas de Detección

- Las estructuras jerárquicas tienen aplicación en memoria de capacidades altas
- Cada nivel de la jerarquía puede tener amplificadores de detección de las topologías diferentes vistas de acuerdo a la necesidad de cada etapa.

Estructuras jerárquicas de Detección

- Las estructuras jerárquicas tienen aplicación en memoria de capacidades altas
- Cada nivel de la jerarquía puede tener amplificadores de detección de las topologías diferentes vistas de acuerdo a la necesidad de cada etapa.
- Por lo general las caches internas a pesar de ser muy densas en los procesadores de alta gama, utilizan una estructura plana porque a diferencia de las SRAM discretas tienen buses de datos muy anchos.

Amplificadores de Escritura

Amplificadores de Escritura

- Su función es manejar las tensiones para invertir el estado de una celda cuando es necesario.

Amplificadores de Escritura

- Su función es manejar las tensiones para invertir el estado de una celda cuando es necesario.
- Mucho mas sencillos que los amplificadores de detección, hacen que la operación de escritura de una DRAM no sea crítica.

Amplificadores de Escritura

- Su función es manejar las tensiones para invertir el estado de una celda cuando es necesario.
- Mucho mas sencillos que los amplificadores de detección, hacen que la operación de escritura de una DRAM no sea crítica.
- Retomando el circuito de una celda de bit SRAM, básicamente solo hay que aplicar una tensión diferencial entre **BL** y **BLB** .

Amplificadores de Escritura

- Su función es manejar las tensiones para invertir el estado de una celda cuando es necesario.
- Mucho mas sencillos que los amplificadores de detección, hacen que la operación de escritura de una DRAM no sea crítica.
- Retomando el circuito de una celda de bit SRAM, básicamente solo hay que aplicar una tensión diferencial entre **`BL`** y **`BLB`** .
- Se asume que ambas se encuentran precargadas a V_{DD}, para forzar que una de ellas se derive a tierra.

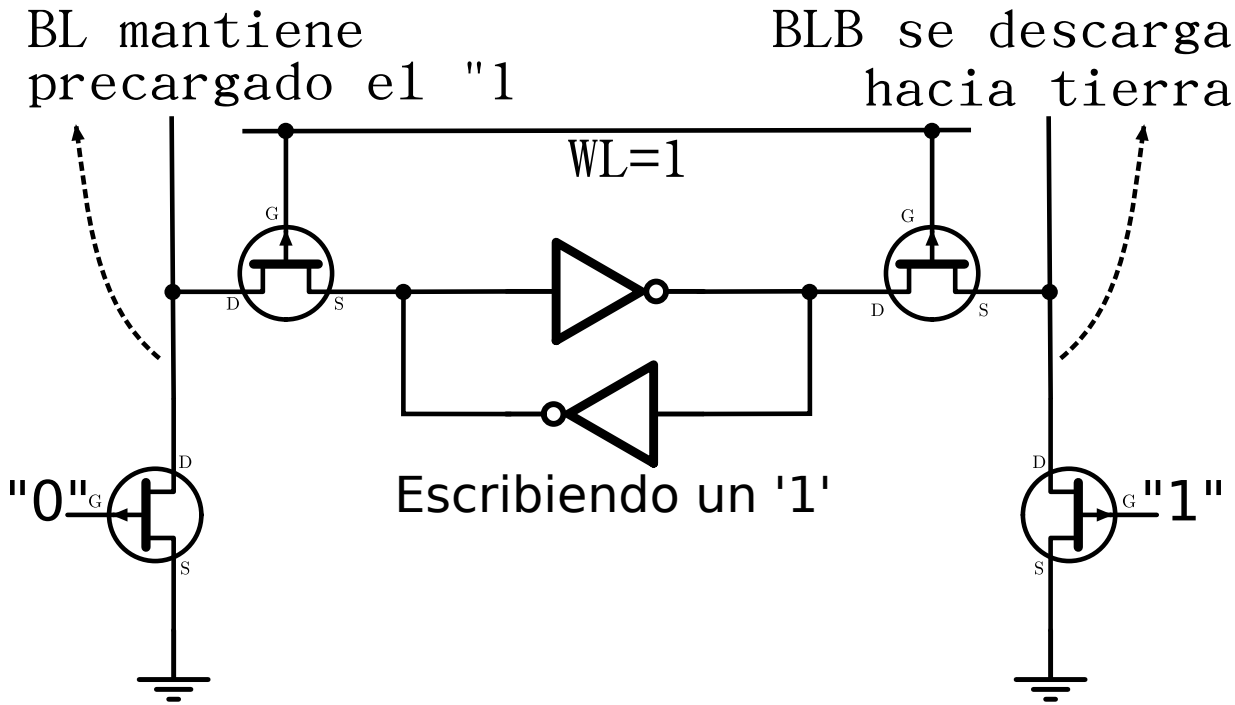
Amplificadores de Escritura

- Su función es manejar las tensiones para invertir el estado de una celda cuando es necesario.
- Mucho mas sencillos que los amplificadores de detección, hacen que la operación de escritura de una DRAM no sea crítica.
- Retomando el circuito de una celda de bit SRAM, básicamente solo hay que aplicar una tensión diferencial entre **BL** y **BLB**.
- Se asume que ambas se encuentran precargadas a V_{DD}, para forzar que una de ellas se derive a tierra.
- Por supuesto debe estar activa **WL**.

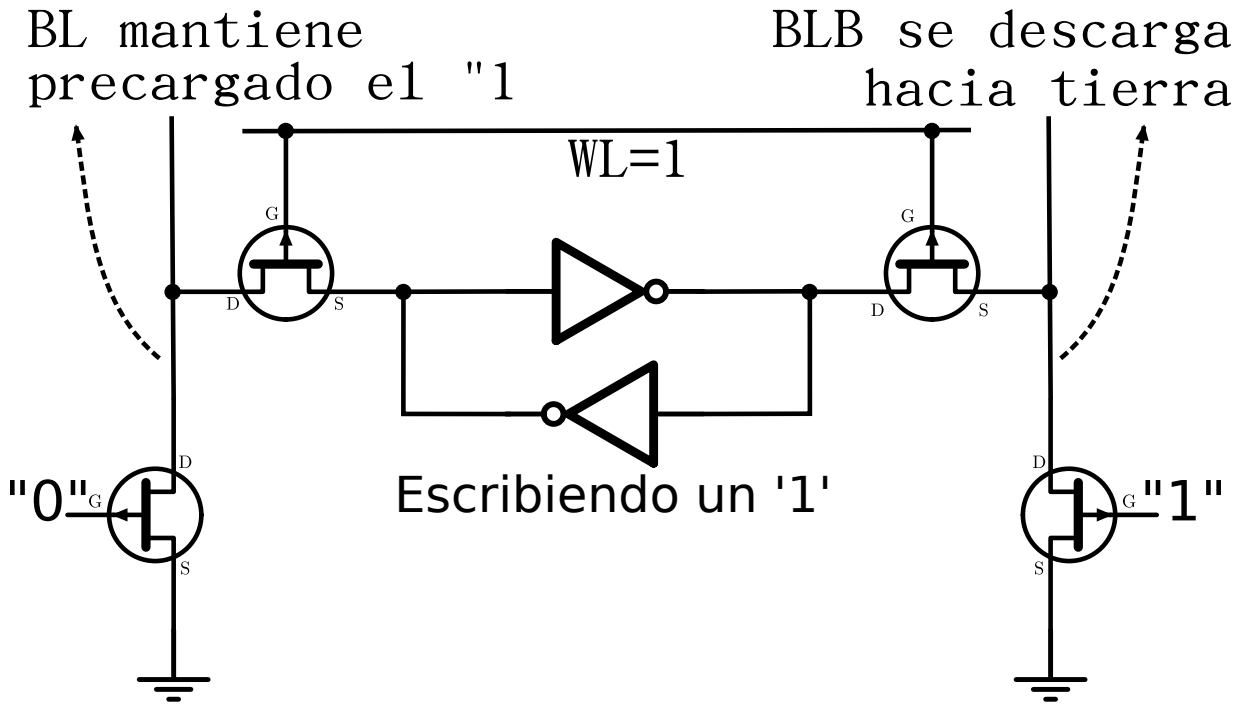
Amplificadores de Escritura

- Su función es manejar las tensiones para invertir el estado de una celda cuando es necesario.
- Mucho mas sencillos que los amplificadores de detección, hacen que la operación de escritura de una DRAM no sea crítica.
- Retomando el circuito de una celda de bit SRAM, básicamente solo hay que aplicar una tensión diferencial entre **`BL`** y **`BLB`**.
- Se asume que ambas se encuentran precargadas a V_{DD}, para forzar que una de ellas se derive a tierra.
- Por supuesto debe estar activa **`WL`**.
- A continuación un ejemplo de escritura de un **`'1'`**.

Amplificadores de Escritura

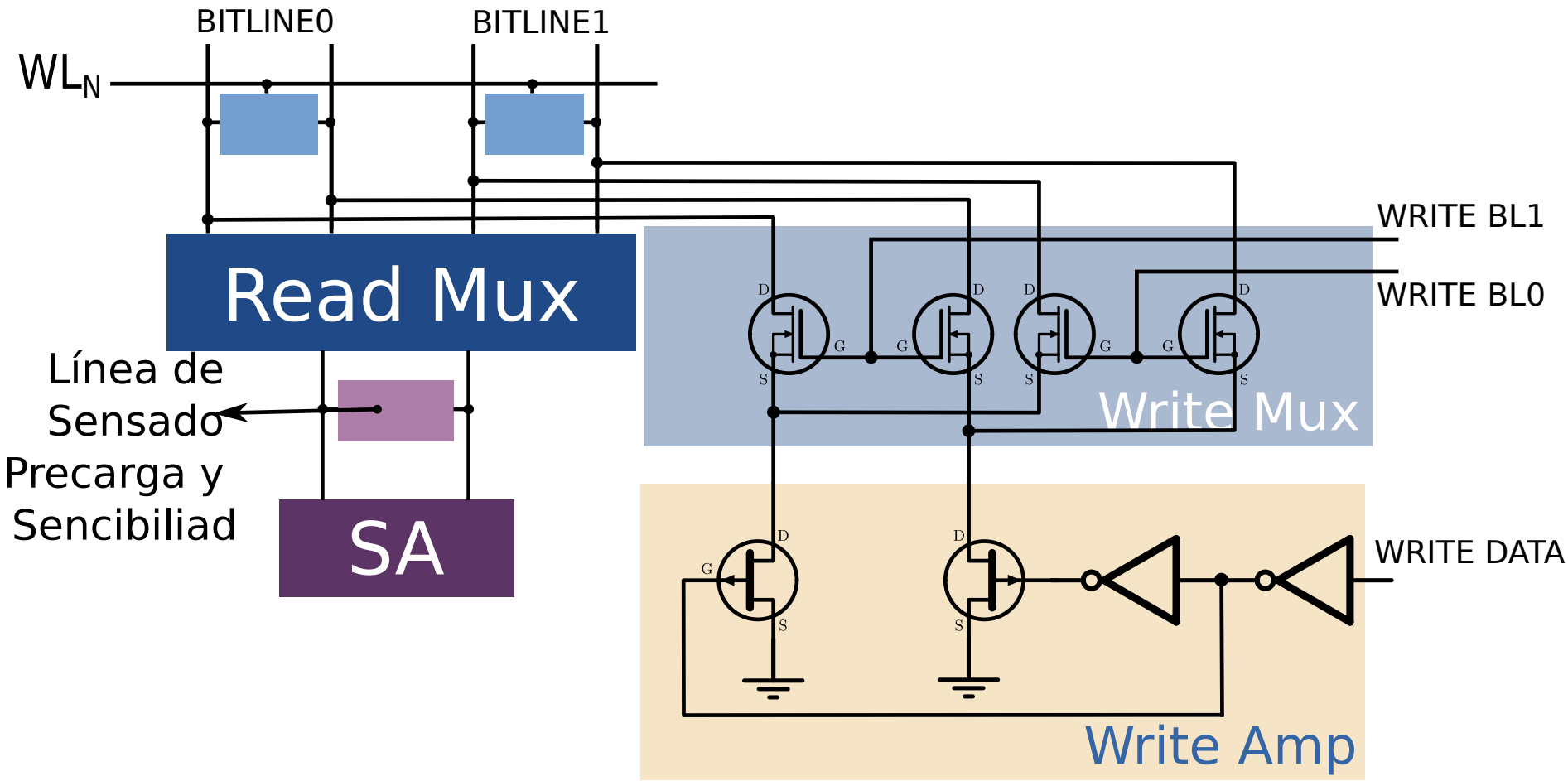


Amplificadores de Escritura

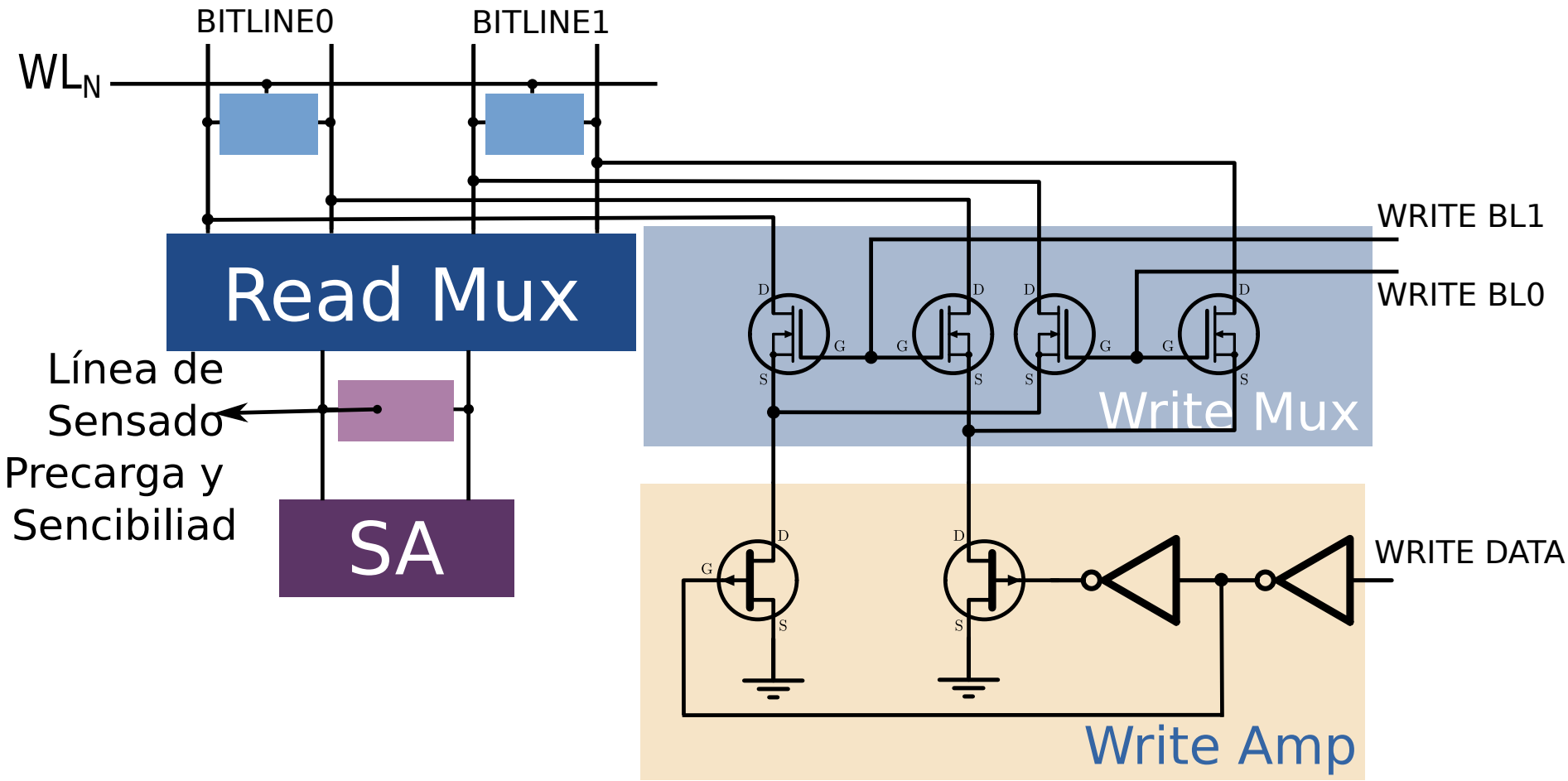


En NMOS de la derecha satura cuando se activa su Gate lo cual fuerza la línea **BLB** a Tierra. Esto hace que el NMOS de paso del lado derecho que tiene en **WL** un **'1'** sostenga el potencial 0 en **BLB**. El inversor fuerza un **'1'** en el Source del NMOS de paso izquierdo, y mantiene la precarga de **BL** dejando el biestable en este nuevo estado: **BL** = **'1'** y **BLB** = **'0'**

Amplificadores de Escritura



Amplificadores de Escritura



En este diagrama vemos el circuito completo del Amplificador de Escritura y lo integramos al resto de la estructura ya explicada para verlo de manera mas integral

Amplificadores de Escritura

Amplificadores de Escritura

- Los dos NMOS del Amplificador de Escritura junto con los transistores del Multiplexor, se calculan de modo de asegurar la descarga de la bitline en la ventana de tiempo asignada.

Amplificadores de Escritura

- Los dos NMOS del Amplificador de Escritura junto con los transistores del Multiplexor, se calculan de modo de asegurar la descarga de la bitline en la ventana de tiempo asignada.
- Normalmente la ventana de tiempo crítica es la de lectura, debido a la mayor complejidad de los circuitos involucrados de modo que en el amplificador y multiplexor de escritura los diseñadores solo necesitan moderar los parámetros que regulan este tiempo.

Amplificadores de Escritura

- Los dos NMOS del Amplificador de Escritura junto con los transistores del Multiplexor, se calculan de modo de asegurar la descarga de la bitline en la ventana de tiempo asignada.
- Normalmente la ventana de tiempo crítica es la de lectura, debido a la mayor complejidad de los circuitos involucrados de modo que en el amplificador y multiplexor de escritura los diseñadores solo necesitan moderar los parámetros que regulan este tiempo.
- Para disminuir la potencia disipada para descargar la bitline, los métodos que pueden implementarse terminan requiriendo modificaciones en la topología de la celda.

Amplificadores de Escritura

- Los dos NMOS del Amplificador de Escritura junto con los transistores del Multiplexor, se calculan de modo de asegurar la descarga de la bitline en la ventana de tiempo asignada.
- Normalmente la ventana de tiempo crítica es la de lectura, debido a la mayor complejidad de los circuitos involucrados de modo que en el amplificador y multiplexor de escritura los diseñadores solo necesitan moderar los parámetros que regulan este tiempo.
- Para disminuir la potencia disipada para descargar la bitline, los métodos que pueden implementarse terminan requiriendo modificaciones en la topología de la celda.
- Por ejemplo escritura en modo corriente, que implica incluir un transistor que conecte a los inversores en contrafase, generando un estado quasi estable.

Amplificadores de Escritura

- Los dos NMOS del Amplificador de Escritura junto con los transistores del Multiplexor, se calculan de modo de asegurar la descarga de la bitline en la ventana de tiempo asignada.
- Normalmente la ventana de tiempo crítica es la de lectura, debido a la mayor complejidad de los circuitos involucrados de modo que en el amplificador y multiplexor de escritura los diseñadores solo necesitan moderar los parámetros que regulan este tiempo.
- Para disminuir la potencia disipada para descargar la bitline, los métodos que pueden implementarse terminan requiriendo modificaciones en la topología de la celda.
- Por ejemplo escritura en modo corriente, que implica incluir un transistor que conecte a los inversores en contrafase, generando un estado quasi estable.
- Debe utilizarse un amplificador para generar la corriente necesaria para escribir el estado final.

Temario

Tipo de conexión

Tipo de conexión

- Las SRAM pueden utilizarse en forma sincrónica o asincrónica.

Tipo de conexión

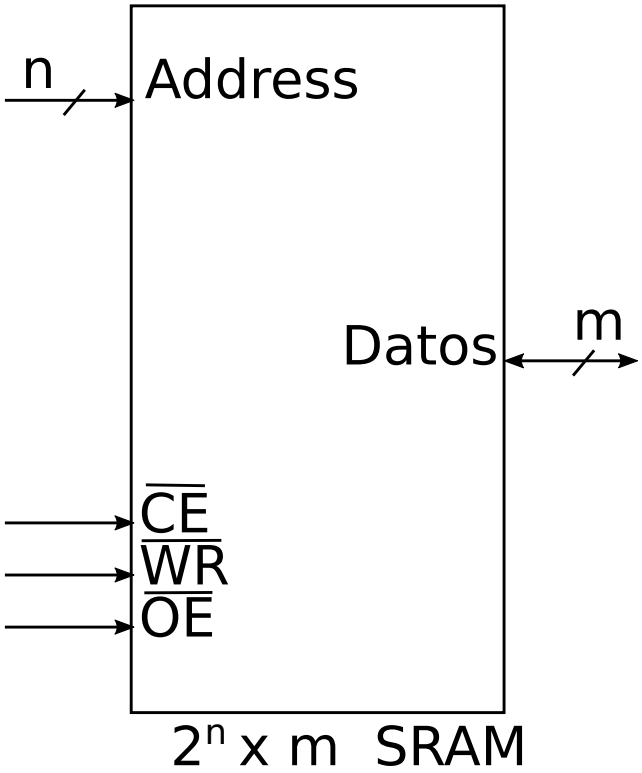
- Las SRAM pueden utilizarse en forma sincrónica o asincrónica.
- Las asincrónicas son las mas elementales: Se acceden mediante un simple juego de señales y se utilizan en sistemas en los que se requieren velocidades de acceso moderadas

Tipo de conexión

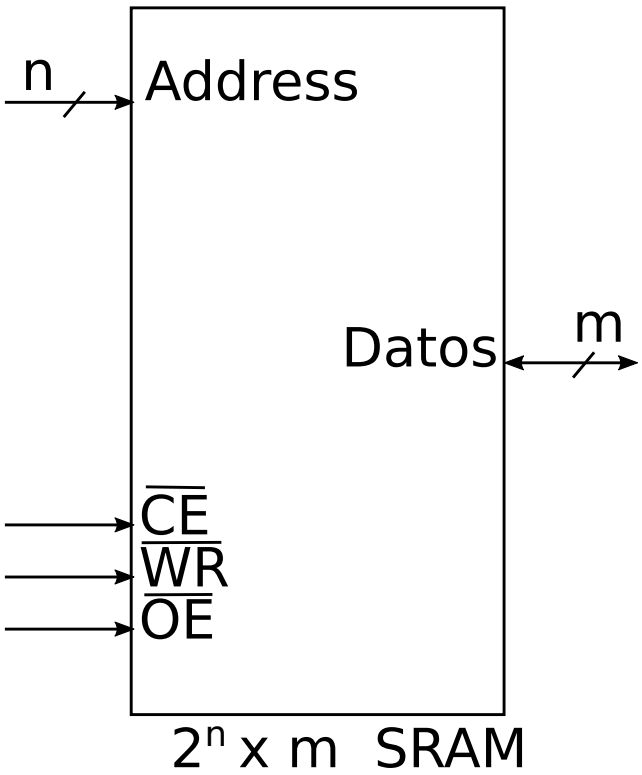
- Las SRAM pueden utilizarse en forma sincrónica o asincrónica.
- Las asincrónicas son las mas elementales: Se acceden mediante un simple juego de señales y se utilizan en sistemas en los que se requieren velocidades de acceso moderadas
- Para configuraciones cache o de SRAM de muy alta velocidad, se requieren técnicas de acceso mas sofisticadas. En estos casos los métodos Sincrónicos , como modos burst son los mas indicados

Interfaz de conexionado Asincrónica

Interfaz de conexionado Asincrónica

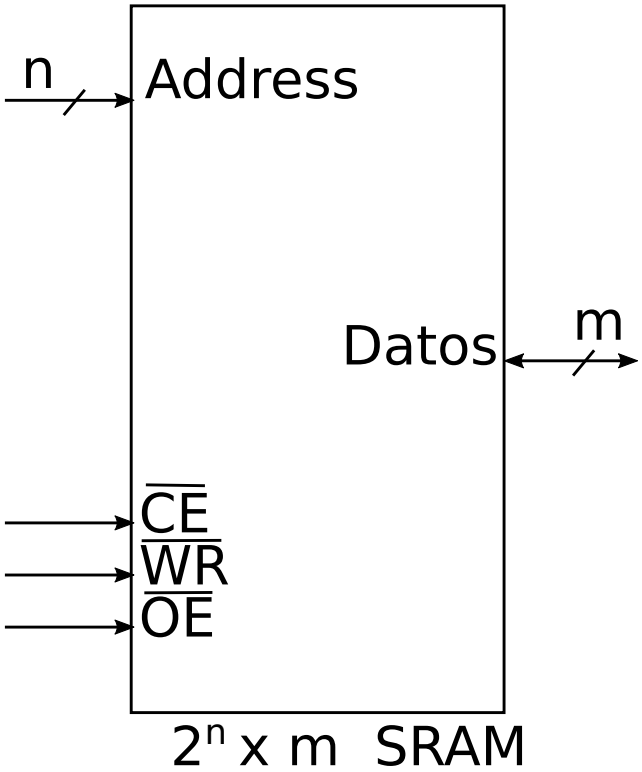


Interfaz de conexionado Asincrónica

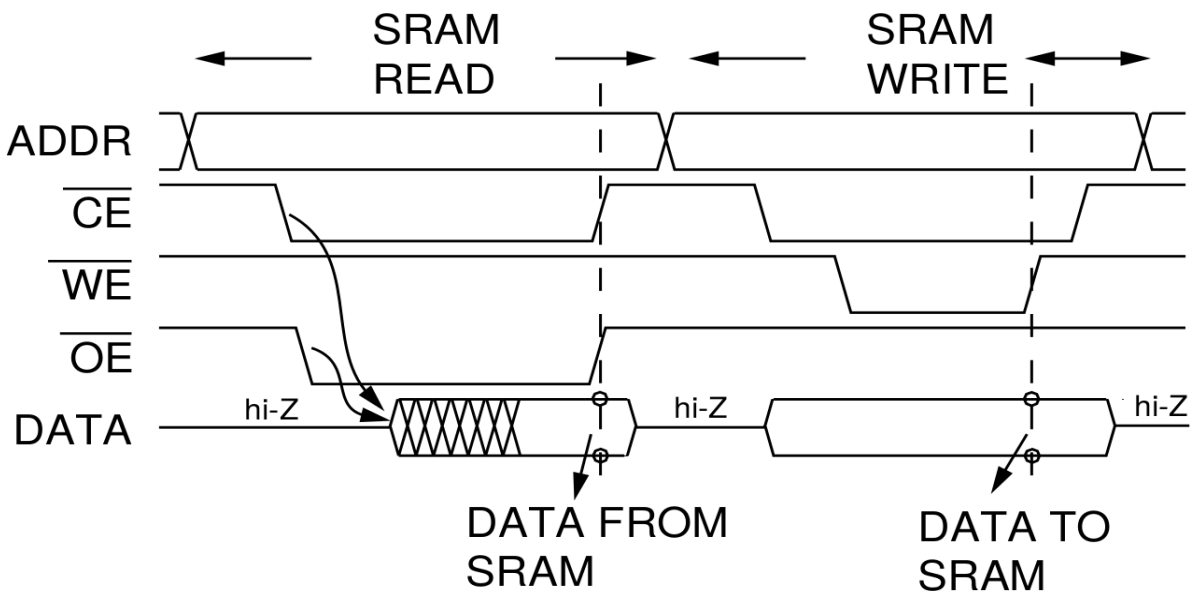


$\overline{CE}$	$\overline{WR}$	$\overline{OE}$	Tipo de Acceso
0	0	1	Escritura SRAM
0	1	0	Lectura SRAM
1	X	X	SRAM Deshabilitada

Interfaz de conexionado Asincrónica



$\overline{CE}$	$\overline{WR}$	$\overline{OE}$	Tipo de Acceso
0	0	1	Escritura SRAM
0	1	0	Lectura SRAM
1	X	X	SRAM Deshabilitada



Interfaz de conexionado Sincrónica (pipelined-burst)

Interfaz de conexionado Sincrónica (pipelined-burst)

- En ocasiones se requiere acceso ultra rápido a SRAM. Ej: Caches.

Interfaz de conexionado Sincrónica (pipelined-burst)

- En ocasiones se requiere acceso ultra rápido a SRAM. Ej: Caches.
- Las celdas y la lógica interna normalmente está diseñada al límite y no es terreno para explorar mejoras sustanciales en la velocidad.

Interfaz de conexionado Sincrónica (pipelined-burst)

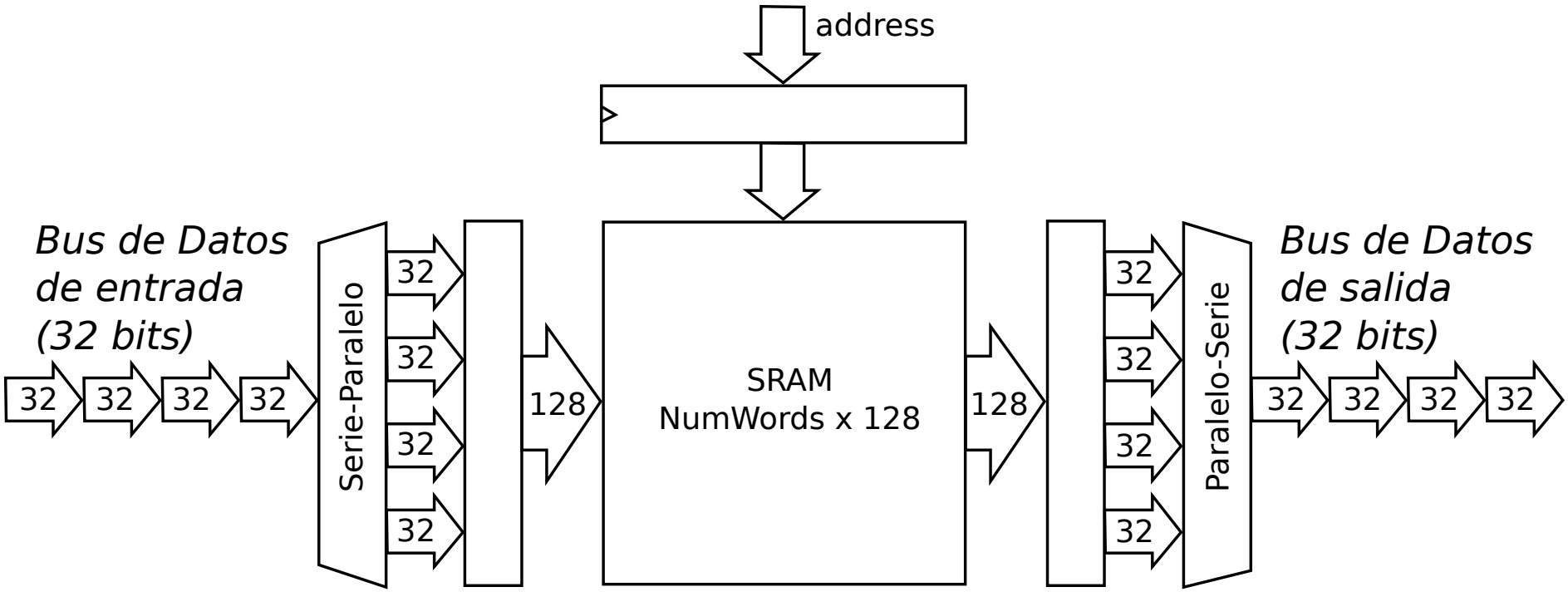
- En ocasiones se requiere acceso ultra rápido a SRAM. Ej: Caches.
- Las celdas y la lógica interna normalmente está diseñada al límite y no es terreno para explorar mejoras sustanciales en la velocidad.
- En estos casos se recurre tanto en SRAM discretas o embedded, a diferentes técnicas de aceleración.

Interfaz de conexionado Sincrónica (pipelined-burst)

- En ocasiones se requiere acceso ultra rápido a SRAM. Ej: Caches.
- Las celdas y la lógica interna normalmente está diseñada al límite y no es terreno para explorar mejoras sustanciales en la velocidad.
- En estos casos se recurre tanto en SRAM discretas o embedded, a diferentes técnicas de aceleración.
- La mas utilizada es *pipelined-burs*t (ráfagas en cadena)

Interfaz de conexionado Sincrónica (pipelined-burst)

- En ocasiones se requiere acceso ultra rápido a SRAM. Ej: Caches.
- Las celdas y la lógica interna normalmente está diseñada al límite y no es terreno para explorar mejoras sustanciales en la velocidad.
- En estos casos se recurre tanto en SRAM discretas o embedded, a diferentes técnicas de aceleración.
- La mas utilizada es *pipelined-burst* (ráfagas en cadena)



Interfaz de conexionado Sincrónica (pipelined-burst)

- El principal concepto está en la interfaz de entrada y salida con el Bus de datos en la cual se realiza un trabajo de wrapping que permite comunicarse con los circuitos externos a una frecuencia mayor que la que se utiliza para acceder a la matriz de celdas (limitada por valores máximos tolerables de disipación de potencia).

Interfaz de conexionado Sincrónica (pipelined-burst)

- El principal concepto está en la interfaz de entrada y salida con el Bus de datos en la cual se realiza un trabajo de wrapping que permite comunicarse con los circuitos externos a una frecuencia mayor que la que se utiliza para acceder a la matriz de celdas (limitada por valores máximos tolerables de disipación de potencia).
- Internamente se organiza en palabras de acceso mas anchas que el bus.

Interfaz de conexionado Sincrónica (pipelined-burst)

- El principal concepto está en la interfaz de entrada y salida con el Bus de datos en la cual se realiza un trabajo de wrapping que permite comunicarse con los circuitos externos a una frecuencia mayor que la que se utiliza para acceder a la matriz de celdas (limitada por valores máximos tolerables de disipación de potencia).
- Internamente se organiza en palabras de acceso mas anchas que el bus.
- La cantidad de veces de ensanchamiento es la cantidad deseada de ráfagas en cadena ($\mathbf{PB_{num}}$).

Interfaz de conexionado Sincrónica (pipelined-burst)

- El principal concepto está en la interfaz de entrada y salida con el Bus de datos en la cual se realiza un trabajo de wrapping que permite comunicarse con los circuitos externos a una frecuencia mayor que la que se utiliza para acceder a la matriz de celdas (limitada por valores máximos tolerables de disipación de potencia).
- Internamente se organiza en palabras de acceso mas anchas que el bus.
- La cantidad de veces de ensanchamiento es la cantidad deseada de ráfagas en cadena ($\mathbf{PB_{num}}$).
- En el ejemplo anterior con un bus de datos de 32 bits de ancho se accede a una Matriz de 128 bits de ancho de palabra.

Interfaz de conexionado Sincrónica (pipelined-burst)

- El principal concepto está en la interfaz de entrada y salida con el Bus de datos en la cual se realiza un trabajo de wrapping que permite comunicarse con los circuitos externos a una frecuencia mayor que la que se utiliza para acceder a la matriz de celdas (limitada por valores máximos tolerables de disipación de potencia).
- Internamente se organiza en palabras de acceso mas anchas que el bus.
- La cantidad de veces de ensanchamiento es la cantidad deseada de ráfagas en cadena ($\mathbf{PB_{num}}$).
- En el ejemplo anterior con un bus de datos de 32 bits de ancho se accede a una Matriz de 128 bits de ancho de palabra.
- La interfaz serie paralelo (SPI) escribe $\mathbf{PB_{num}}$ palabras de 32 bits a mayor frecuencia de trabajo antes de escribirlas de una vez en la matriz de celdas interna

Interfaz de conexionado Sincrónica (pipelined-burst)

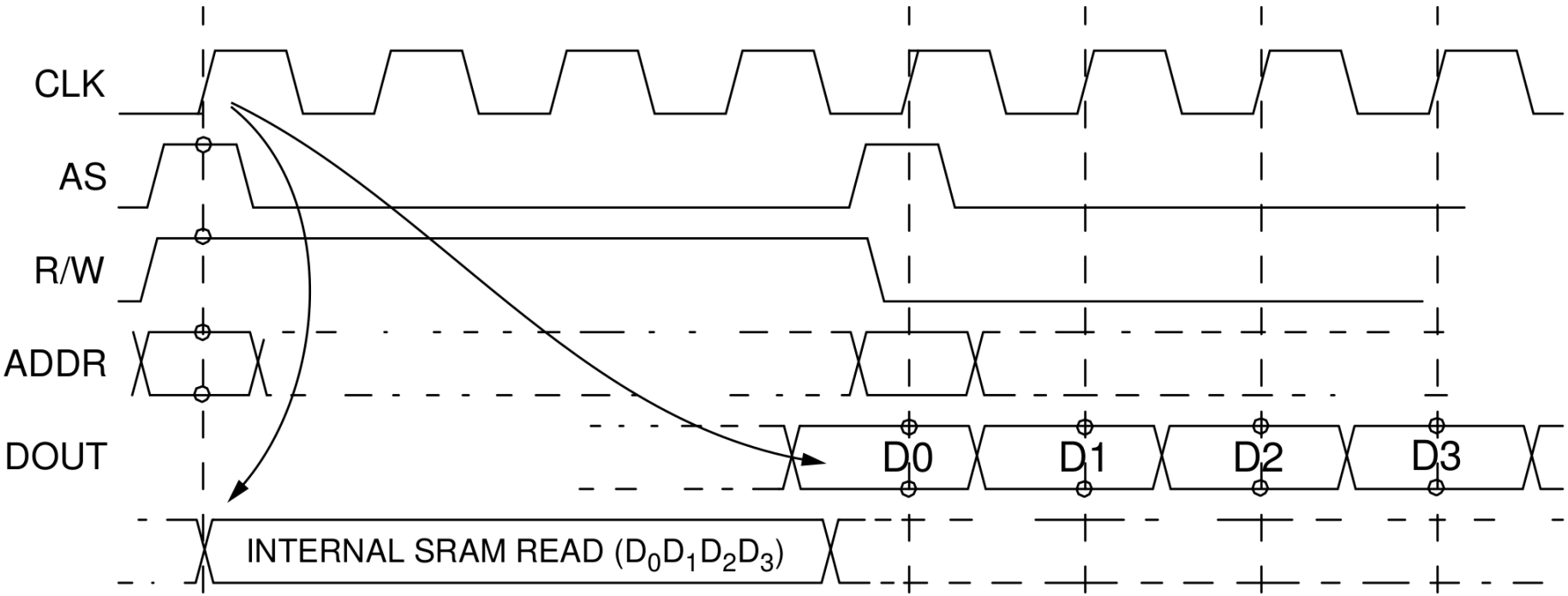
- Para leer la interfaz PSI lee la palabra más ancha en un solo ciclo, la divide en $\mathbf{PB_{num}}$ bloques que despacha en ráfaga a mayor frecuencia hacia el exterior.

Interfaz de conexionado Sincrónica (pipelined-burst)

- Para leer la interfaz PSI lee la palabra más ancha en un solo ciclo, la divide en $\mathbf{PB_{num}}$ bloques que despacha en ráfaga a mayor frecuencia hacia el exterior.
- Para ganar velocidad de acceso se sacrifica demora (latency): El dato leído en el ciclo N-1, estará disponible en el ciclo N+$\mathbf{PB_{num}}$, ya que debe transcurrir el tiempo del PSI.

Interfaz de conexionado Sincrónica (pipelined-burst)

- Para leer la interfaz PSI lee la palabra más ancha en un solo ciclo, la divide en $\mathbf{PB_{num}}$ bloques que despacha en ráfaga a mayor frecuencia hacia el exterior.
- Para ganar velocidad de acceso se sacrifica demora (latency): El dato leído en el ciclo N-1, estará disponible en el ciclo N+$\mathbf{PB_{num}}$, ya que debe transcurrir el tiempo del PSI.



Caso Práctico: R1WV6416R Renesas

- Se trata de una SRAM de 64 Mbits organizada en 4 M words de 16 bits de ancho (también es posible usarla como 8M words x8).

Caso Práctico: R1WV6416R Renesas

- Se trata de una SRAM de 64 Mbits organizada en 4 M words de 16 bits de ancho (también es posible usarla como 8M words x8).
- Alimentación 2.7V a 3.6V.

Caso Práctico: R1WV6416R Renesas

- Se trata de una SRAM de 64 Mbits organizada en 4 M words de 16 bits de ancho (también es posible usarla como 8M words x8).
- Alimentación 2.7V a 3.6V.
- Tiempo de acceso típico entre 55nseg y 70 nseg (depende del modelo)

Caso Práctico: R1WV6416R Renesas

- Se trata de una SRAM de 64 Mbits organizada en 4 M words de 16 bits de ancho (también es posible usarla como 8M words x8).
- Alimentación 2.7V a 3.6V.
- Tiempo de acceso típico entre 55nseg y 70 nseg (depende del modelo)
- Disipación de Potencia promedio 0.7 Watts

Caso Práctico: R1WV6416R Renesas

- Se trata de una SRAM de 64 Mbits organizada en 4 M words de 16 bits de ancho (también es posible usarla como 8M words x8).
- Alimentación 2.7V a 3.6V.
- Tiempo de acceso típico entre 55nseg y 70 nseg (depende del modelo)
- Disipación de Potencia promedio 0.7 Watts
- Corriente de leakage del orden de 1μA, y Corriente de operación de 60 mA en máxima velocidad de acceso. En modo standby (mínimo consumo) la corriente promedio es 0,3 mA.

Caso Práctico: R1WV6416R Renesas

- Se trata de una SRAM de 64 Mbits organizada en 4 M words de 16 bits de ancho (también es posible usarla como 8M words x8).
- Alimentación 2.7V a 3.6V.
- Tiempo de acceso típico entre 55nseg y 70 nseg (depende del modelo)
- Disipación de Potencia promedio 0.7 Watts
- Corriente de leakage del orden de 1μA, y Corriente de operación de 60 mA en máxima velocidad de acceso. En modo standby (mínimo consumo) la corriente promedio es 0,3 mA.
- Capacidad de entrada promedio 20pF.

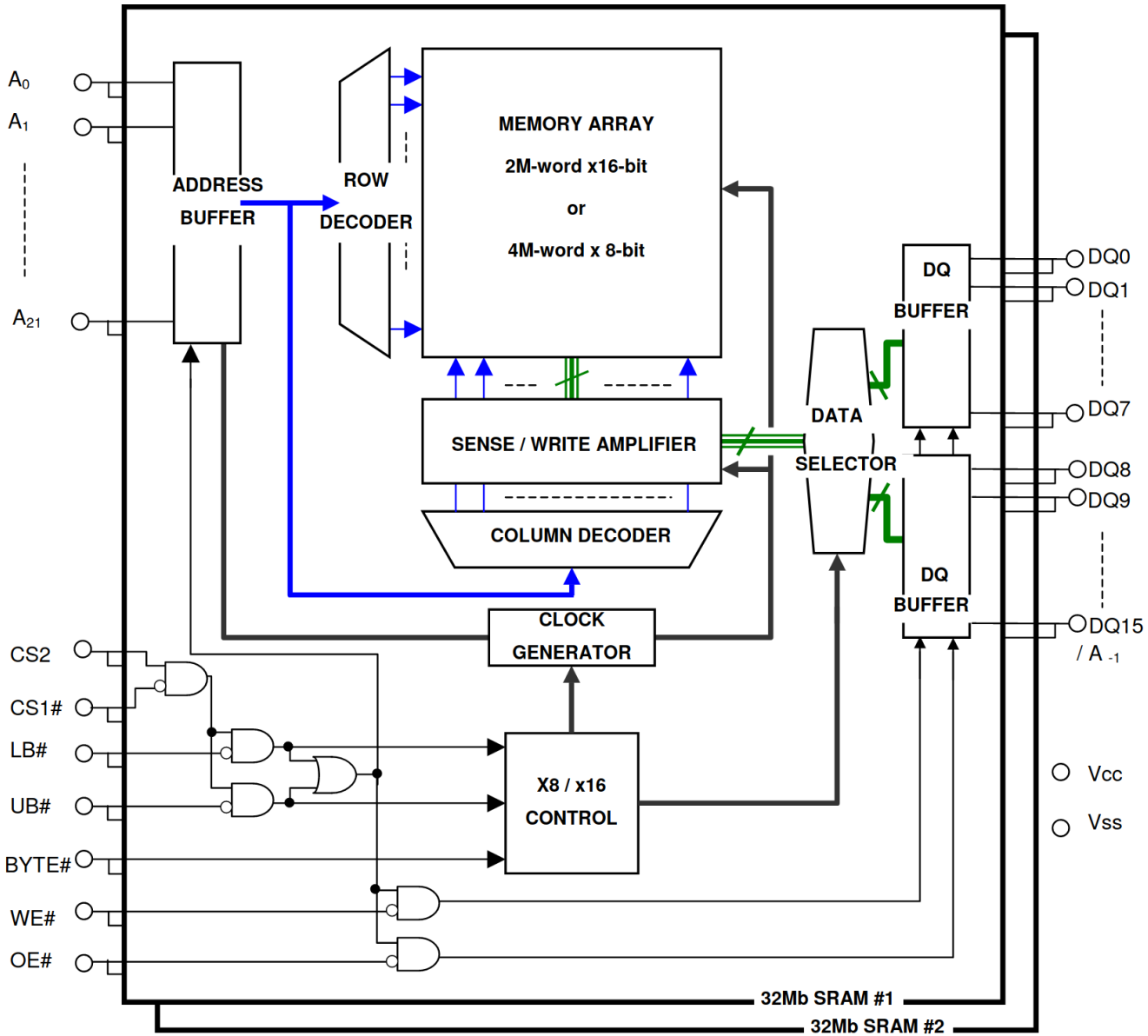
Caso Práctico: R1WV6416R Renesas

- Se trata de una SRAM de 64 Mbits organizada en 4 M words de 16 bits de ancho (también es posible usarla como 8M words x8).
- Alimentación 2.7V a 3.6V.
- Tiempo de acceso típico entre 55nseg y 70 nseg (depende del modelo)
- Disipación de Potencia promedio 0.7 Watts
- Corriente de leakage del orden de 1μA, y Corriente de operación de 60 mA en máxima velocidad de acceso. En modo standby (mínimo consumo) la corriente promedio es 0,3 mA.
- Capacidad de entrada promedio 20pF.
- Encapsulado TSOP μTSOP o f-BGA

Caso Práctico: R1WV6416R Renesas

- Se trata de una SRAM de 64 Mbits organizada en 4 M words de 16 bits de ancho (también es posible usarla como 8M words x8).
- Alimentación 2.7V a 3.6V.
- Tiempo de acceso típico entre 55nseg y 70 nseg (depende del modelo)
- Disipación de Potencia promedio 0.7 Watts
- Corriente de leakage del orden de 1μA, y Corriente de operación de 60 mA en máxima velocidad de acceso. En modo standby (mínimo consumo) la corriente promedio es 0,3 mA.
- Capacidad de entrada promedio 20pF.
- Encapsulado TSOP μTSOP o f-BGA
- Precio aproximado u$s 100

Caso Práctico: R1WV6416R Renesas



Caso Práctico: R1WV6416R Renesas

CS1#	CS2	BYTE#	LB#	UB#	WE#	OE#	DQ0~7	DQ8~14	DQ15	Operation
H	X	X	X	X	X	X	High-Z	High-Z	High-Z	Stand-by
X	L	X	X	X	X	X	High-Z	High-Z	High-Z	Stand-by
X	X	H	H	H	X	X	High-Z	High-Z	High-Z	Stand-by
L	H	H	L	H	L	X	Din	High-Z	High-Z	Write in lower byte
L	H	H	L	H	H	L	Dout	High-Z	High-Z	Read in lower byte
L	H	H	L	H	H	H	High-Z	High-Z	High-Z	Output disable
L	H	H	H	L	L	X	High-Z	Din	Din	Write in upper byte
L	H	H	H	L	H	L	High-Z	Dout	Dout	Read in upper byte
L	H	H	H	L	H	H	High-Z	High-Z	High-Z	Output disable
L	H	H	L	L	L	X	Din	Din	Din	Word write
L	H	H	L	L	H	L	Dout	Dout	Dout	Word read
L	H	H	L	L	H	H	High-Z	High-Z	High-Z	Output disable
L	H	L	L	L	L	X	Din	High-Z	A-1	Byte write
L	H	L	L	L	H	L	Dout	High-Z	A-1	Byte read
L	H	L	L	L	H	H	High-Z	High-Z	A-1	Output disable

Note1. H: V_{IH} L:V_{IL} X: V_{IH} or V_{IL}

2. BYTE# pin is supported for 48-pin TSOP (I) and 52-pin µTSOP (II) packages.

3. When apply BYTE# =“L”, please assign LB#=UB#=“L”.

Caso Práctico: CY7C1318KV18 Cypress

- Se trata de una SRAM de 18 Mbits organizada en 1 M words de 18bits de ancho.

Caso Práctico: CY7C1318KV18 Cypress

- Se trata de una SRAM de 18 Mbits organizada en 1 M words de 18bits de ancho.
- Alimentación: Soporta V_{DD} de 1.5V y 1.8V.

Caso Práctico: CY7C1318KV18 Cypress

- Se trata de una SRAM de 18 Mbits organizada en 1 M words de 18bits de ancho.
- Alimentación: Soporta V_{DD} de 1.5V y 1.8V.
- Frecuencia de operación 333MHz. (Accedida en DDR se lee a 666 MHz)

Caso Práctico: CY7C1318KV18 Cypress

- Se trata de una SRAM de 18 Mbits organizada en 1 M words de 18bits de ancho.
- Alimentación: Soporta V_{DD} de 1.5V y 1.8V.
- Frecuencia de operación 333MHz. (Accedida en DDR se lee a 666 MHz)
- Pipelined-Burst de 2 palabras (Latency 1 ciclo)

Caso Práctico: CY7C1318KV18 Cypress

- Se trata de una SRAM de 18 Mbits organizada en 1 M words de 18bits de ancho.
- Alimentación: Soporta V_{DD} de 1.5V y 1.8V.
- Frecuencia de operación 333MHz. (Accedida en DDR se lee a 666 MHz)
- Pipelined-Burst de 2 palabras (Latency 1 ciclo)
- Capacidad de entrada promedio 4pF.

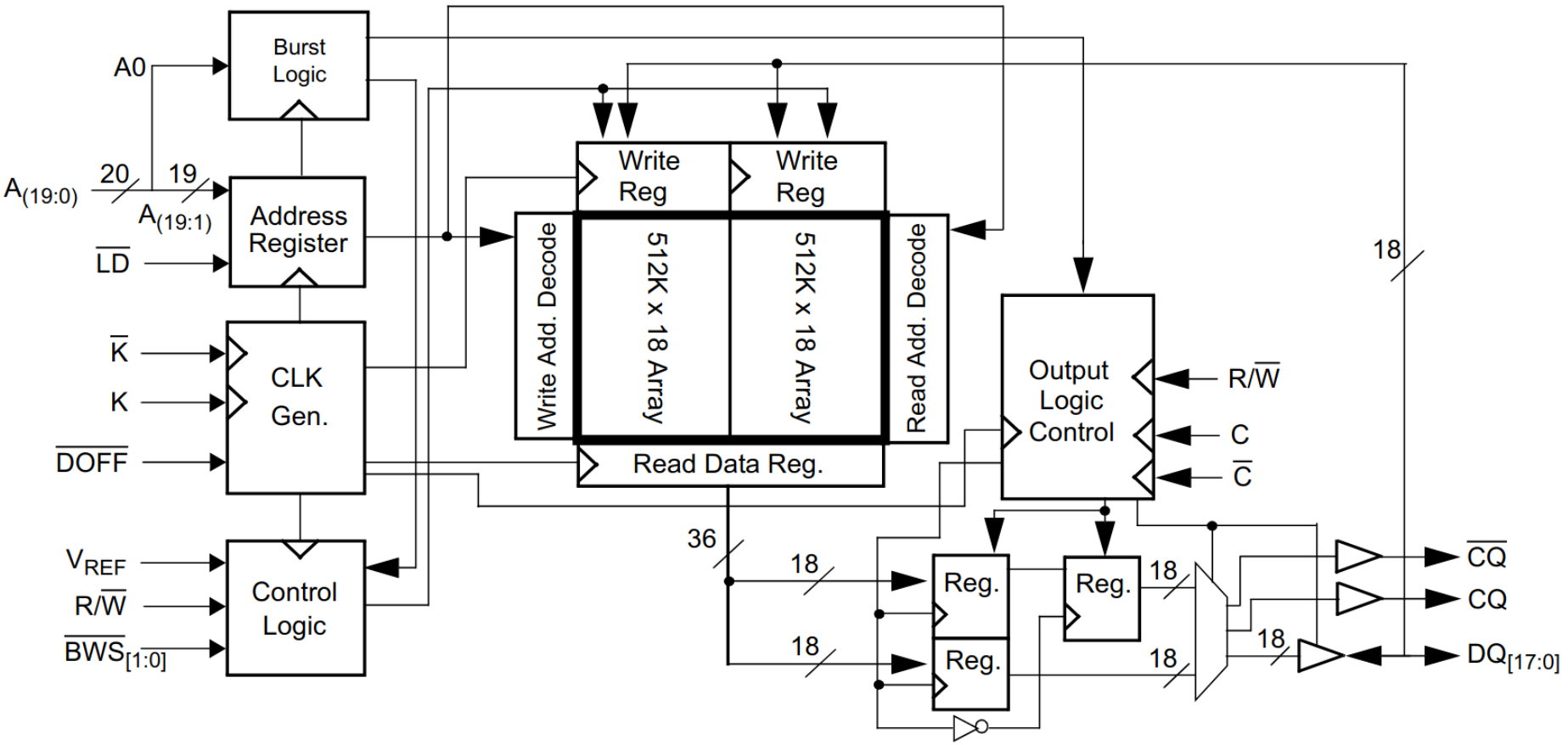
Caso Práctico: CY7C1318KV18 Cypress

- Se trata de una SRAM de 18 Mbits organizada en 1 M words de 18bits de ancho.
- Alimentación: Soporta V_{DD} de 1.5V y 1.8V.
- Frecuencia de operación 333MHz. (Accedida en DDR se lee a 666 MHz)
- Pipelined-Burst de 2 palabras (Latency 1 ciclo)
- Capacidad de entrada promedio 4pF.
- Encapsulado f-BGA

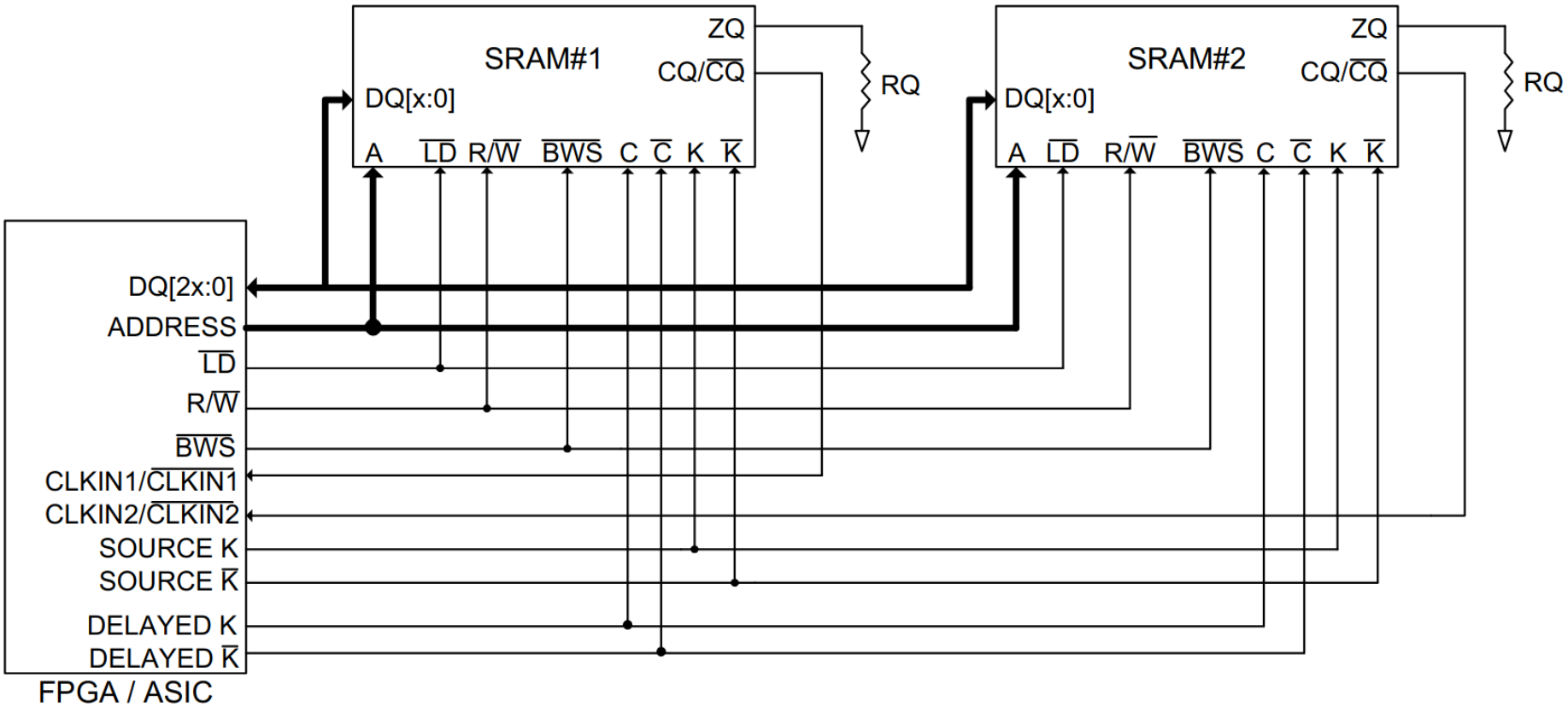
Caso Práctico: CY7C1318KV18 Cypress

- Se trata de una SRAM de 18 Mbits organizada en 1 M words de 18bits de ancho.
- Alimentación: Soporta V_{DD} de 1.5V y 1.8V.
- Frecuencia de operación 333MHz. (Accedida en DDR se lee a 666 MHz)
- Pipelined-Burst de 2 palabras (Latency 1 ciclo)
- Capacidad de entrada promedio 4pF.
- Encapsulado f-BGA
- Precio aproximado u$s 28

Caso Práctico: CY7C1318KV18 Cypress



Caso Práctico: CY7C1318KV18 Cypress



Temario

Estrategias para reducir Potencia y Energía

- Se demuestra [3], que la tensión de alimentación decrementa el delay en circuitos CMOS de alta velocidad, que es la inversa de la frecuencia máxima de operación:

$$T_d = \frac{C_L.V_{DD}}{\mu.C_{ox}.(W/L).(V_{DD} - V_t)^2} \tag{14}$$

[3] CMOS Circuit Design, Layout, and Simulation (4ta. Ed). R. JACOB BAKER. IEEE Press Wiley. 2018

Estrategias para reducir Potencia y Energía

- Se demuestra [3], que la tensión de alimentación decrementa el delay en circuitos CMOS de alta velocidad, que es la inversa de la frecuencia máxima de operación:

$$T_d = \frac{C_L.V_{DD}}{\mu.C_{ox}.(W/L).(V_{DD} - V_t)^2} \tag{14}$$

- μ es la movilidad de portadores, C_L la capacidad total de carga del nodo, C_{ox} la capacitancia de la capa de óxido, W/L la relación ancho / largo del transistor, V_t la tensión de umbral del transistor, V_{DD} la tensión de alimentación del circuito, y T_d el tiempo de delay del circuito, es decir la reciproca de la frecuencia de operación.

[3] CMOS Circuit Design, Layout, and Simulation (4ta. Ed). R. JACOB BAKER. IEEE Press Wiley. 2018

Estrategias para reducir Potencia y Energía

- Hemos visto que la tensión de umbral de conducción depende de las dimensiones del canal, y por lo tanto puede hacer proporcionalmente mayor el efecto del leakage.

Estrategias para reducir Potencia y Energía

- Hemos visto que la tensión de umbral de conducción depende de las dimensiones del canal, y por lo tanto puede hacer proporcionalmente mayor el efecto del leakage.
- Por su parte la tensión de alimentación cuando disminuye, disminuye cuadráticamente su efecto en el consumo de energía.

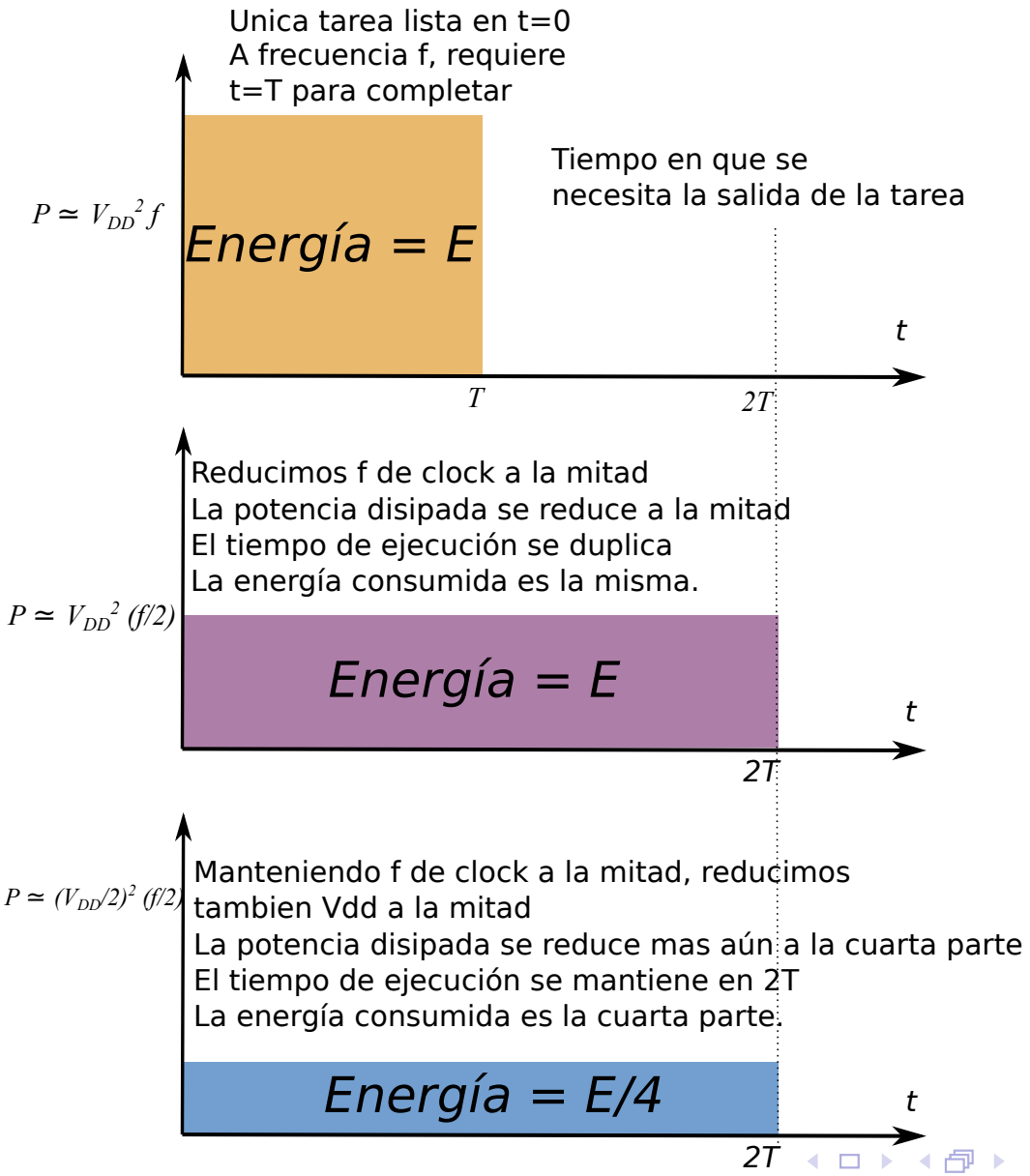
Estrategias para reducir Potencia y Energía

- Hemos visto que la tensión de umbral de conducción depende de las dimensiones del canal, y por lo tanto puede hacer proporcionalmente mayor el efecto del leakage.
- Por su parte la tensión de alimentación cuando disminuye, disminuye cuadráticamente su efecto en el consumo de energía.
- Por lo tanto hay variables que al mejorar el consumo perjudican la performance. Por ejemplo la tensión de alimentación. Por tal motivo la frecuencia máxima de operación de un chip muchas veces se establece de acuerdo con límites de compromiso entre ambas magnitudes.

Estrategias para reducir Potencia y Energía

- Hemos visto que la tensión de umbral de conducción depende de las dimensiones del canal, y por lo tanto puede hacer proporcionalmente mayor el efecto del leakage.
- Por su parte la tensión de alimentación cuando disminuye, disminuye cuadráticamente su efecto en el consumo de energía.
- Por lo tanto hay variables que al mejorar el consumo perjudican la performance. Por ejemplo la tensión de alimentación. Por tal motivo la frecuencia máxima de operación de un chip muchas veces se establece de acuerdo con límites de compromiso entre ambas magnitudes.
- Una técnica que empezaron a incluir los procesadores y memorias desde hace una década aproximadamente es DVS (Dynamic Voltage Scaling) que consiste en bajar la tensión de alimentación conforme se requiere disminuir la frecuenca en situaciones de baja carga de procesamiento.

Dynamic Voltage Scaling



Dynamic Voltage Scaling

- Dynamic Voltage Scaling no es trivial de implementar.

Dynamic Voltage Scaling

- Dynamic Voltage Scaling no es trivial de implementar.
- Requiere de circuitos que de manera Eurística determinen la conveniencia de reducir alimentación y frecuencia de trabajo, en correlación con el software del sistema.

Dynamic Voltage Scaling

- Dynamic Voltage Scaling no es trivial de implementar.
- Requiere de circuitos que de manera Eurística determinen la conveniencia de reducir alimentación y frecuencia de trabajo, en correlación con el software del sistema.
- Estos métodos favorecen cuadráticamente la reducción de Energía dinámica pero la energía estática si bien se reduce al bajar V_{DD}, lo hacen tan solo linealmente

Dynamic Voltage Scaling

- Dynamic Voltage Scaling no es trivial de implementar.
- Requiere de circuitos que de manera Eurística determinen la conveniencia de reducir alimentación y frecuencia de trabajo, en correlación con el software del sistema.
- Estos métodos favorecen cuadráticamente la reducción de Energía dinámica pero la energía estática si bien se reduce al bajar V_{DD}, lo hacen tan solo linealmente
- El leakage es cada vez un problema mas significativo y de mas compleja resolución.

Dynamic Voltage Scaling

- Dynamic Voltage Scaling no es trivial de implementar.
- Requiere de circuitos que de manera Eurística determinen la conveniencia de reducir alimentación y frecuencia de trabajo, en correlación con el software del sistema.
- Estos métodos favorecen cuadráticamente la reducción de Energía dinámica pero la energía estática si bien se reduce al bajar V_{DD}, lo hacen tan solo linealmente
- El leakage es cada vez un problema mas significativo y de mas compleja resolución.
- Si bien **DVS** fue pensado inicialmente para procesadores, su efecto en las memorias en especial las SRAM (Estáticas) es igual de efectivo ya que al bajar la frecuencia de operación se exige menos a l memoria y ésta también reduce su consumo.

Powering Down unused Blocks

- Se apagan los circuitos que no se utilizan

Powering Down unused Blocks

- Se apagan los circuitos que no se utilizan
- No es a nivel microarquitectura. Es a nivel circuital. En la arena del *Si*.

Powering Down unused Blocks

- Se apagan los circuitos que no se utilizan
- No es a nivel microarquitectura. Es a nivel circuital. En la arena del *Si*.
- ***Clock gating***. Utiliza un transistor MOSFET como llave en serie con la línea de clock.

Powering Down unused Blocks

- Se apagan los circuitos que no se utilizan
- No es a nivel microarquitectura. Es a nivel circuital. En la arena del *Si*.
- ***Clock gating***. Utiliza un transistor MOSFET como llave en serie con la línea de clock.

✓ Cuando un bloque no se usa, se interrumpe la señal de clock.

Powering Down unused Blocks

- Se apagan los circuitos que no se utilizan
- No es a nivel microarquitectura. Es a nivel circuital. En la arena del *Si*.
- ***Clock gating***. Utiliza un transistor MOSFET como llave en serie con la línea de clock.

✓ Cuando un bloque no se usa, se interrumpe la señal de clock.

✓ Los transistores que componen los circuitos estáticos sostienen sus salidas pero no conmutan.

Powering Down unused Blocks

- Se apagan los circuitos que no se utilizan
- No es a nivel microarquitectura. Es a nivel circuital. En la arena del *Si*.
- ***Clock gating***. Utiliza un transistor MOSFET como llave en serie con la línea de clock.

✓ Cuando un bloque no se usa, se interrumpe la señal de clock.

✓ Los transistores que componen los circuitos estáticos sostienen sus salidas pero no conmutan.

✓ No tiene efecto sobre el leakage que sigue ocurriendo.

Powering Down unused Blocks

- Se apagan los circuitos que no se utilizan
- No es a nivel microarquitectura. Es a nivel circuital. En la arena del *Si*.
- ***Clock gating***. Utiliza un transistor MOSFET como llave en serie con la línea de clock.

✓ Cuando un bloque no se usa, se interrumpe la señal de clock.

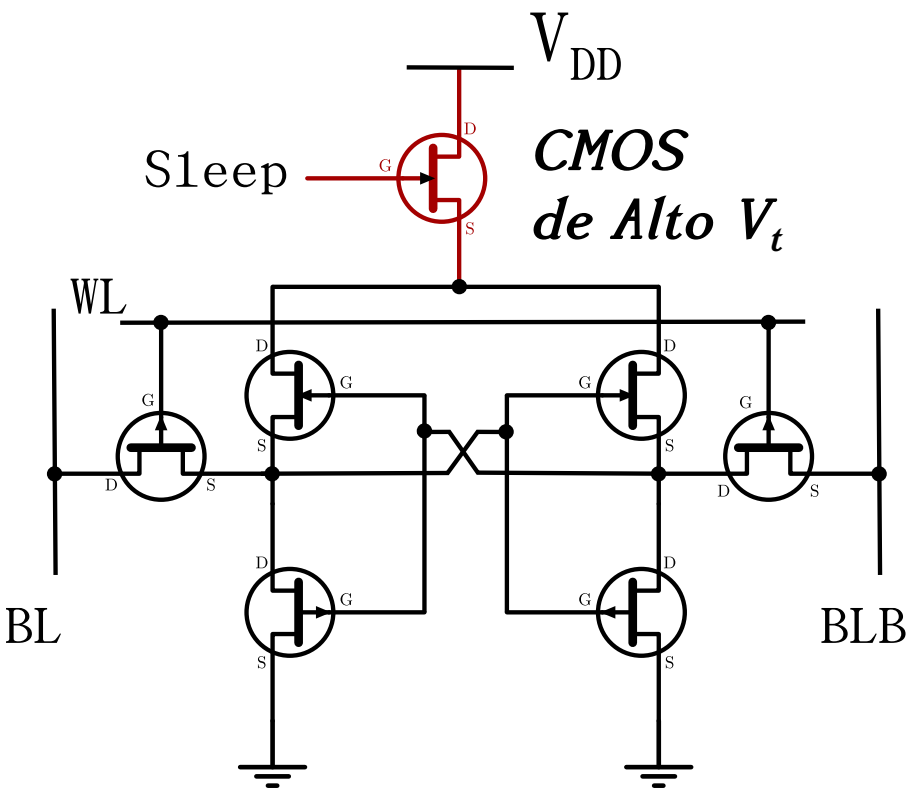
✓ Los transistores que componen los circuitos estáticos sostienen sus salidas pero no conmutan.

✓ No tiene efecto sobre el leakage que sigue ocurriendo.

- *Gated-V_{dd}*[4], fue la clave para minimizar el leakage, y su aplicación por excelencia es a las memorias estáticas (SRAM).

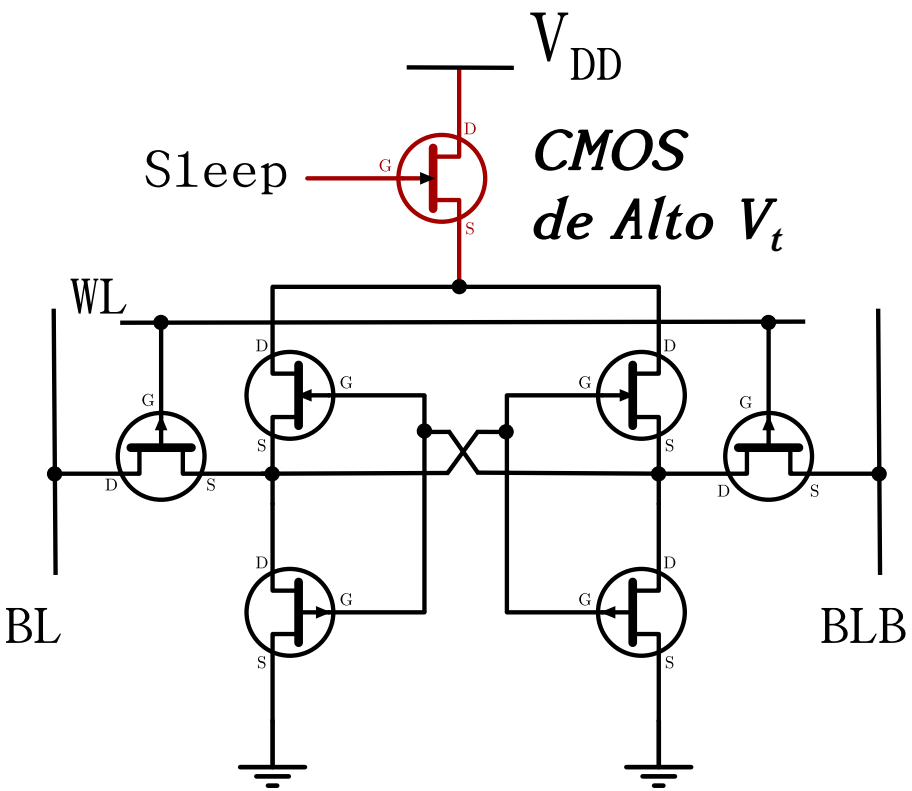
[4] ***Gated-Vdd***: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories. Power et. al. Proc. IEEE/ACM Int. Symp. on Low Power Electronics and Design (ISLPED), pp. 90–95, 2000.

Gating-V_{dd}



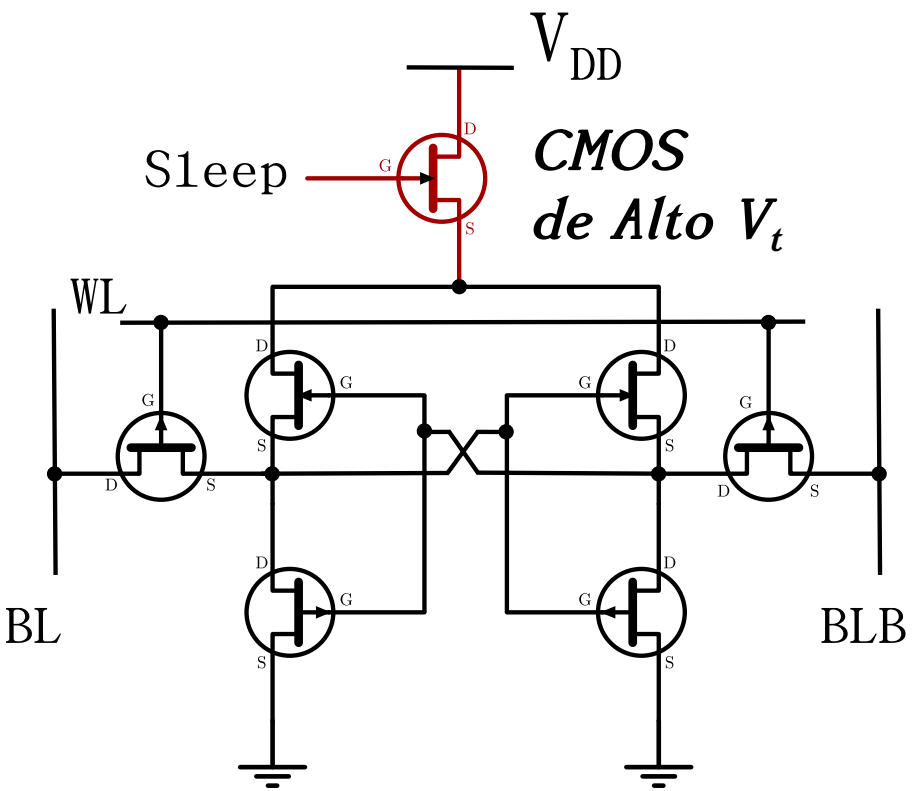
- Logra disminuir la actividad del dispositivo (Potencia dinámica), ya que tampoco conduce, aunque tenga clock.

Gating-V_{dd}



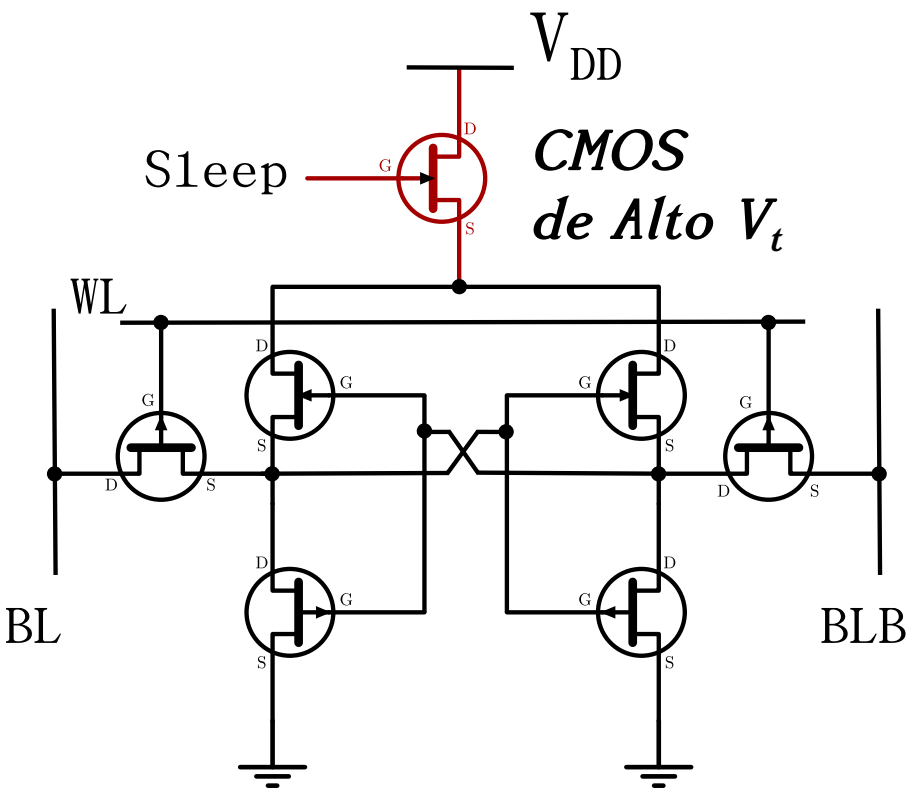
- Logra disminuir la actividad del dispositivo (Potencia dinámica), ya que tampoco conduce, aunque tenga clock.
- Obviamente el leakage es solo el del transistor que trabaja como llave.

Gating-V_{dd}



- Logra disminuir la actividad del dispositivo (Potencia dinámica), ya que tampoco conduce, aunque tenga clock.
- Obviamente el leakage es solo el del transistor que trabaja como llave.
- En el caso de las memorias SRAM con un solo transistor se pueden manejar múltiples celdas.

Gating-V_{dd}



- Logra disminuir la actividad del dispositivo (Potencia dinámica), ya que tampoco conduce, aunque tenga clock.
- Obviamente el leakage es solo el del transistor que trabaja como llave.
- En el caso de las memorias SRAM con un solo transistor se pueden manejar múltiples celdas.

Potencia de leakage en SRAM

Potencia de leakage en SRAM

- Es particularmente delicado el consumo de las SRAM ya que es su principal debilidad frente a las DRAM. El leakage considerando que aproximadamente el 50 % de los transistores están en estado de corte, es crítico.

Potencia de leakage en SRAM

- Es particularmente delicado el consumo de las SRAM ya que es su principal debilidad frente a las DRAM. El leakage considerando que aproximadamente el 50 % de los transistores están en estado de corte, es crítico.
- Una técnica es trabajar con Multithreshold, es decir manejar tensiones de umbral altas (bajo leakage) en zonas no críticas del chip (poco utilizadas), y guardar la tensión de umbral baja para las zonas mas críticas respecto de performance

Potencia de leakage en SRAM

- Es particularmente delicado el consumo de las SRAM ya que es su principal debilidad frente a las DRAM. El leakage considerando que aproximadamente el 50 % de los transistores están en estado de corte, es crítico.
- Una técnica es trabajar con Multithreshold, es decir manejar tensiones de umbral altas (bajo leakage) en zonas no críticas del chip (poco utilizadas), y guardar la tensión de umbral baja para las zonas mas críticas respecto de performance