

Optimizaciones de código

// TO-DO

Memoria Principal

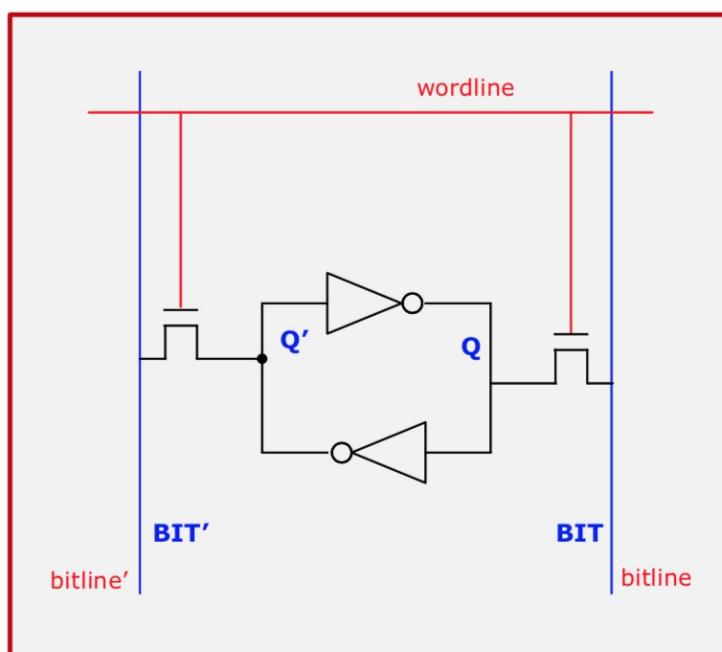
```
m[@] = x      // escritura  
x = m[@]      // lectura
```

La ram es volátil

- Los DVD no son memorias de acceso directo, porque en el DVD tú tienes puntos de acceso y a partir de ese punto, accedo secuencialmente.

La caché es estática (se guardan en biestables, 6-8 transistores) y la RAM es dinámica (porque se guarda en un condensador, 1-1.x transistores)

RAM estática (SRAM)

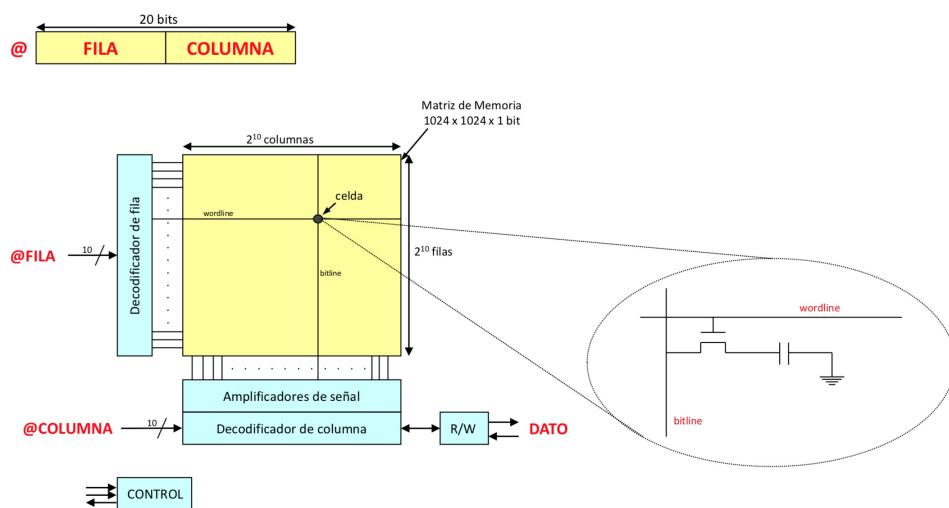
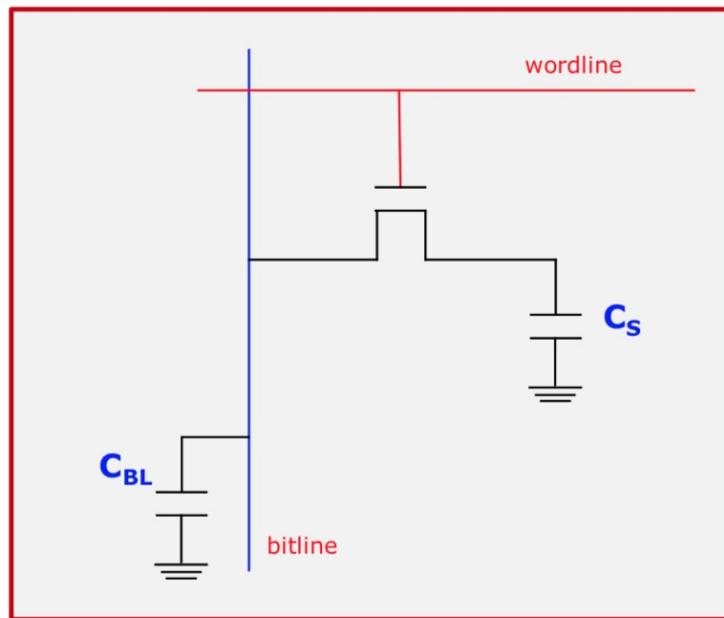


RAM dinámica (DRAM)

Cuando descargo el condensador, el bit se pierde, por lo tanto, una vez yo hago la

operación de lectura, tengo que volver a cargar el condensador con el bit que había.

Si yo llevo mucho tiempo sin leer el condensador, el condensador se va descargando, así que antes de que se descargue, tengo que volver a cargarlo. Eso se llama **REFRESCO** y se hace unas 20 veces por segundo.



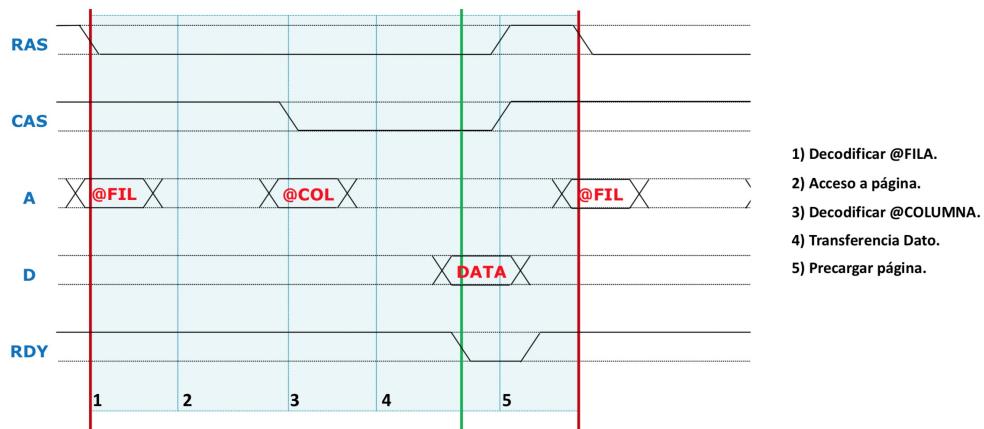
Por las mismas patas de la CPU se envía primero los 10 bits de fila y luego los 10 bits de columna. Así leo 1024 bits de golpe.

Abajo hay un amplificador de señal para saber con certeza si es un 0 o es un 1.

Luego me llegan los bits de columna, sé a cual quiero acceder y de esos 1024 envío aquellos que me han pedido.

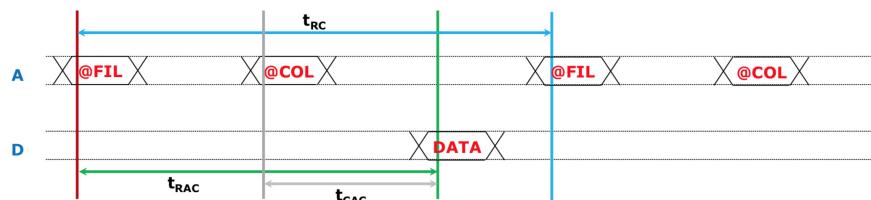
Finalmente vuelvo a cargar los condensadores con su valor original (al leerlos se descargan).

La **precarga** consiste en dejar el condensador en un estado intermedio de carga y así cuesta lo mismo cargar un 1 que descargar hasta un 0. Si un 1 \approx 5V y un 0 \approx 0V, precargo el condensador a 2,5V para que cargar un 1 no sea tan lento.



Antes todo era asíncrono. Y se lanzaban diversas señales (Row Address Strobe / Column Address Strobe / Ready) para indicar diversos estados.

Al ser **asíncrono** era bastante lento.



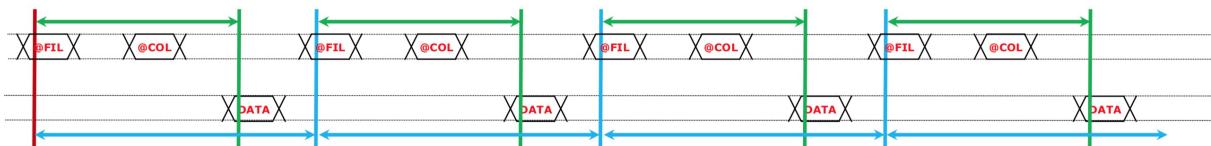
1. **Tiempo de acceso (t_{RAC}):**
retardo máximo desde que se suministra la dirección de fila hasta que se obtiene el dato
→ **latencia de memoria**.
2. **Tiempo de ciclo (t_{RC}):**
intervalo de tiempo mínimo entre dos accesos consecutivos a memoria
→ **ancho de banda**.
3. **Tiempo de acceso a columna (t_{CAC}):**
retardo máximo desde que se suministra la dirección de columna hasta que se obtiene el dato.

Esos tiempos son propios de la RAM.

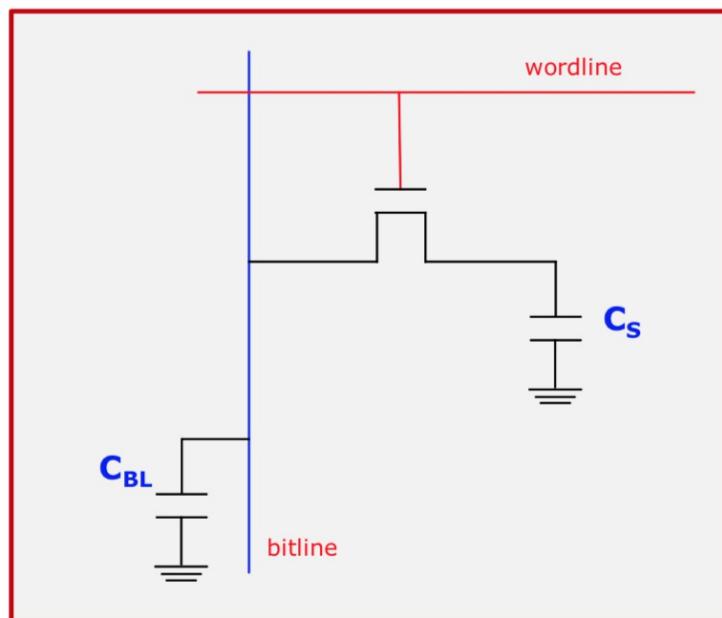
- La memoria principal suele estar organizada en DIMMs
- Se accede en paralelo a todos los chips . En un acceso se accede (p.e.) a 64 bits (8B)



- ¡Todos los accesos a memoria principal son para leer/escribir líneas de cache!
- Tamaño típico de una línea de cache: 32B
- Si queremos leer una línea de cache:
 - ¡Es necesario realizar 4 accesos consecutivos a memoria!



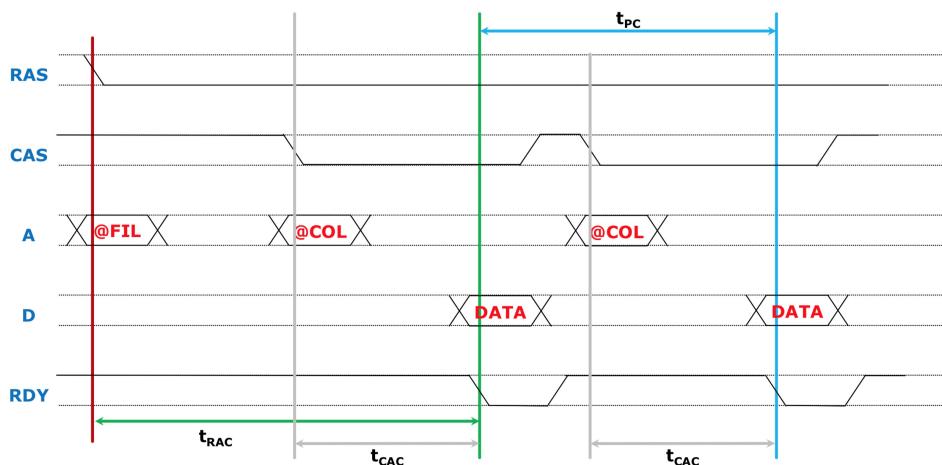
Hay que hacer 4 accesos para leer los 32B de una línea de caché.



Acceder a diferentes columnas de la misma fila

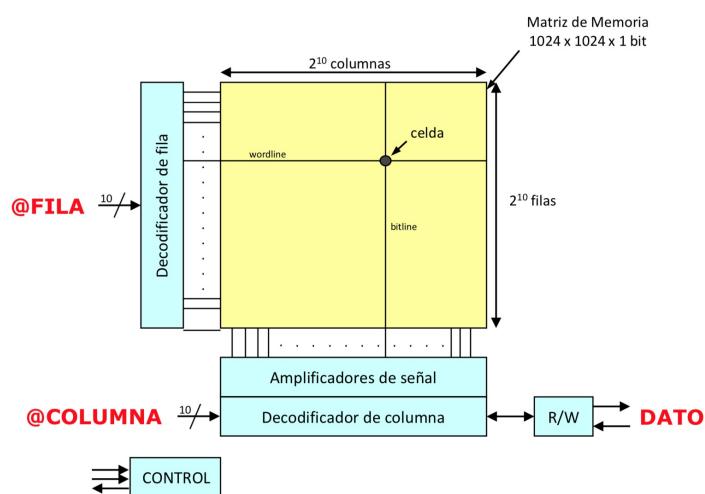
■ **Idea Fundamental:** una vez accedida la fila, se puede acceder a varias columnas simplemente cambiando la @COL.

→ Aprovechamos la **localidad espacial**

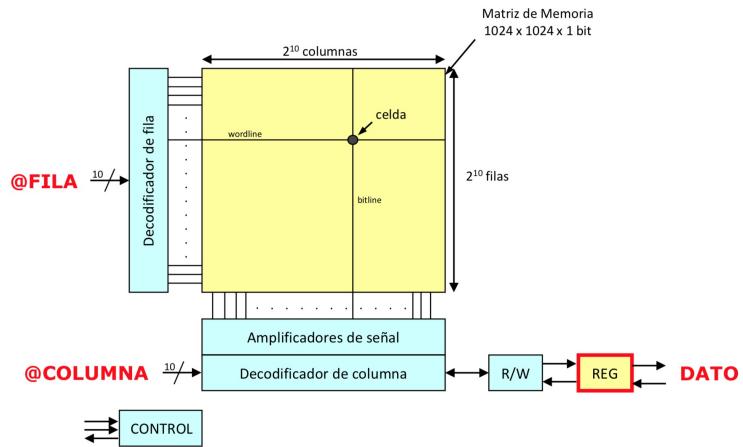


Segmentar la salida del dato

Esto:

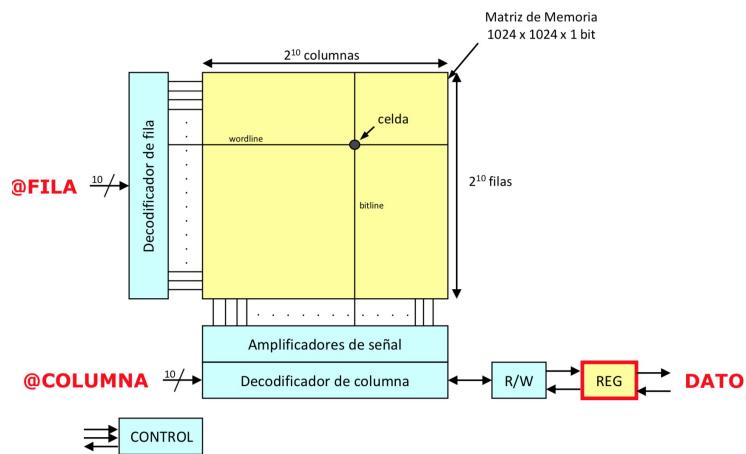


Pasa a ser:

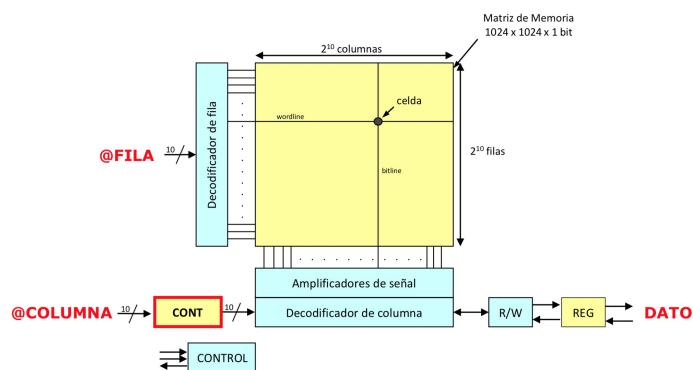


Contador para generar col+1, col+2, col+3...

Esto:



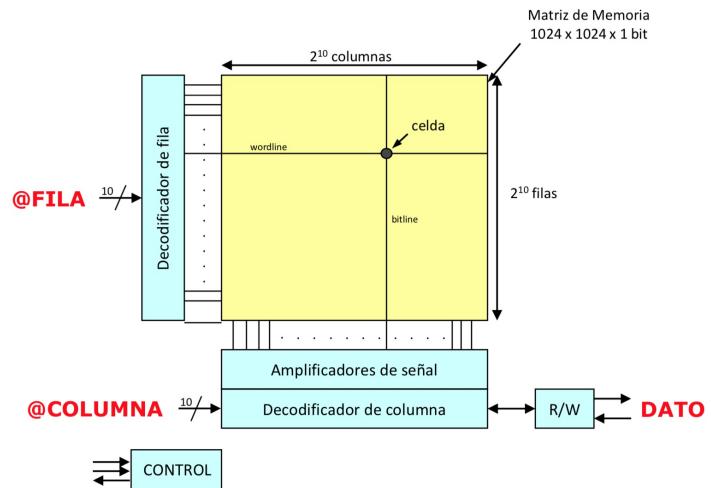
Pasa a ser:



Asíncrona --> Síncrona

Ahora, el fabricante te da un tiempo de ciclo con el cuál ya no necesitas comunicarte

con señales.

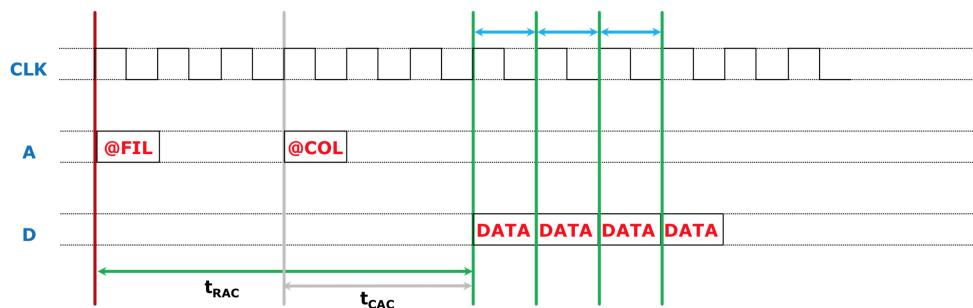


DDR (memorias actuales)

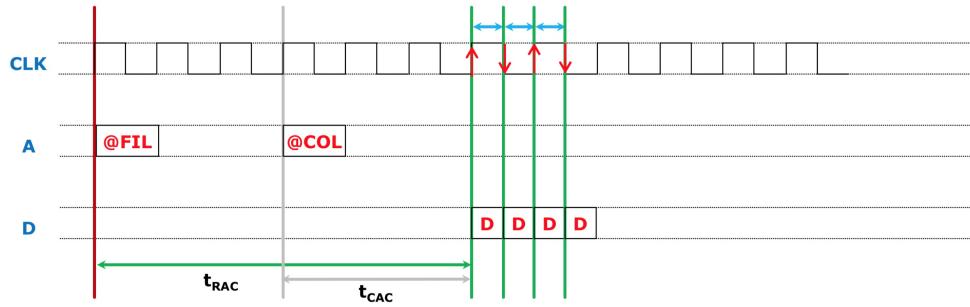
DDR SDRAM: Double Data Rate Synchronous Dynamic Random-Access Memory

Cada flanco de ciclo se lee un dato.

Esto:

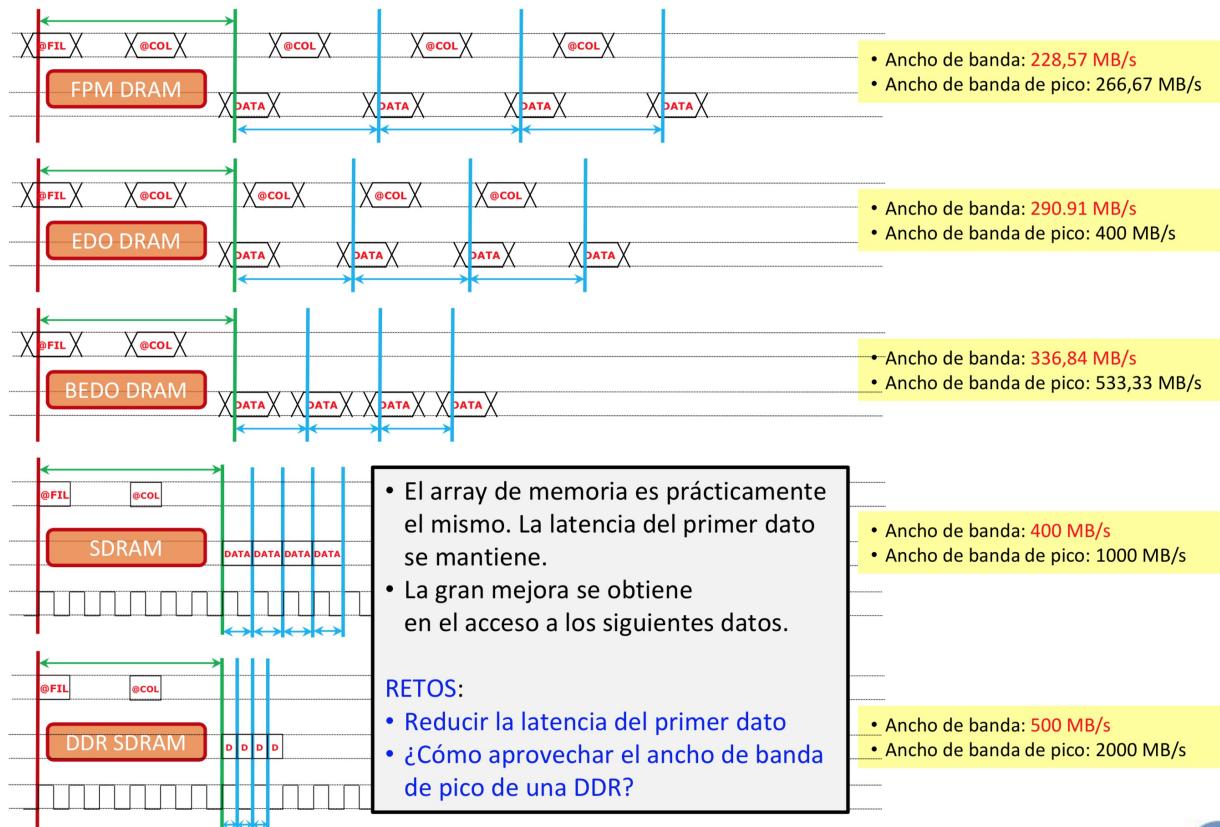


Pasa a ser:



La memoria se organiza por módulos (bancos) y mientras leo de uno, cargo el dato del otro.

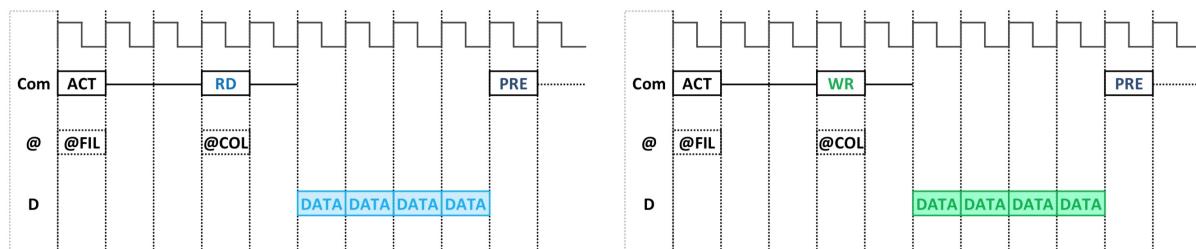
Unos cronogramas por aquí...



■ Todas las SDRAM, incluyendo los diferentes tipos de DDR, funcionan con comandos:

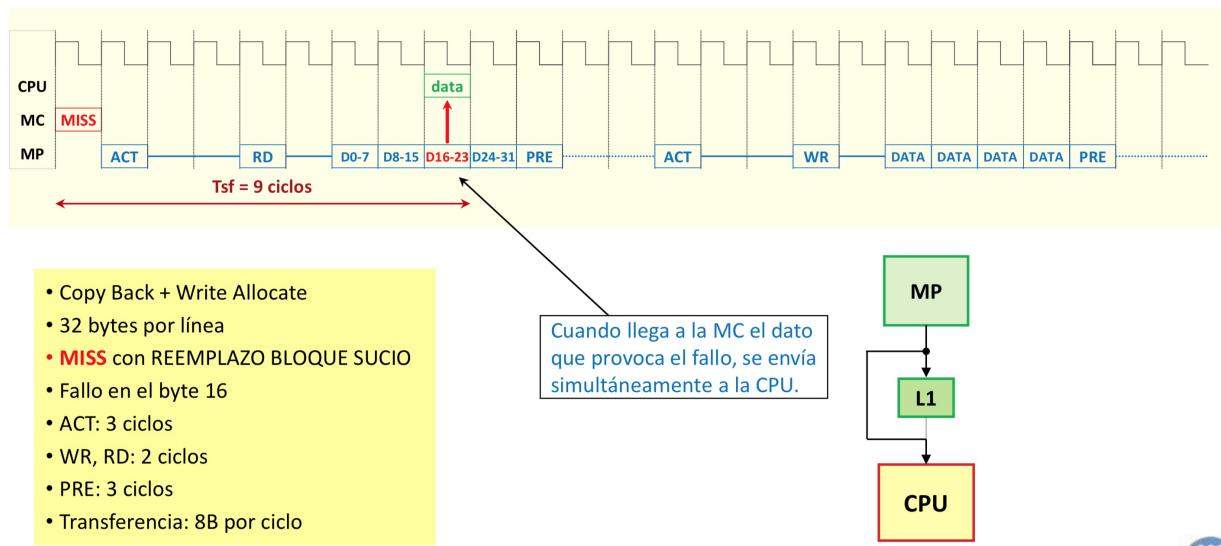
- ACTIVE (ACT)
- READ (RD)
- WRITE (WR)
- PRECHARGE (PRE)

■ Cronogramas de acceso en **Lectura** y en **Escritura**

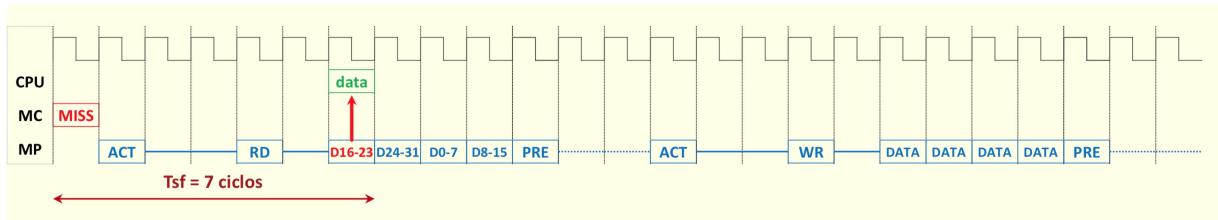


IMPORTANTE: ¡El tiempo de ciclo de la memoria no es mismo tiempo de ciclo que el sistema!

■ **Continuación Anticipada (*Early Restart*):**
en cuanto llega el dato que provoca el fallo, se envía al procesador



- **Transferencia en desorden:** se envía en primer lugar la palabra que ha provocado el fallo.



- Copy Back + Write Allocate
- 32 bytes por línea
- **MISS** con REEMPLAZO BLOQUE SUCIO
- Fallo en el byte 16
- ACT: 3 ciclos
- WR, RD: 2 ciclos
- PRE: 3 ciclos
- Transferencia: 8B por ciclo

En todos los casos, y para que estos mecanismos sean efectivos, cuando se pasa el dato al procesador y mientras se acaba de servir el fallo, la MC ha de ser capaz de recibir nuevos accesos (Cache no bloqueante).

