



# Microarquitectura - Memoria Dinámica

Alejandro Furfaro

9 de noviembre de 2020

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

- ▶ **Parámetros físicos:** No solo el pinout sino sus características eléctricas (resistencia, inductancia, capacitancia, etc.) y su comportamiento cuando se conectan con el hardware controlador.
- ▶ **Parámetros de funcionamiento:** La memoria de trabajo es la que se encuentra físicamente en el módulo de memoria. El tiempo de acceso define la memoria caché.
- ▶ **Parámetros de operación:** Definición de los comandos para el RDB y las señales de control que se necesitan para su funcionamiento.
- ▶ **Parámetros de tiempo:** Tiempo de acceso, tiempo de acceso para lectura y escritura.
- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

Pines: No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

Capacitancia: El tiempo que tarda en cargar y descargar

Inductancia: El tiempo que tarda en establecer la corriente

Resistencia: El tiempo que tarda en establecer el voltaje

Capacitancia: El tiempo que tarda en establecer la corriente

Inductancia: El tiempo que tarda en establecer el voltaje

Resistencia: El tiempo que tarda en establecer la corriente

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

Pines No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

Señalización Handshake con el hardware controlador

Signal Integrity En función de la frecuencia de trabajo

Encapsulado Define la manufacturabilidad.

Clock y sincro Define criterios de montaje en el PCB y adaptaciones eléctricas al bus

Timing Refresco, tiempo de acceso para lectura y escritura

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

**Pines** No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

Señalización Handshake con el hardware controlador

Signal Integrity En función de la frecuencia de trabajo

Encapsulado Define la manufacturabilidad.

Clock y sincro Define criterios de montaje en el PCB y adaptaciones eléctricas al bus

Timing Refresco, tiempo de acceso para lectura y escritura

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

**Pines** No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

**Señalización** Handshake con el hardware controlador

**Signal Integrity** En función de la frecuencia de trabajo

**Encapsulado** Define la manufacturabilidad.

**Clock y sincro** Define criterios de montaje en el PCB y adaptaciones eléctricas al bus

**Timing** Refresco, tiempo de acceso para lectura y escritura

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.



# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

**Pines** No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

**Señalización** Handshake con el hardware controlador

**Signal Integrity** En función de la frecuencia de trabajo

**Encapsulado** Define la manufacturabilidad.

**Clock y sincro** Define criterios de montaje en el PCB y adaptaciones eléctricas al bus

**Timing** Refresco, tiempo de acceso para lectura y escritura

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

**Pines** No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

**Señalización** Handshake con el hardware controlador

**Signal Integrity** En función de la frecuencia de trabajo

**Encapsulado** Define la manufacturabilidad.

**Clock y sincro** Define criterios de montaje en el PCB y adaptaciones eléctricas al bus

**Timing** Refresco, tiempo de acceso para lectura y escritura

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

**Pines** No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

**Señalización** Handshake con el hardware controlador

**Signal Integrity** En función de la frecuencia de trabajo

**Encapsulado** Define la manufacturabilidad.

**Clock y sincro** Define criterios de montaje en el PCB y adaptaciones eléctricas al bus

**Timing** Refresco, tiempo de acceso para lectura y escritura

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

**Pines** No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

**Señalización** Handshake con el hardware controlador

**Signal Integrity** En función de la frecuencia de trabajo

**Encapsulado** Define la manufacturabilidad.

**Clock y sincro** Define criterios de montaje en el PCB y adaptaciones eléctricas al bus

**Timing** Refresco, tiempo de acceso para lectura y escritura

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Conceptos Básicos

- ▶ A diferencia de las memorias cache que van generalmente en el mismo chip de la CPU, estas memoria al momento se ubican en uno o mas chips separados del de la CPU.
- ▶ Por tal motivo se deben considerar en su diseño los siguientes aspectos:

**Pines** No solo el pinout sino sus características eléctricas (inductancia, capacitancia, etc.)

**Señalización** Handshake con el hardware controlador

**Signal Integrity** En función de la frecuencia de trabajo

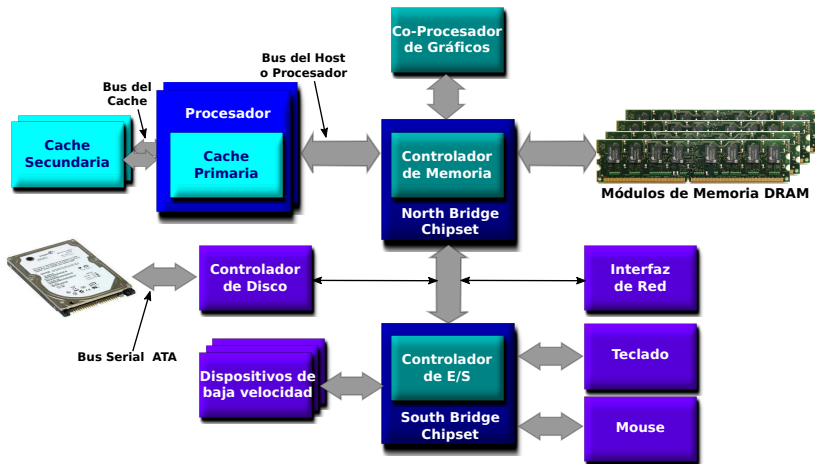
**Encapsulado** Define la manufacturabilidad.

**Clock y sincro** Define criterios de montaje en el PCB y adaptaciones eléctricas al bus

**Timing** Refresco, tiempo de acceso para lectura y escritura

- ▶ Si no se respetan estos aspectos se puede caer en diseños no óptimos o no funcionales.

# Interacción dentro de un computador



# Interacción dentro de un computador

- ▶ Del gráfico anterior puede observarse la presencia de un Controlador de memoria Dinámica.
- ▶ Este dispositivo media entre la CPU y el/los chip/s de DRAM.
- ▶ Su presencia sugiere demoras en el tiempo de acceso (veremos que no es el único factor, pero evidentemente una mediación implica etapas lógicas).
- ▶ Sugiere además una complejidad en el sistema de DRAM que a esta altura no estamos en condiciones de valorar adecuadamente, pero que al cabo es estos slides podremos verificar.
- ▶ Pero también, como veremos, aporta independencia a la CPU de los detalles de la DRAM (timing, refresco, manejo de la estructura interna, por citar los mas obvios)

# Interacción dentro de un computador

- ▶ Del gráfico anterior puede observarse la presencia de un Controlador de memoria Dinámica.
- ▶ Este dispositivo media entre la CPU y el/los chip/s de DRAM.
- ▶ Su presencia sugiere demoras en el tiempo de acceso (veremos que no es el único factor, pero evidentemente una mediación implica etapas lógicas).
- ▶ Sugiere además una complejidad en el sistema de DRAM que a esta altura no estamos en condiciones de valorar adecuadamente, pero que al cabo es estos slides podremos verificar.
- ▶ Pero también, como veremos, aporta independencia a la CPU de los detalles de la DRAM (timing, refresco, manejo de la estructura interna, por citar los mas obvios)



# Interacción dentro de un computador

- ▶ Del gráfico anterior puede observarse la presencia de un Controlador de memoria Dinámica.
- ▶ Este dispositivo media entre la CPU y el/los chip/s de DRAM.
- ▶ Su presencia sugiere demoras en el tiempo de acceso (veremos que no es el único factor, pero evidentemente una mediación implica etapas lógicas).
- ▶ Sugiere además una complejidad en el sistema de DRAM que a esta altura no estamos en condiciones de valorar adecuadamente, pero que al cabo es estos slides podremos verificar.
- ▶ Pero también, como veremos, aporta independencia a la CPU de los detalles de la DRAM (timing, refresco, manejo de la estructura interna, por citar los mas obvios)

# Interacción dentro de un computador

- ▶ Del gráfico anterior puede observarse la presencia de un Controlador de memoria Dinámica.
- ▶ Este dispositivo media entre la CPU y el/los chip/s de DRAM.
- ▶ Su presencia sugiere demoras en el tiempo de acceso (veremos que no es el único factor, pero evidentemente una mediación implica etapas lógicas).
- ▶ Sugiere además una complejidad en el sistema de DRAM que a esta altura no estamos en condiciones de valorar adecuadamente, pero que al cabo es estos slides podremos verificar.
- ▶ Pero también, como veremos, aporta independencia a la CPU de los detalles de la DRAM (timing, refresco, manejo de la estructura interna, por citar los mas obvios)

# Interacción dentro de un computador

- ▶ Del gráfico anterior puede observarse la presencia de un Controlador de memoria Dinámica.
- ▶ Este dispositivo media entre la CPU y el/los chip/s de DRAM.
- ▶ Su presencia sugiere demoras en el tiempo de acceso (veremos que no es el único factor, pero evidentemente una mediación implica etapas lógicas).
- ▶ Sugiere además una complejidad en el sistema de DRAM que a esta altura no estamos en condiciones de valorar adecuadamente, pero que al cabo es estos slides podremos verificar.
- ▶ Pero también, como veremos, aporta independencia a la CPU de los detalles de la DRAM (timing, refresco, manejo de la estructura interna, por citar los mas obvios)

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Organización de una DRAM genérica

- ▶ Es una memoria que implementa cada bit con un par transistor - capacitor.
- ▶ El término dinámica proviene del hecho de que al no implementarse con transistores ideales las corrientes de fuga hacen que se pierda la carga del capacitor.
- ▶ Este hecho hace que deba ser refrescada periódicamente para que no se pierda la información almacenada.

# Organización de una DRAM genérica

- ▶ Es una memoria que implementa cada bit con un par transistor - capacitor.
- ▶ El término dinámica proviene del hecho de que al no implementarse con transistores ideales las corrientes de fuga hacen que se pierda la carga del capacitor.
- ▶ Este hecho hace que deba ser refrescada periódicamente para que no se pierda la información almacenada.

# Organización de una DRAM genérica

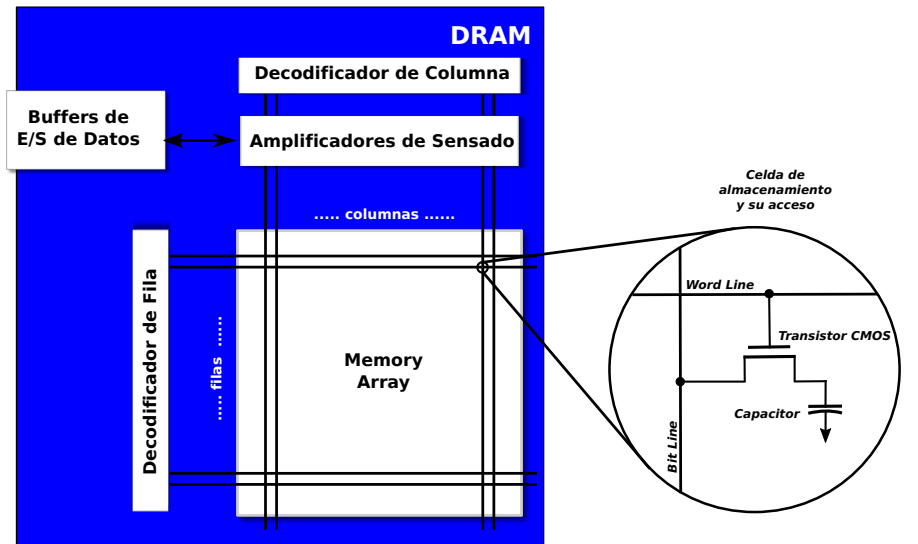
- ▶ Es una memoria que implementa cada bit con un par transistor - capacitor.
- ▶ El término dinámica proviene del hecho de que al no implementarse con transistores ideales las corrientes de fuga hacen que se pierda la carga del capacitor.
- ▶ Este hecho hace que deba ser refrescada periódicamente para que no se pierda la información almacenada.

# Organización de una DRAM genérica

- ▶ Es una memoria que implementa cada bit con un par transistor - capacitor.
- ▶ El término dinámica proviene del hecho de que al no implementarse con transistores ideales las corrientes de fuga hacen que se pierda la carga del capacitor.
- ▶ Este hecho hace que deba ser refrescada periódicamente para que no se pierda la información almacenada.



# Organización de una DRAM genérica



# Organización de una DRAM genérica

- ▶ Cada *die* de DRAM, contiene uno o varios arrays de  $n \times m$  celdas.
- ▶ Cada celda es un bit de almacenamiento (transistor + capacitor)
- ▶ Cada array se organiza en filas (rows) y columnas (cols)
- ▶ Para acceder a cada bit debemos especificar fila y columna dentro del array que lo contiene.
- ▶ Los arrays dentro de cada celda pueden trabajar completamente asociados, completamente disociados, o en configuraciones intermedias entre las opciones anteriores.
- ▶ Si trabajan en forma asociada, todos los arrays transmiten o reciben el bit correspondiente al valor de fila y columna, proveyendo acceso a un número de tantos bits como arrays haya en el die. Por ejemplo una x4 DRAM (se dice por cuatro), entrega un nibble por cada valor de fila y columna que se le provee.

# Organización de una DRAM genérica

- ▶ Cada *die* de DRAM, contiene uno o varios arrays de  $n \times m$  celdas.
- ▶ Cada celda es un bit de almacenamiento (transistor + capacitor)
- ▶ Cada array se organiza en filas (rows) y columnas (cols)
- ▶ Para acceder a cada bit debemos especificar fila y columna dentro del array que lo contiene.
- ▶ Los arrays dentro de cada celda pueden trabajar completamente asociados, completamente disociados, o en configuraciones intermedias entre las opciones anteriores.
- ▶ Si trabajan en forma asociada, todos los arrays transmiten o reciben el bit correspondiente al valor de fila y columna, proveyendo acceso a un número de tantos bits como arrays haya en el die. Por ejemplo una x4 DRAM (se dice por cuatro), entrega un nibble por cada valor de fila y columna que se le provee.

# Organización de una DRAM genérica

- ▶ Cada *die* de DRAM, contiene uno o varios arrays de  $n \times m$  celdas.
- ▶ Cada celda es un bit de almacenamiento (transistor + capacitor)
- ▶ Cada array se organiza en filas (rows) y columnas (cols)
- ▶ Para acceder a cada bit debemos especificar fila y columna dentro del array que lo contiene.
- ▶ Los arrays dentro de cada celda pueden trabajar completamente asociados, completamente disociados, o en configuraciones intermedias entre las opciones anteriores.
- ▶ Si trabajan en forma asociada, todos los arrays transmiten o reciben el bit correspondiente al valor de fila y columna, proveyendo acceso a un número de tantos bits como arrays haya en el die. Por ejemplo una x4 DRAM (se dice por cuatro), entrega un nibble por cada valor de fila y columna que se le provee.

# Organización de una DRAM genérica

- ▶ Cada *die* de DRAM, contiene uno o varios arrays de  $n \times m$  celdas.
- ▶ Cada celda es un bit de almacenamiento (transistor + capacitor)
- ▶ Cada array se organiza en filas (rows) y columnas (cols)
- ▶ Para acceder a cada bit debemos especificar fila y columna dentro del array que lo contiene.
- ▶ Los arrays dentro de cada celda pueden trabajar completamente asociados, completamente disociados, o en configuraciones intermedias entre las opciones anteriores.
- ▶ Si trabajan en forma asociada, todos los arrays transmiten o reciben el bit correspondiente al valor de fila y columna, proveyendo acceso a un número de tantos bits como arrays haya en el die. Por ejemplo una x4 DRAM (se dice por cuatro), entrega un nibble por cada valor de fila y columna que se le provee.

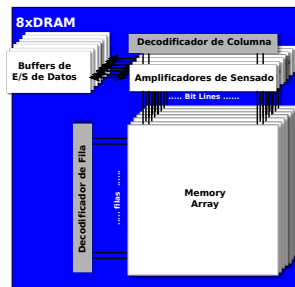
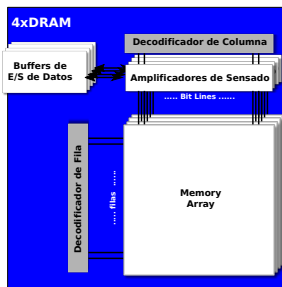
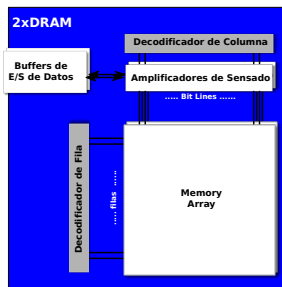
# Organización de una DRAM genérica

- ▶ Cada *die* de DRAM, contiene uno o varios arrays de  $n \times m$  celdas.
- ▶ Cada celda es un bit de almacenamiento (transistor + capacitor)
- ▶ Cada array se organiza en filas (rows) y columnas (cols)
- ▶ Para acceder a cada bit debemos especificar fila y columna dentro del array que lo contiene.
- ▶ Los arrays dentro de cada celda pueden trabajar completamente asociados, completamente disociados, o en configuraciones intermedias entre las opciones anteriores.
- ▶ Si trabajan en forma asociada, todos los arrays transmiten o reciben el bit correspondiente al valor de fila y columna, proveyendo acceso a un número de tantos bits como arrays haya en el die. Por ejemplo una x4 DRAM (se dice por cuatro), entrega un nibble por cada valor de fila y columna que se le provee.

# Organización de una DRAM genérica

- ▶ Cada *die* de DRAM, contiene uno o varios arrays de  $n \times m$  celdas.
- ▶ Cada celda es un bit de almacenamiento (transistor + capacitor)
- ▶ Cada array se organiza en filas (rows) y columnas (cols)
- ▶ Para acceder a cada bit debemos especificar fila y columna dentro del array que lo contiene.
- ▶ Los arrays dentro de cada celda pueden trabajar completamente asociados, completamente disociados, o en configuraciones intermedias entre las opciones anteriores.
- ▶ Si trabajan en forma asociada, todos los arrays transmiten o reciben el bit correspondiente al valor de fila y columna, proveyendo acceso a un número de tantos bits como arrays haya en el die. Por ejemplo una x4 DRAM (se dice por cuatro), entrega un nibble por cada valor de fila y columna que se le provee.

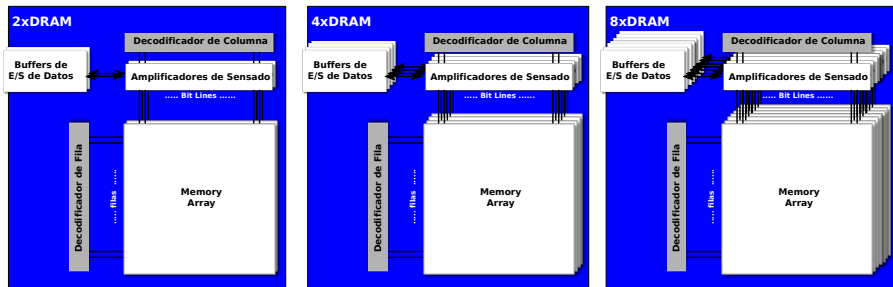
# Organización de una DRAM genérica



- ▶ Las configuraciones habituales para sistemas de media capacidad son x2, x4, y x8.
- ▶ Para las PCs actuales y servidores de mayor capacidad en los 90 comenzaron a producirse x16 y x32
- ▶ Los arrays de la figura anterior conforman bancos que operarán independientemente de otros bancos.

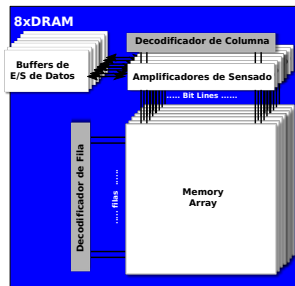
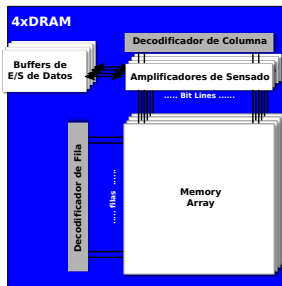
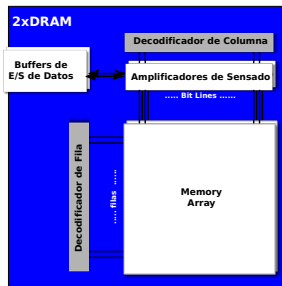


# Organización de una DRAM genérica



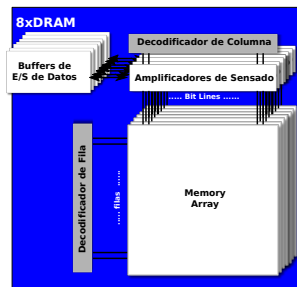
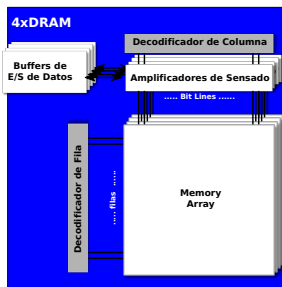
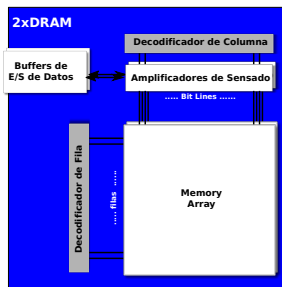
- ▶ Las configuraciones habituales para sistemas de media capacidad son x2, x4, y x8.
- ▶ Para las PCs actuales y servidores de mayor capacidad en los 90 comenzaron a producirse x16 y x32
- ▶ Los arrays de la figura anterior conforman bancos que operarán independientemente de otros bancos.

# Organización de una DRAM genérica



- ▶ Las configuraciones habituales para sistemas de media capacidad son x2, x4, y x8.
- ▶ Para las PCs actuales y servidores de mayor capacidad en los 90 comenzaron a producirse x16 y x32
- ▶ Los arrays de la figura anterior conforman bancos que operarán independientemente de otros bancos.

# Organización de una DRAM genérica



- ▶ Las configuraciones habituales para sistemas de media capacidad son x2, x4, y x8.
- ▶ Para las PCs actuales y servidores de mayor capacidad en los 90 comenzaron a producirse x16 y x32
- ▶ Los arrays de la figura anterior conforman bancos que operarán independientemente de otros bancos.

# Bancos de DRAM

- ▶ Operan en forma independiente como set de arrays con algunas restricciones:
- ▶ Debe ser activados, precargados, leídos, etc. al mismo tiempo que otros bancos, estén todos dentro del mismo dispositivo DRAM o en dispositivos diferentes.
- ▶ El uso de bancos independientes permite aumentar la velocidad de acceso a estos dispositivos (la cual es mas lenta que en el caso de memorias estáticas), mediante el entrelazado de los accesos a bancos diferentes de memoria.
- ▶ Esto se consigue con hardware de mediación entre la memoria y el procesador.
- ▶ Si los bancos tienen un tiempo de acceso e 10 nseg., leer en forma entrelazada dos bancos da un acceso de 5 nseg. Si se hace round robin con 4 bancos, el tiempo de acceso percibido por el procesador es de 2,5 nseg.

# Bancos de DRAM

- ▶ Operan en forma independiente como set de arrays con algunas restricciones:
- ▶ Debe ser activados, precargados, leídos, etc. al mismo tiempo que otros bancos, estén todos dentro del mismo dispositivo DRAM o en dispositivos diferentes.
- ▶ El uso de bancos independientes permite aumentar la velocidad de acceso a estos dispositivos (la cual es mas lenta que en el caso de memorias estáticas), mediante el entrelazado de los accesos a bancos diferentes de memoria.
- ▶ Esto se consigue con hardware de mediación entre la memoria y el procesador.
- ▶ Si los bancos tienen un tiempo de acceso e 10 nseg., leer en forma entrelazada dos bancos da un acceso de 5 nseg. Si se hace round robin con 4 bancos, el tiempo de acceso percibido por el procesador es de 2,5 nseg.

# Bancos de DRAM

- ▶ Operan en forma independiente como set de arrays con algunas restricciones:
- ▶ Debe ser activados, precargados, leídos, etc. al mismo tiempo que otros bancos, estén todos dentro del mismo dispositivo DRAM o en dispositivos diferentes.
- ▶ El uso de bancos independientes permite aumentar la velocidad de acceso a estos dispositivos (la cual es mas lenta que en el caso de memorias estáticas), mediante el entrelazado de los accesos a bancos diferentes de memoria.
- ▶ Esto se consigue con hardware de mediación entre la memoria y el procesador.
- ▶ Si los bancos tienen un tiempo de acceso e 10 nseg., leer en forma entrelazada dos bancos da un acceso de 5 nseg. Si se hace round robin con 4 bancos, el tiempo de acceso percibido por el procesador es de 2,5 nseg.

# Bancos de DRAM

- ▶ Operan en forma independiente como set de arrays con algunas restricciones:
- ▶ Debe ser activados, precargados, leídos, etc. al mismo tiempo que otros bancos, estén todos dentro del mismo dispositivo DRAM o en dispositivos diferentes.
- ▶ El uso de bancos independientes permite aumentar la velocidad de acceso a estos dispositivos (la cual es mas lenta que en el caso de memorias estáticas), mediante el entrelazado de los accesos a bancos diferentes de memoria.
- ▶ Esto se consigue con hardware de mediación entre la memoria y el procesador.
- ▶ Si los bancos tienen un tiempo de acceso e 10 nseg., leer en forma entrelazada dos bancos da un acceso de 5 nseg. Si se hace round robin con 4 bancos, el tiempo de acceso percibido por el procesador es de 2,5 nseg.

# Bancos de DRAM

- ▶ Operan en forma independiente como set de arrays con algunas restricciones:
- ▶ Debe ser activados, precargados, leídos, etc. al mismo tiempo que otros bancos, estén todos dentro del mismo dispositivo DRAM o en dispositivos diferentes.
- ▶ El uso de bancos independientes permite aumentar la velocidad de acceso a estos dispositivos (la cual es mas lenta que en el caso de memorias estáticas), mediante el entrelazado de los accesos a bancos diferentes de memoria.
- ▶ Esto se consigue con hardware de mediación entre la memoria y el procesador.
- ▶ Si los bancos tienen un tiempo de acceso e 10 nseg., leer en forma entrelazada dos bancos da un acceso de 5 nseg. Si se hace round robin con 4 bancos, el tiempo de acceso percibido por el procesador es de 2,5 nseg.



# Dual In Module Memory: DIMM

- ▶ En sistemas que requieren alta capacidad de almacenamiento las DRAM Se organizan en DIMM's.
- ▶ Los dispositivos DRAM se montan a ambos lados del PCB
- ▶ Cada DIMM puede pensarse como un banco independiente.
- ▶ O bien, cada dispositivo o grupo de dispositivos DRAM de un DIMM se puede asignar a un banco independiente.
- ▶ Para diferenciar esta situación se introdujo el concepto de *ran-go* ("rank") para diferenciar la operación independiente a nivel de DIMM de la operación independiente a nivel de banco interno.

# Dual In Module Memory: DIMM

- ▶ En sistemas que requieren alta capacidad de almacenamiento las DRAM Se organizan en DIMM's.
- ▶ Los dispositivos DRAM se montan a ambos lados del PCB
- ▶ Cada DIMM puede pensarse como un banco independiente.
- ▶ O bien, cada dispositivo o grupo de dispositivos DRAM de un DIMM se puede asignar a un banco independiente.
- ▶ Para diferenciar esta situación se introdujo el concepto de *ran-go* ("rank") para diferenciar la operación independiente a nivel de DIMM de la operación independiente a nivel de banco interno.

# Dual In Module Memory: DIMM

- ▶ En sistemas que requieren alta capacidad de almacenamiento las DRAM Se organizan en DIMM's.
- ▶ Los dispositivos DRAM se montan a ambos lados del PCB
- ▶ Cada DIMM puede pensarse como un banco independiente.
- ▶ O bien, cada dispositivo o grupo de dispositivos DRAM de un DIMM se puede asignar a un banco independiente.
- ▶ Para diferenciar esta situación se introdujo el concepto de *ran-go* ("rank") para diferenciar la operación independiente a nivel de DIMM de la operación independiente a nivel de banco interno.

# Dual In Module Memory: DIMM

- ▶ En sistemas que requieren alta capacidad de almacenamiento las DRAM Se organizan en DIMM's.
- ▶ Los dispositivos DRAM se montan a ambos lados del PCB
- ▶ Cada DIMM puede pensarse como un banco independiente.
- ▶ O bien, cada dispositivo o grupo de dispositivos DRAM de un DIMM se puede asignar a un banco independiente.
- ▶ Para diferenciar esta situación se introdujo el concepto de *ran-go* ("rank") para diferenciar la operación independiente a nivel de DIMM de la operación independiente a nivel de banco interno.

# Dual In Module Memory: DIMM

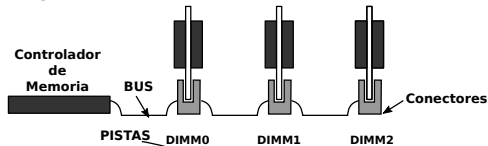
- ▶ En sistemas que requieren alta capacidad de almacenamiento las DRAM Se organizan en DIMM's.
- ▶ Los dispositivos DRAM se montan a ambos lados del PCB
- ▶ Cada DIMM puede pensarse como un banco independiente.
- ▶ O bien, cada dispositivo o grupo de dispositivos DRAM de un DIMM se puede asignar a un banco independiente.
- ▶ Para diferenciar esta situación se introdujo el concepto de *ran-go* ("rank") para diferenciar la operación independiente a nivel de DIMM de la operación independiente a nivel de banco interno.

# Dual In Module Memory: DIMM

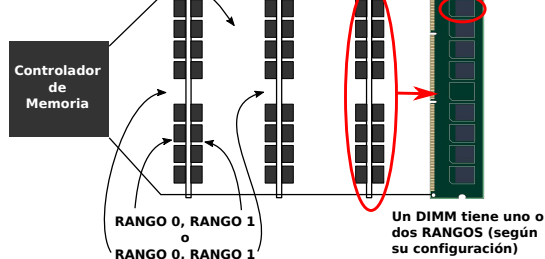
- ▶ En sistemas que requieren alta capacidad de almacenamiento las DRAM Se organizan en DIMM's.
- ▶ Los dispositivos DRAM se montan a ambos lados del PCB
- ▶ Cada DIMM puede pensarse como un banco independiente.
- ▶ O bien, cada dispositivo o grupo de dispositivos DRAM de un DIMM se puede asignar a un banco independiente.
- ▶ Para diferenciar esta situación se introdujo el concepto de *ran-go* ("rank") para diferenciar la operación independiente a nivel de DIMM de la operación independiente a nivel de banco interno.



### VISTA LATERAL



**Un dispositivo DRAM con 8 BANCOS internos.**  
**Los 8 BANCOS se conectan a un bus de E/S compartido**



## Memory Array

## Un BANCO de 4 ARRAYS

Un BANCO de DRAM consta de  $2^n$  ARRAYS dependiendo de las características de la parte. En este ejemplo vemos 4 ARRAYS lo cual indica una parte x4

## Rango

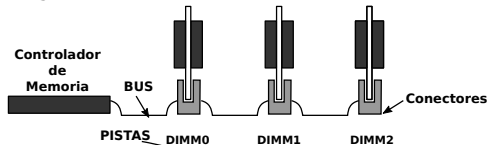
Es un set de dispositivos DRAM (todos los de un DIMM o una parte de ellos), que operan en forma conjunta.



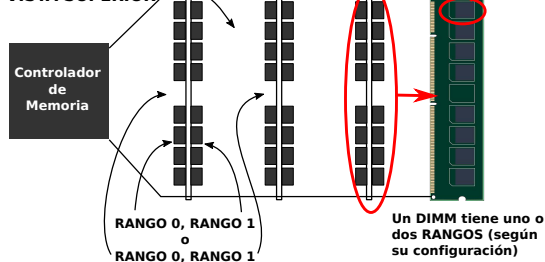


## Organización general de un DIMM

### VISTA LATERAL



## VISTA SUPERIOR



**Un dispositivo DRAM con 8 BANCOS internos.**  
**Los 8 BANCOS se conectan a un bus de E/S compartido**

## Memory Array

## Un BANCO de 4 ARRAYS

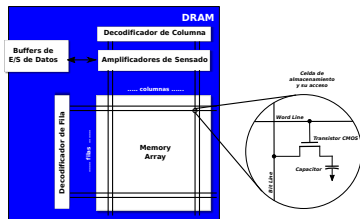
Un BANCO de DRAM consta de  $2^n$  ARRAYS dependiendo de las características de la parte. En este ejemplo vemos 4 ARRAYS lo cual indica una parte x4

# Array

Cada banco de un dispositivo DRAM se compone de un conjunto de arrays esclavos, cuyo número determina el ancho del dispositivo DRAM (x2, x4, etc).

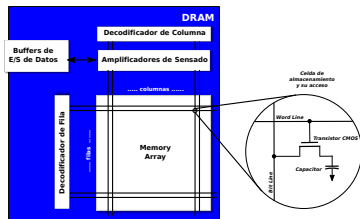


# Dentro del DRAM Device



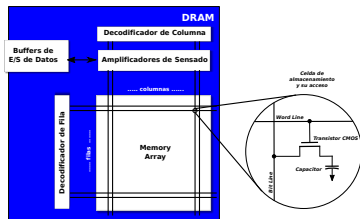
- ▶ Recordemos que cada capacitor se ubica en la intersección entre un wordline (decodificación de la fila) y la bitline (decodificación de la columna).
- ▶ Se conecta al bitline mediante un transistor controlado por la wordline.
- ▶ En los circuitos lógicos un transistor se comporta como un switch.

# Dentro del DRAM Device



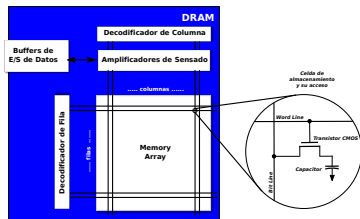
- ▶ Recordemos que cada capacitor se ubica en la intersección entre un wordline (decodificación de la fila) y la bitline (decodificación de la columna).
- ▶ Se conecta al bitline mediante un transistor controlado por la wordline.
- ▶ En los circuitos lógicos un transistor se comporta como un switch.

# Dentro del DRAM Device



- ▶ Recordemos que cada capacitor se ubica en la intersección entre un wordline (decodificación de la fila) y la bitline (decodificación de la columna).
- ▶ Se conecta al bitline mediante un transistor controlado por la wordline.
- ▶ En los circuitos lógicos un transistor se comporta como un switch.

# Dentro del DRAM Device



- ▶ Recordemos que cada capacitor se ubica en la intersección entre un wordline (decodificación de la fila) y la bitline (decodificación de la columna).
- ▶ Se conecta al bitline mediante un transistor controlado por la wordline.
- ▶ En los circuitos lógicos un transistor se comporta como un switch.

Por eso cuando la tensión eléctrica en el wordline es alta, todos los transistores cuyo gate está conectado a esa wordline se ponen en saturación y se asimila a un switch cerrado conectando al capacitor al bitline.

# Dentro del DRAM Device

- ▶ En las escalas actuales de integración la cantidad de electrones que puede almacenar el capacitor es minúscula comparada con las características físicas del bitline.
- ▶ Por lo tanto se requiere un conjunto de amplificadores de sensado, capaces de detectar los valores eléctricos almacenados en los capacitores cuyas bitlines han sido habilitadas por la wordline decodificada a partir del valor de fila (row) enviado por el controlador de DRAM.

# Dentro del DRAM Device

- ▶ En las escalas actuales de integración la cantidad de electrones que puede almacenar el capacitor es minúscula comparada con las características físicas del bitline.
- ▶ Por lo tanto se requiere un conjunto de amplificadores de sensado. capaces de detectar los valores eléctricos almacenados en los capacitores cuyas bitlines han sido habilitadas por la wordline decodificada a partir del valor de fila (row) enviado por el controlador de DRAM.



# Dentro del DRAM Device

- ▶ El amplificador de sensado, en principio, *precarga* las bitlines con un valor eléctrico intermedio entre '0' y '1'.
- ▶ Al habilitarse una wordline, el capacitor de la bitline habilitada produce una alteración muy pequeña en el valor de tensión precargado, en mas o en menos dependiendo del estado de su carga.
- ▶ Esa diferencia es muy pequeña, pero el amplificador de sensado está diseñado para detectarla.
- ▶ En función del sentido de la variación de tensión detectada el amplificador lleva la bitline a un '0' o a un '1'.
- ▶ Este pull a '0' o a '1' por parte del amplificador de sensado, por otra parte restituye el valor de carga de los capacitores cuya carga se había perdido al modificar el nivel de tensión precargada en cada bitline.

# Dentro del DRAM Device

- ▶ El amplificador de sensado, en principio, *precarga* las bitlines con un valor eléctrico intermedio entre '0' y '1'.
- ▶ Al habilitarse una wordline, el capacitor de la bitline habilitada produce una alteración muy pequeña en el valor de tensión precargado, en mas o en menos dependiendo del estado de su carga.
- ▶ Esa diferencia es muy pequeña, pero el amplificador de sensado está diseñado para detectarla.
- ▶ En función del sentido de la variación de tensión detectada el amplificador lleva la bitline a un '0' o a un '1'.
- ▶ Este pull a '0' o a '1' por parte del amplificador de sensado, por otra parte restituye el valor de carga de los capacitores cuya carga se había perdido al modificar el nivel de tensión precargada en cada bitline.

# Dentro del DRAM Device

- ▶ El amplificador de sensado, en principio, *precarga* las bitlines con un valor eléctrico intermedio entre '0' y '1'.
- ▶ Al habilitarse una wordline, el capacitor de la bitline habilitada produce una alteración muy pequeña en el valor de tensión precargado, en mas o en menos dependiendo del estado de su carga.
- ▶ Esa diferencia es muy pequeña, pero el amplificador de sensado está diseñado para detectarla.
- ▶ En función del sentido de la variación de tensión detectada el amplificador lleva la bitline a un '0' o a un '1'.
- ▶ Este pull a '0' o a '1' por parte del amplificador de sensado, por otra parte restituye el valor de carga de los capacitores cuya carga se había perdido al modificar el nivel de tensión precargada en cada bitline.

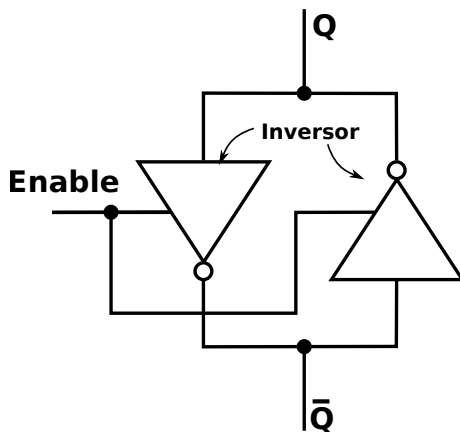
# Dentro del DRAM Device

- ▶ El amplificador de sensado, en principio, *precarga* las bitlines con un valor eléctrico intermedio entre '0' y '1'.
- ▶ Al habilitarse una wordline, el capacitor de la bitline habilitada produce una alteración muy pequeña en el valor de tensión precargado, en mas o en menos dependiendo del estado de su carga.
- ▶ Esa diferencia es muy pequeña, pero el amplificador de sensado está diseñado para detectarla.
- ▶ En función del sentido de la variación de tensión detectada el amplificador lleva la bitline a un '0' o a un '1'.
- ▶ Este pull a '0' o a '1' por parte del amplificador de sensado, por otra parte restituye el valor de carga de los capacitores cuya carga se había perdido al modificar el nivel de tensión precargada en cada bitline.

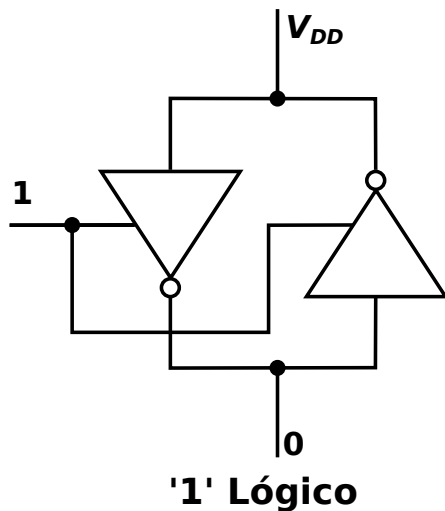
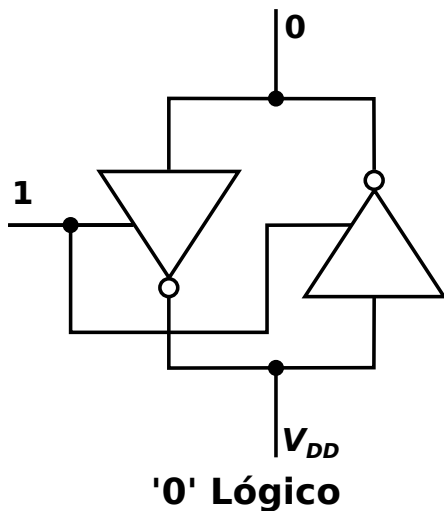
# Dentro del DRAM Device

- ▶ El amplificador de sensado, en principio, *precarga* las bitlines con un valor eléctrico intermedio entre '0' y '1'.
- ▶ Al habilitarse una wordline, el capacitor de la bitline habilitada produce una alteración muy pequeña en el valor de tensión precargado, en mas o en menos dependiendo del estado de su carga.
- ▶ Esa diferencia es muy pequeña, pero el amplificador de sensado está diseñado para detectarla.
- ▶ En función del sentido de la variación de tensión detectada el amplificador lleva la bitline a un '0' o a un '1'.
- ▶ Este pull a '0' o a '1' por parte del amplificador de sensado, por otra parte restituye el valor de carga de los capacitores cuya carga se había perdido al modificar el nivel de tensión precargada en cada bitline.

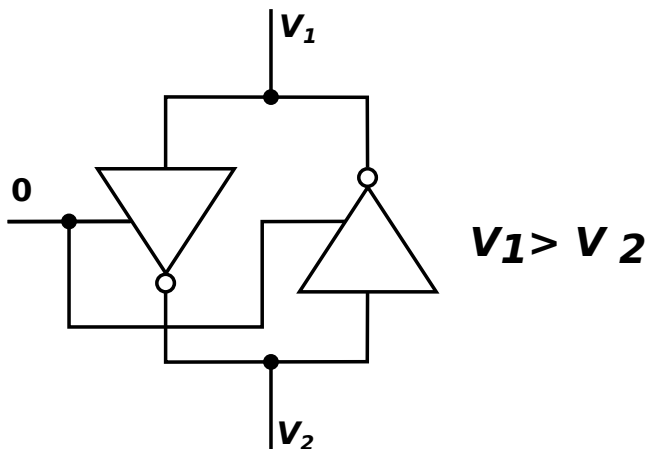
# El amplificador de sentido



# Estados estables del amplificador de sensado

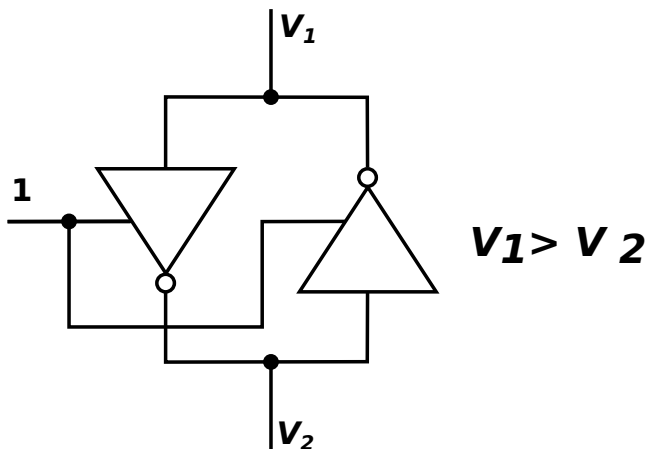


# Operación del amplificador de sensado

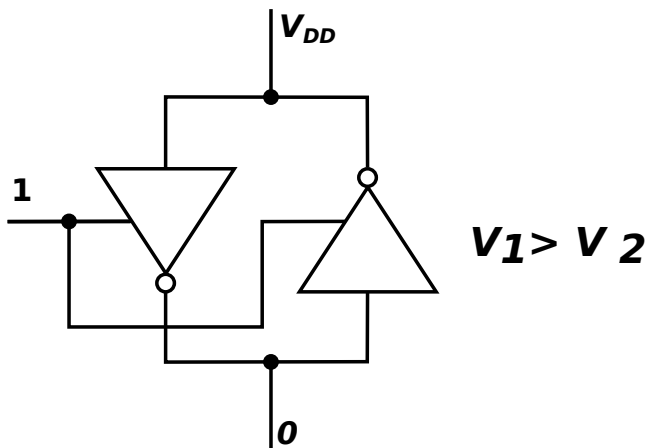




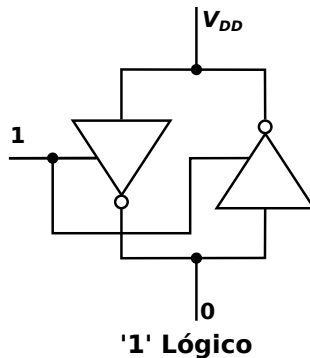
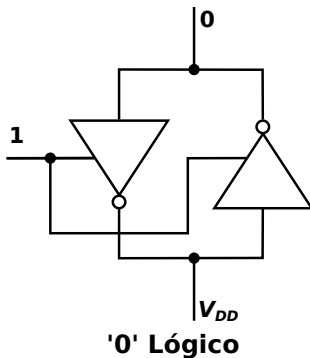
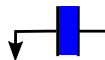
# Operación del amplificador de sensado



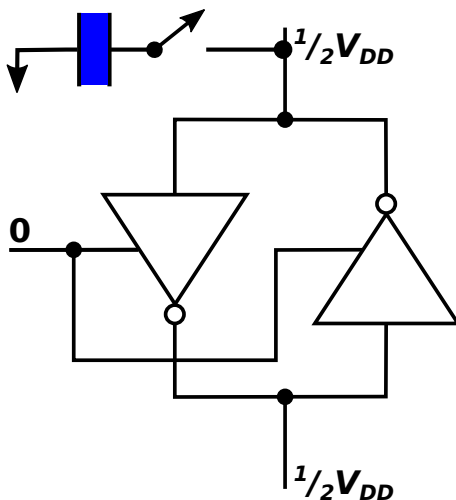
# Operación del amplificador de sensado



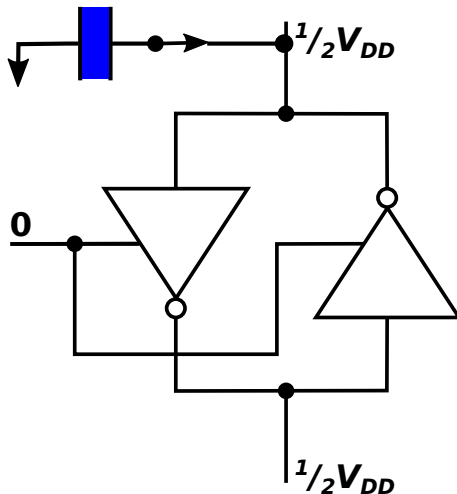
# El capacitor mas el amplificador de sensado



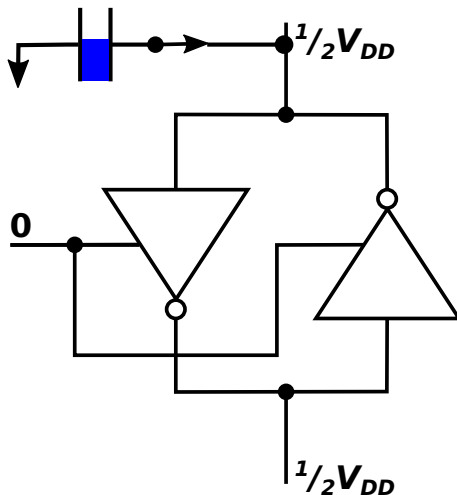
# Operación de la celda



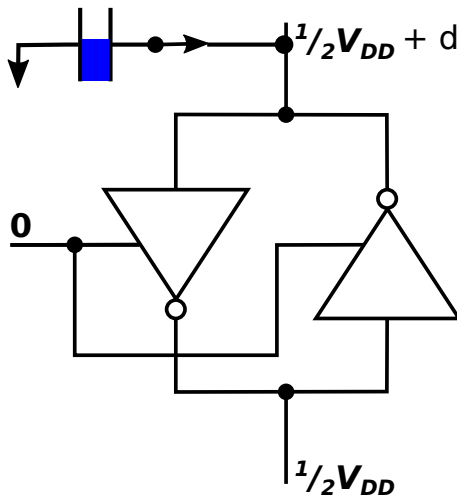
# Operación de la celda



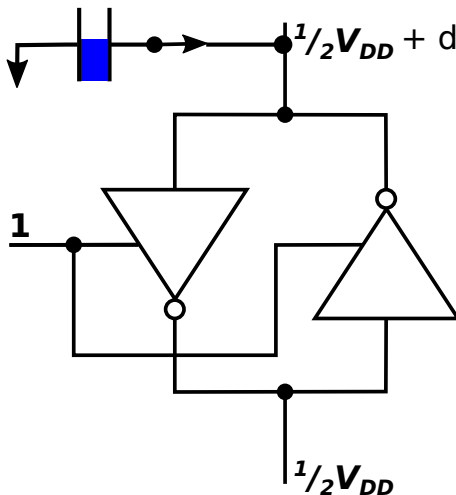
# Operación de la celda



## Operación de la celda

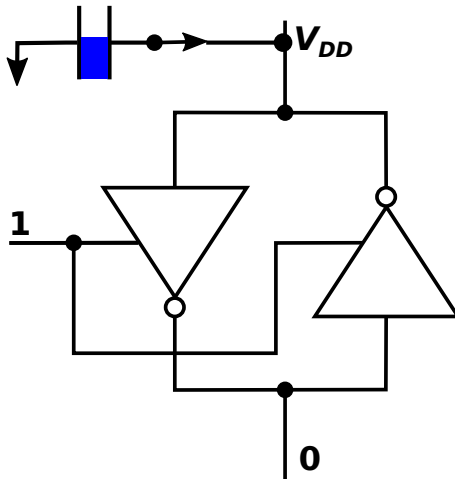


# Operación de la celda

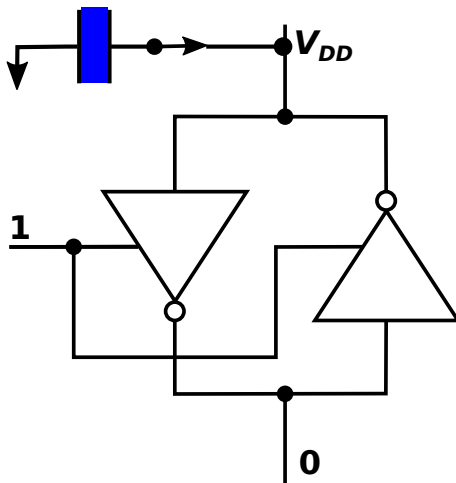




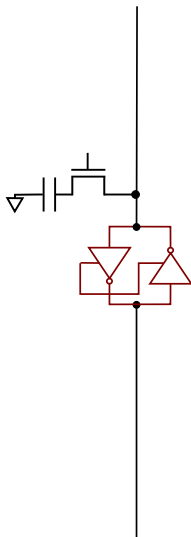
# Operación de la celda



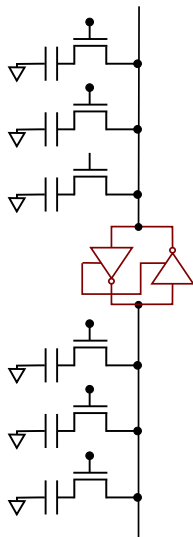
# Operación de la celda



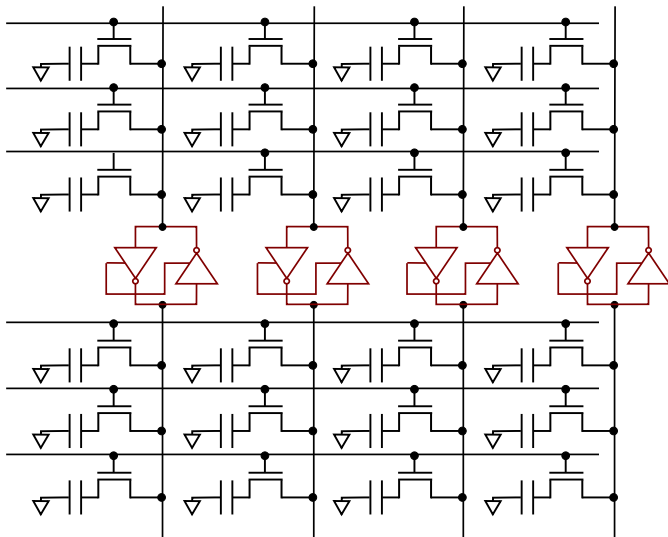
## Operación de la celda



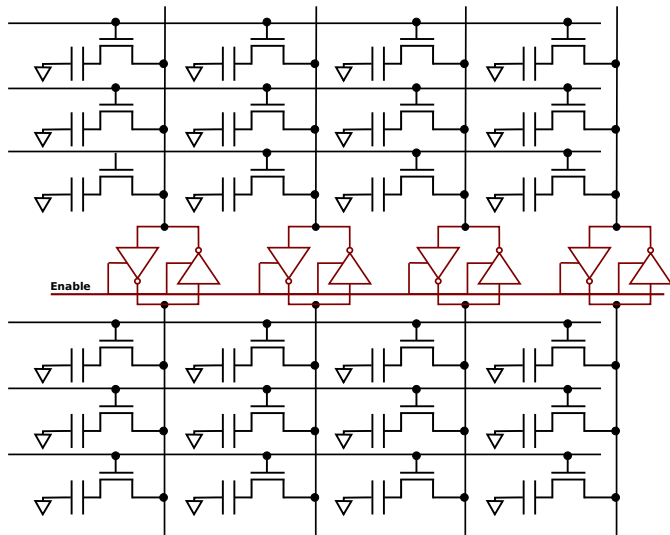
# Operación de la celda



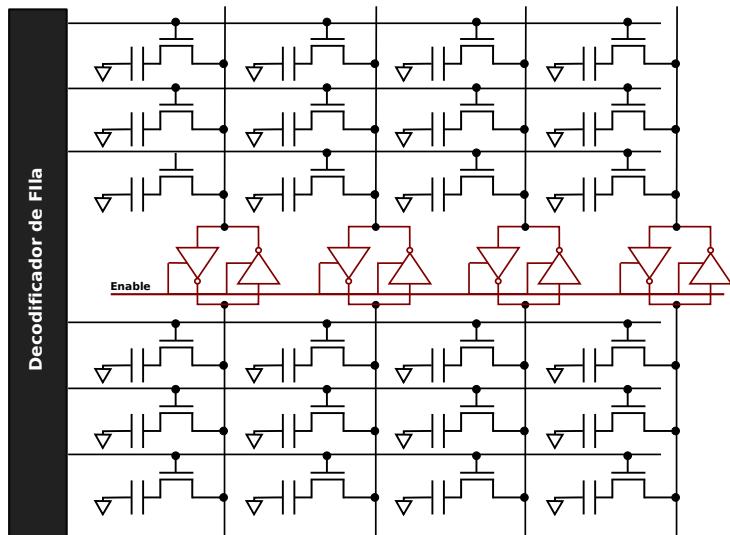
# Operación de la celda



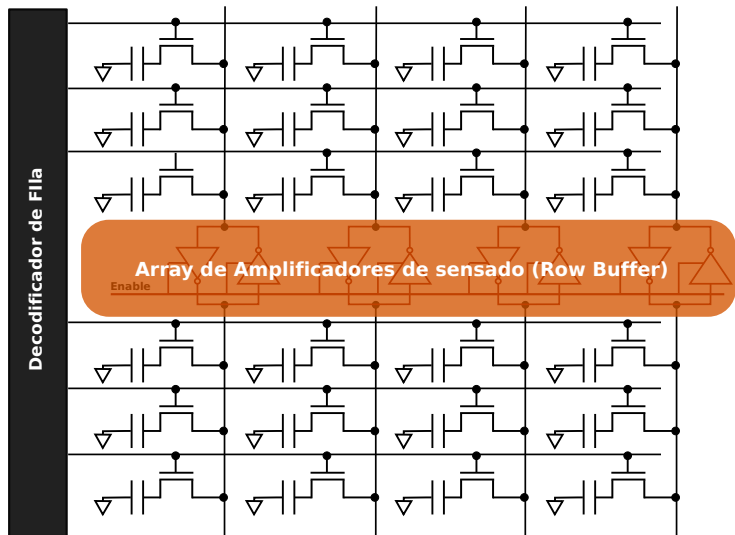
# Operación de la celda



# Operación de la celda



# Operación de la celda

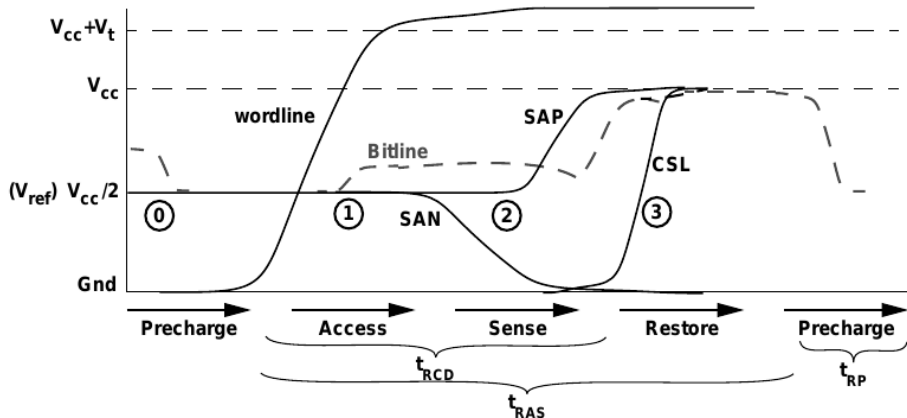




# Operación de la celda



# Operación del amplificador de sensado



# Acceso

- ▶ El controlador de memoria a partir de la dirección física recibida, decodifica el rango de memoria al que se debe dirigir el requerimiento, el banco dentro de ese rango y los valores de fila y columna dentro del banco.
- ▶ Con los valores de fila y columna se componen la *row address* y la *column address*.
- ▶ Una vez determinado el banco se genera un grupo de bits adicionales a la dirección.
- ▶ Con la determinación del rango el controlador genera el valor a colocar en el bus de chip select .
- ▶ Al recibir la información de direcciones el DRAM device precarga (es decir, pullea mediante sus amplificadores de sensado a un valor intermedio entre '0' y '1') las bitlines del banco seleccionado.

# Acceso

- ▶ El controlador de memoria a partir de la dirección física recibida, decodifica el rango de memoria al que se debe dirigir el requerimiento, el banco dentro de ese rango y los valores de fila y columna dentro del banco.
- ▶ Con los valores de fila y columna se componen la **row address** y la **column address**.
- ▶ Una vez determinado el banco se genera un grupo de bits adicionales a la dirección.
- ▶ Con la determinación del rango el controlador genera el valor a colocar en el bus de chip select .
- ▶ Al recibir la información de direcciones el DRAM device precarga (es decir, pullea mediante sus amplificadores de sensado a un valor intermedio entre '0' y '1') las bitlines del banco seleccionado.

# Acceso

- ▶ El controlador de memoria a partir de la dirección física recibida, decodifica el rango de memoria al que se debe dirigir el requerimiento, el banco dentro de ese rango y los valores de fila y columna dentro del banco.
- ▶ Con los valores de fila y columna se componen la **row address** y la **column address**.
- ▶ Una vez determinado el banco se genera un grupo de bits adicionales a la dirección.
- ▶ Con la determinación del rango el controlador genera el valor a colocar en el bus de chip select .
- ▶ Al recibir la información de direcciones el DRAM device precarga (es decir, pullea mediante sus amplificadores de sensado a un valor intermedio entre '0' y '1') las bitlines del banco seleccionado.

# Acceso

- ▶ El controlador de memoria a partir de la dirección física recibida, decodifica el rango de memoria al que se debe dirigir el requerimiento, el banco dentro de ese rango y los valores de fila y columna dentro del banco.
- ▶ Con los valores de fila y columna se componen la **row address** y la **column address**.
- ▶ Una vez determinado el banco se genera un grupo de bits adicionales a la dirección.
- ▶ Con la determinación del rango el controlador genera el valor a colocar en el bus de chip select .
- ▶ Al recibir la información de direcciones el DRAM device precarga (es decir, pullea mediante sus amplificadores de sensado a un valor intermedio entre '0' y '1') las bitlines del banco seleccionado.

# Acceso

- ▶ El controlador de memoria a partir de la dirección física recibida, decodifica el rango de memoria al que se debe dirigir el requerimiento, el banco dentro de ese rango y los valores de fila y columna dentro del banco.
- ▶ Con los valores de fila y columna se componen la **row address** y la **column address**.
- ▶ Una vez determinado el banco se genera un grupo de bits adicionales a la dirección.
- ▶ Con la determinación del rango el controlador genera el valor a colocar en el bus de chip select .
- ▶ Al recibir la información de direcciones el DRAM device precarga (es decir, pullea mediante sus amplificadores de sensado a un valor intermedio entre '0' y '1') las bitlines del banco seleccionado.

# Acceso

- ▶ El DRAM Device selecciona la fila entera (cientos de celdas)
- ▶ Cada fila ataca a su correspondiente amplificador de sensado dentro del array de amplificadores de sensado,
- ▶ Cada amplificador detecta las diferencias de tensión aplicada por el capacitor conectado a su bitline a través del transistor, y fuerza (“pull”) el valor lógico definitivo. Esto lleva en las tecnologías actuales típicamente decenas de nanosegundos.
- ▶ Luego de esto se activa la señal en el pin CAS (Column Address Strobe, para activar la wordline deseada del banco seleccionado, lo cual selecciona a los amplificadores de sensado que deben ser conectados al buffer de salida para finalmente ser derivados al bus de datos.



# Acceso

- ▶ El DRAM Device selecciona la fila entera (cientos de celdas)
- ▶ Cada fila ataca a su correspondiente amplificador de sensado dentro del array de amplificadores de sensado,
- ▶ Cada amplificador detecta las diferencias de tensión aplicada por el capacitor conectado a su bitline a través del transistor, y fuerza ("pull") el valor lógico definitivo. Esto lleva en las tecnologías actuales típicamente decenas de nanosegundos.
- ▶ Luego de esto se activa la señal en el pin CAS (Column Address Strobe, para activar la wordline deseada del banco seleccionado, lo cual selecciona a los amplificadores de sensado que deben ser conectados al buffer de salida para finalmente ser derivados al bus de datos.

# Acceso

- ▶ El DRAM Device selecciona la fila entera (cientos de celdas)
- ▶ Cada fila ataca a su correspondiente amplificador de sensado dentro del array de amplificadores de sensado,
- ▶ Cada amplificador detecta las diferencias de tensión aplicada por el capacitor conectado a su bitline a través del transistor, y fuerza (“pull”) el valor lógico definitivo. Esto lleva en las tecnologías actuales típicamente decenas de nanosegundos.
- ▶ Luego de esto se activa la señal en el pin CAS (Column Address Strobe, para activar la wordline deseada del banco seleccionado, lo cual selecciona a los amplificadores de sensado que deben ser conectados al buffer de salida para finalmente ser derivados al bus de datos.

# Acceso

- ▶ El DRAM Device selecciona la fila entera (cientos de celdas)
- ▶ Cada fila ataca a su correspondiente amplificador de sensado dentro del array de amplificadores de sensado,
- ▶ Cada amplificador detecta las diferencias de tensión aplicada por el capacitor conectado a su bitline a través del transistor, y fuerza (“pull”) el valor lógico definitivo. Esto lleva en las tecnologías actuales típicamente decenas de nanosegundos.
- ▶ Luego de esto se activa la señal en el pin CAS (Column Address Strobe, para activar la wordline deseada del banco seleccionado, lo cual selecciona a los amplificadores de sensado que deben ser conectados al buffer de salida para finalmente ser derivados al bus de datos.

# Acceso

- ▶ Cuando el controlador de memoria recibe los datos los retransmite al procesador.
- ▶ Una vez precargada la bitline se debe activar la fila correspondiente en el banco seleccionado del rango seleccionado.
- ▶ Para ello se activa el chip select que selecciona los DRAM devices correspondientes de acuerdo con la información puesta en el bus, y se activa la fila seleccionada con la **row address** activando el terminal RAS (Row Address Strobe) del DRAM Device.

# Acceso

- ▶ Cuando el controlador de memoria recibe los datos los retransmite al procesador.
- ▶ Una vez precargada la bitline se debe activar la fila correspondiente en el banco seleccionado del rango seleccionado.
- ▶ Para ello se activa el chip select que selecciona los DRAM devices correspondientes de acuerdo con la información puesta en el bus, y se activa la fila seleccionada con la **row address** activando el terminal RAS (Row Address Strobe) del DRAM Device.

# Acceso

- ▶ Cuando el controlador de memoria recibe los datos los retransmite al procesador.
- ▶ Una vez precargada la bitline se debe activar la fila correspondiente en el banco seleccionado del rango seleccionado.
- ▶ Para ello se activa el chip select que selecciona los DRAM devices correspondientes de acuerdo con la información puesta en el bus, y se activa la fila seleccionada con la **row address** activando el terminal RAS (Row Address Strobe) del DRAM Device.

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# DRAMs: un cuello de botella para la performance

- ▶ En la segunda mitad de los 80' la velocidad de los microprocesadores comenzó a despegarse de la velocidad de las memorias y hacia inicios de los 90' éstas se transforman en un cuello de botella.
- ▶ Comienza la etapa de interfaz procesador-memoria, además de los caches de SRAM.
- ▶ Las interfaces evolucionan, pero la arquitectura interna de los DRAM Devices permanece inalterada.
- ▶ Se plantearon dos caminos respecto del uso o no del clock.
- ▶ En las DRAM asincrónicas no hay una señal de clock del sistema conectada al DRAM device, sino que es el controlador de memoria quien determina mediante handshake a través del bus las acciones que deben desarrollarse dentro del DRAM device.
- ▶ Las DRAM sincrónicas trabajan de acuerdo a la señal de clock del sistema, utilizando uno de sus flancos o ambos flancos.



# DRAMs: un cuello de botella para la performance

- ▶ En la segunda mitad de los 80' la velocidad de los microprocesadores comenzó a despegarse de la velocidad de las memorias y hacia inicios de los 90' éstas se transforman en un cuello de botella.
- ▶ Comienza la etapa de interfaz procesador-memoria, además de los caches de SRAM.
- ▶ Las interfaces evolucionan, pero la arquitectura interna de los DRAM Devices permanece inalterada.
- ▶ Se plantearon dos caminos respecto del uso o no del clock.
- ▶ En las DRAM asincrónicas no hay una señal de clock del sistema conectada al DRAM device, sino que es el controlador de memoria quien determina mediante handshake a través del bus las acciones que deben desarrollarse dentro del DRAM device.
- ▶ Las DRAM sincrónicas trabajan de acuerdo a la señal de clock del sistema, utilizando uno de sus flancos o ambos flancos.

# DRAMs: un cuello de botella para la performance

- ▶ En la segunda mitad de los 80' la velocidad de los microprocesadores comenzó a despegarse de la velocidad de las memorias y hacia inicios de los 90' éstas se transforman en un cuello de botella.
- ▶ Comienza la etapa de interfaz procesador-memoria, además de los caches de SRAM.
- ▶ Las interfaces evolucionan, pero la arquitectura interna de los DRAM Devices permanece inalterada.
- ▶ Se plantearon dos caminos respecto del uso o no del clock.
- ▶ En las DRAM asincrónicas no hay una señal de clock del sistema conectada al DRAM device, sino que es el controlador de memoria quien determina mediante handshake a través del bus las acciones que deben desarrollarse dentro del DRAM device.
- ▶ Las DRAM sincrónicas trabajan de acuerdo a la señal de clock del sistema, utilizando uno de sus flancos o ambos flancos.

# DRAMs: un cuello de botella para la performance

- ▶ En la segunda mitad de los 80' la velocidad de los microprocesadores comenzó a despegarse de la velocidad de las memorias y hacia inicios de los 90' éstas se transforman en un cuello de botella.
- ▶ Comienza la etapa de interfaz procesador-memoria, además de los caches de SRAM.
- ▶ Las interfaces evolucionan, pero la arquitectura interna de los DRAM Devices permanece inalterada.
- ▶ Se plantearon dos caminos respecto del uso o no del clock.
- ▶ En las DRAM asincrónicas no hay una señal de clock del sistema conectada al DRAM device, sino que es el controlador de memoria quien determina mediante handshake a través del bus las acciones que deben desarrollarse dentro del DRAM device.
- ▶ Las DRAM sincrónicas trabajan de acuerdo a la señal de clock del sistema, utilizando uno de sus flancos o ambos flancos.

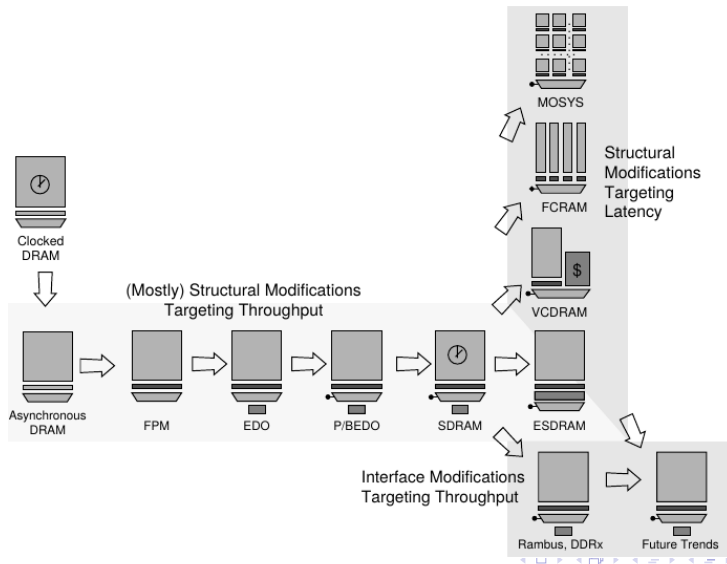
# DRAMs: un cuello de botella para la performance

- ▶ En la segunda mitad de los 80' la velocidad de los microprocesadores comenzó a despegarse de la velocidad de las memorias y hacia inicios de los 90' éstas se transforman en un cuello de botella.
- ▶ Comienza la etapa de interfaz procesador-memoria, además de los caches de SRAM.
- ▶ Las interfaces evolucionan, pero la arquitectura interna de los DRAM Devices permanece inalterada.
- ▶ Se plantearon dos caminos respecto del uso o no del clock.
- ▶ En las DRAM asincrónicas no hay una señal de clock del sistema conectada al DRAM device, sino que es el controlador de memoria quien determina mediante handshake a través del bus las acciones que deben desarrollarse dentro del DRAM device.
- ▶ Las DRAM sincrónicas trabajan de acuerdo a la señal de clock del sistema, utilizando uno de sus flancos o ambos flancos.

# DRAMs: un cuello de botella para la performance

- ▶ En la segunda mitad de los 80' la velocidad de los microprocesadores comenzó a despegarse de la velocidad de las memorias y hacia inicios de los 90' éstas se transforman en un cuello de botella.
- ▶ Comienza la etapa de interfaz procesador-memoria, además de los caches de SRAM.
- ▶ Las interfaces evolucionan, pero la arquitectura interna de los DRAM Devices permanece inalterada.
- ▶ Se plantearon dos caminos respecto del uso o no del clock.
- ▶ En las DRAM asincrónicas no hay una señal de clock del sistema conectada al DRAM device, sino que es el controlador de memoria quien determina mediante handshake a través del bus las acciones que deben desarrollarse dentro del DRAM device.
- ▶ Las DRAM sincrónicas trabajan de acuerdo a la señal de clock del sistema, utilizando uno de sus flancos o ambos flancos.

# Evolución de las tecnologías DRAM



# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Tecnologías orientadas al Throughput

- ▶ **Clocked DRAM:** DRAM original que se utilizó en los 60's y 70's.
- ▶ **DRAM Asíncrona:** Se introducen en los 70's. Es la que estamos habituados a usar en pequeños sistemas. Responde al mecanismo de acceso descripto, pero manejada directo por el procesador con lógica externa. CAS y RAS se debían aplicar en forma simultánea.
- ▶ **FPM (Fast Page Mode):** Es la primer evolución que bufferea la información de fila, de modo de leer sucesivas columnas. Implementa un cache de facto en los amplificadores de sensado. Esto acelera los accesos sucesivos a la misma fila, cambiando solo CAS mientras que RAS permanece activa. La lectura de columnas sucesivas comienza a considerarse en base al principio de vecindad.



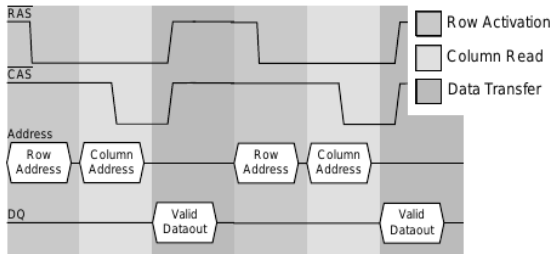
# Tecnologías orientadas al Throughput

- ▶ **Clocked DRAM:** DRAM original que se utilizó en los 60's y 70's.
- ▶ **DRAM Asincrónica:** Se introducen en los 70's. Es la que estamos habituados a usar en pequeños sistemas. Responde al mecanismo de acceso descripto, pero manejada directo por el procesador con lógica externa. CAS y RAS se debían aplicar en forma simultánea.
- ▶ **FPM (Fast Page Mode):** Es la primer evolución que bufferea la información de fila, de modo de leer sucesivas columnas. Implementa un cache de facto en los amplificadores de sensado. Esto acelera los accesos sucesivos a la misma fila, cambiando solo CAS mientras que RAS permanece activa. La lectura de columnas sucesivas comienza a considerarse en base al principio de vecindad.

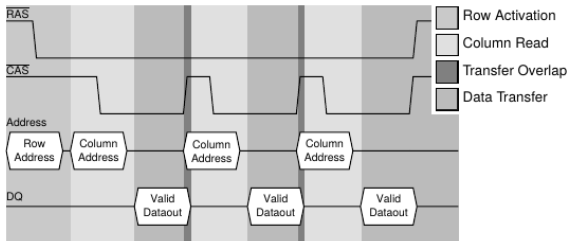
# Tecnologías orientadas al Throughput

- ▶ **Clocked DRAM:** DRAM original que se utilizó en los 60's y 70's.
- ▶ **DRAM Asincrónica:** Se introducen en los 70's. Es la que estamos habituados a usar en pequeños sistemas. Responde al mecanismo de acceso descripto, pero manejada directo por el procesador con lógica externa. CAS y RAS se debían aplicar en forma simultánea.
- ▶ **FPM (Fast Page Mode):** Es la primer evolución que bufferea la información de fila, de modo de leer sucesivas columnas. Implementa un cache de facto en los amplificadores de sensado. Esto acelera los accesos sucesivos a la misma fila, cambiando solo CAS mientras que RAS permanece activa. La lectura de columnas sucesivas comienza a considerarse en base al principio de vecindad.

# Tecnologías orientadas al Throughput



## Timing DRAM Asíncrona



## Timing DRAM Fast Page Mode

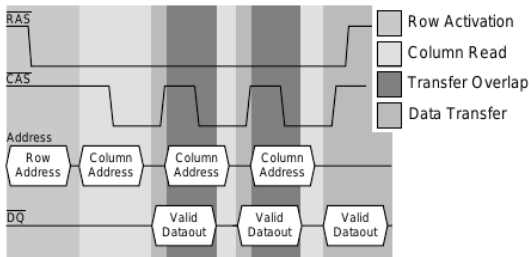
# Tecnologías orientadas al Throughput

- ▶ **EDO DRAM (Extended Data Out)**: Agregando unos pocos transistores al driver de salida de un FPM, implementa un latch entre los amplificadores de sensado y los pines de salida de datos. Así mantiene la salida permitiendo retirar CAS mas rápidamente, y que el array de memoria se precargue antes. Además el latch naturalmente extiende el tiempo de validez de la salida (EDO). Mejora en un 15 % la velocidad de acceso respecto de FPM
- ▶ **BEDO DRAM (Bursted EDO)**: Fue rápidamente reemplazada por la siguiente generación pero marca una nueva generación de memorias. Incluye un contador que incrementa el número de columna con la pulsación de la señal CAS. De este modo permite leer ráfagas (burst) columnas consecutivas sin necesidad de colocar la dirección de columna. Nuevamente pone de relevancia la validez del principio de vecindad.

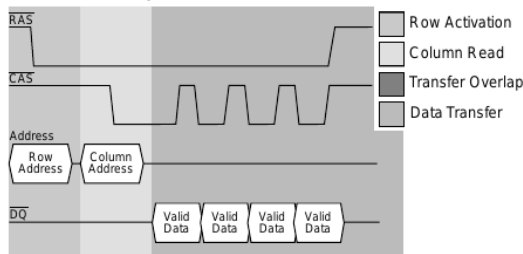
# Tecnologías orientadas al Throughput

- ▶ **EDO DRAM (Extended Data Out)**: Agregando unos pocos transistores al driver de salida de un FPM, implementa un latch entre los amplificadores de sensado y los pines de salida de datos. Así mantiene la salida permitiendo retirar CAS mas rápidamente, y que el array de memoria se precargue antes. Además el latch naturalmente extiende el tiempo de validez de la salida (EDO). Mejora en un 15 % la velocidad de acceso respecto de FPM
- ▶ **BEDO DRAM (Bursted EDO)**: Fue rápidamente reemplazada por la siguiente generación pero marca una nueva generación de memorias. Incluye un contador que incrementa el número de columna con la pulsación de la señal CAS. De este modo permite leer ráfagas (burst) columnas consecutivas sin necesidad de colocar la dirección de columna. Nuevamente pone de relevancia la validez del principio de vecindad.

# Tecnologías orientadas al Throughput



## Timing DRAM Extended Data Out



## Timing DRAM Bursted EDO

# Tecnologías orientadas al Throughput

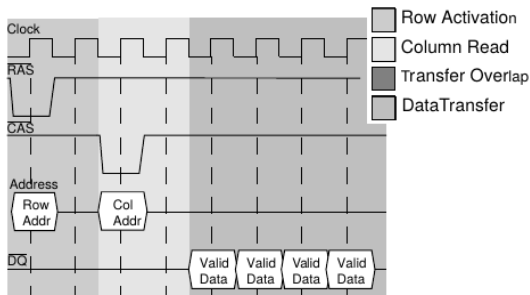
- ▶ **IBM High Speed Toggle Mode DRAM:** A fines de los 80's IBM diseñó y fabricó una interfaz de memoria de alta velocidad. La presentó en la International Solid Circuit Conference en Febrero de 1990, y al JEDEC en Setiembre del 90. Utilizaba ambos flancos de la señal de strobe. Como la frecuencia de esta señal era sumamente alta para ese momento, se conmutaba (Toggle) entre dos buffers de salida por cada flanco de strobe.

# Tecnologías orientadas al Throughput

- ▶ **SDRAM Synchronous DRAM:** Todas las generaciones anteriores eran asincrónicas. Sincronizar hace posible aplicar CAS y RAS al mismo tiempo, ya que se dispone de latches internos que aseguran la información que se aplica a la matriz. Esto deriva en que el tiempo de acceso es mas predecible, por independizar el acceso cada dispositivo DRAM del resto (de-skewing), y minimiza el tiempo de conmutación de un DIMM al próximo (cuando se tienen varios DIMMs). Conserva el burst mode introducido por BEDO. Los SDRAM devices tienen un registro para programar el números de accesos burst. Esto evita enviar un CAS por cada lectura reduciendo el ancho de banda de bus dedicado a handshake aumentando el porcentaje relativo dedicado a transferir datos.



# Tecnologías orientadas al Throughput

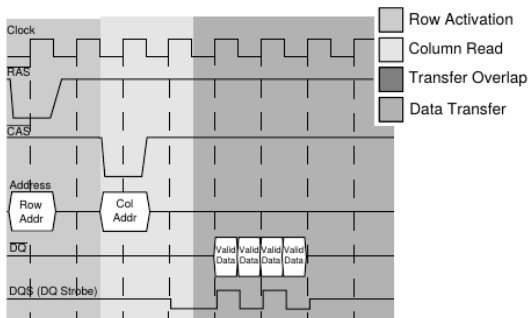


Timing DRAM Sincrónica (SDRAM)

# Interfaces orientadas al Throughput

- ▶ **DDR-SDRAM Double Data Rate SDRAM:** Es el equivalente moderno al IBM Toggle Mode, en cuanto a que se utilizan ambos flancos de la señal para transferir datos. En este caso al ser una memoria Sincrónica, se utilizan ambos flancos de la señal de clock. En las restantes características es similar a la SDRAM: utilizan la misma tecnología de señalización, la misma especificación de interfaz, y los mismos pinouts en los DIMM carriers. Sin embargo tienen el doble de velocidad de transferencia que las SDRAM. Durante las escrituras se prescinde de la señal de clock y se utiliza una señal DQS en cuyos flancos ascendente y descendente se escriben los datos. Esto contradice lo estipulado por JEDEC para SDRAMs y hace que las DDR se asimilen al standard IBM Toggle.

# Interfaces orientadas al Throughput



Timing Double Data Rate SDRAM

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# JEDEC

- ▶ Las DRAM actualmente son un commodity.
- ▶ Cualquier DRAM o DIMM actual tiene especificaciones equivalentes: ancho de bus, capacidad, velocidad, interfaz, por citar las mas importantes.
- ▶ Esta compatibilidad está gobernada por JEDEC (**J**oint **E**lectron **D**evice **E**ngineering **C**ouncil).
- ▶ Hemos mencionado ya la presencia de un controlador de memoria.
- ▶ Entre otros aspectos se definen cuatro buses independientes para una organización de memoria: *data, address, control, y chip select*.

# JEDEC

- ▶ Las DRAM actualmente son un commodity.
- ▶ Cualquier DRAM o DIMM actual tiene especificaciones equivalentes: ancho de bus, capacidad, velocidad, interfaz, por citar las mas importantes.
- ▶ Esta compatibilidad está gobernada por JEDEC (Joint Electron Device Engineering Council).
- ▶ Hemos mencionado ya la presencia de un controlador de memoria.
- ▶ Entre otros aspectos se definen cuatro buses independientes para una organización de memoria: *data, address, control, y chip select*.

# JEDEC

- ▶ Las DRAM actualmente son un commodity.
- ▶ Cualquier DRAM o DIMM actual tiene especificaciones equivalentes: ancho de bus, capacidad, velocidad, interfaz, por citar las mas importantes.
- ▶ Esta compatibilidad está gobernada por JEDEC (**J**oint **E**lectron **D**evice **E**ngineering **C**ouncil).
- ▶ Hemos mencionado ya la presencia de un controlador de memoria.
- ▶ Entre otros aspectos se definen cuatro buses independientes para una organización de memoria: *data, address, control, y chip select*.

# JEDEC

- ▶ Las DRAM actualmente son un commodity.
- ▶ Cualquier DRAM o DIMM actual tiene especificaciones equivalentes: ancho de bus, capacidad, velocidad, interfaz, por citar las mas importantes.
- ▶ Esta compatibilidad está gobernada por JEDEC (**J**oint **E**lectron **D**evice **E**ngineering **C**ouncil).
- ▶ Hemos mencionado ya la presencia de un controlador de memoria.
- ▶ Entre otros aspectos se definen cuatro buses independientes para una organización de memoria: *data, address, control, y chip select*.



# JEDEC

- ▶ Las DRAM actualmente son un commodity.
- ▶ Cualquier DRAM o DIMM actual tiene especificaciones equivalentes: ancho de bus, capacidad, velocidad, interfaz, por citar las mas importantes.
- ▶ Esta compatibilidad está gobernada por JEDEC (**J**oint **E**lectron **D**evice **E**ngineering **C**ouncil).
- ▶ Hemos mencionado ya la presencia de un controlador de memoria.
- ▶ Entre otros aspectos se definen cuatro buses independientes para una organización de memoria: *data*, *address*, *control*, y *chip select*.

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- **JEDEC SDRAM**
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

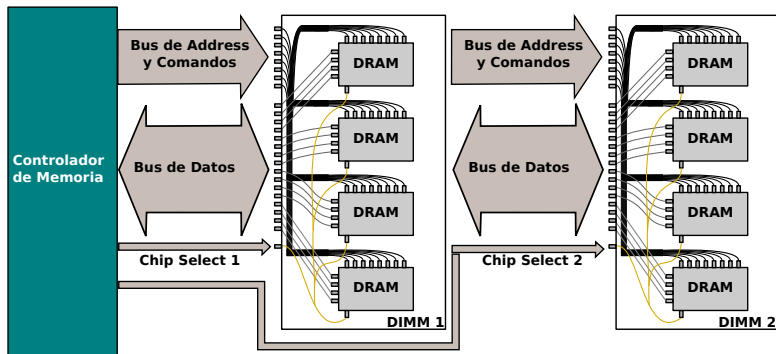
## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

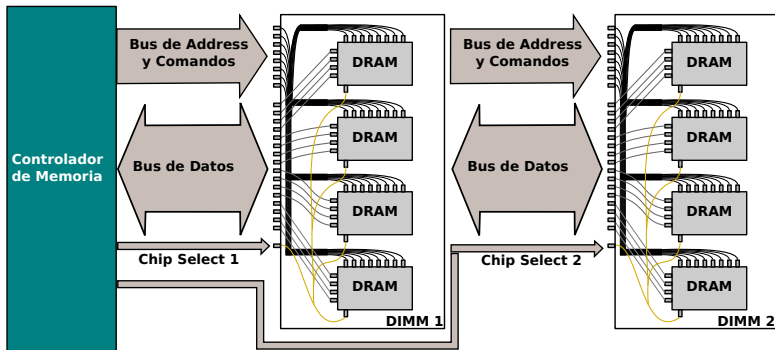
- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Estándares: JEDEC



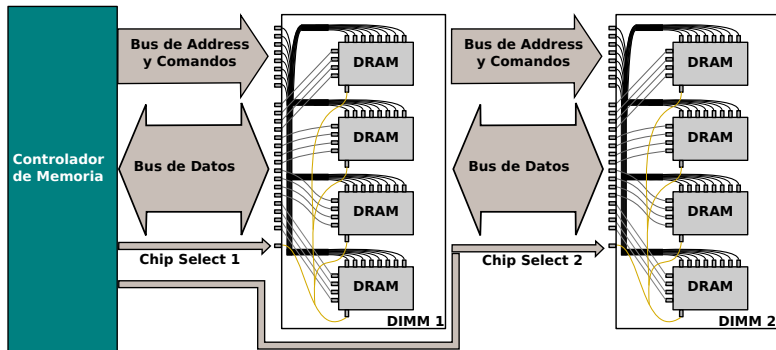
- El *data* bus es por lo general el mas ancho, por razones de optimización del ancho de banda. Típicamente es de 64 bits aunque en sistemas de alto rendimiento se encuentran *data* buses mas anchos (hasta 256 bits)

# Estándares: JEDEC



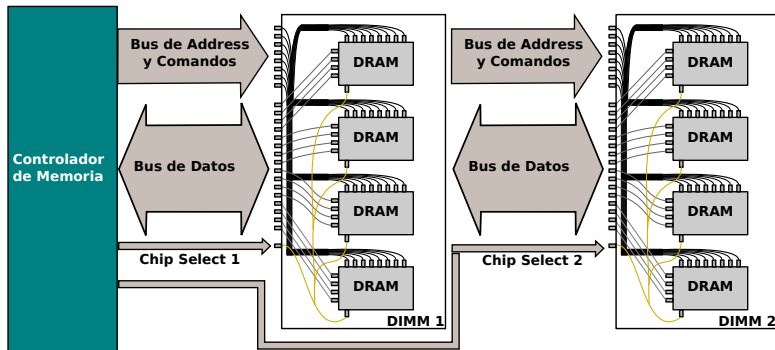
- ▶ El *address* bus envía la información de row y col a los DIMMs. Su ancho es proporcional a la capacidad de DRAM instalada
- ▶ El *control* bus lleva los strobes de row y col, output enable, clock enable, clock, etc.
- ▶ El *chip select* bus es individual para cada rango. Ej: 2 rangos por DIMM, implican dos *chip select* bus por cada DIMM.

# Estándares: JEDEC



- ▶ El *address* bus envía la información de row y col a los DIMMs. Su ancho es proporcional a la capacidad de DRAM instalada
- ▶ El *control* bus lleva los strobes de row y col, output enable, clock enable, clock, etc.
- ▶ El *chip select* bus es individual para cada rango. Ej: 2 rangos por DIMM, implican dos *chip select* bus por cada DIMM.

# Estándares: JEDEC



- ▶ El *address* bus envía la información de row y col a los DIMMs. Su ancho es proporcional a la capacidad de DRAM instalada
- ▶ El *control* bus lleva los strobes de row y col, output enable, clock enable, clock, etc.
- ▶ El *chip select* bus es individual para cada rango. Ej: 2 rangos por DIMM, implican dos *chip select* bus por cada DIMM.

# Topología básica

- ▶ El/los procesador/es un sistema no se conecta/n directamente a los dispositivos de memoria DRAM, sino que intermedia/n uno o mas controladores de memoria.
- ▶ Un procesador puede conectarse a uno o mas controladores, o varios procesadores pueden utilizar un único controlador.
- ▶ Las redes de conexionado (bus, crossbar, etc.) entre el procesador y el controlador de memoria son diferentes de las que conectan al controlador con la DRAM (manejan diferentes señales).
- ▶ Así el procesador se independiza de las características físicas de los diferentes DRAM devices.
- ▶ Un controlador de memoria se conecta a uno o mas DIMMs,  $\implies$  controla varios DRAM devices.

# Topología básica

- ▶ El/los procesador/es un sistema no se conecta/n directamente a los dispositivos de memoria DRAM, sino que intermedia/n uno o mas controladores de memoria.
- ▶ Un procesador puede conectarse a uno o mas controladores, o varios procesadores pueden utilizar un único controlador.
- ▶ Las redes de conexionado (bus, crossbar, etc.) entre el procesador y el controlador de memoria son diferentes de las que conectan al controlador con la DRAM (manejan diferentes señales).
- ▶ Así el procesador se independiza de las características físicas de los diferentes DRAM devices.
- ▶ Un controlador de memoria se conecta a uno o mas DIMMs,  $\implies$  controla varios DRAM devices.



# Topología básica

- ▶ El/los procesador/es un sistema no se conecta/n directamente a los dispositivos de memoria DRAM, sino que intermedia/n uno o mas controladores de memoria.
- ▶ Un procesador puede conectarse a uno o mas controladores, o varios procesadores pueden utilizar un único controlador.
- ▶ Las redes de conexionado (bus, crossbar, etc.) entre el procesador y el controlador de memoria son diferentes de las que conectan al controlador con la DRAM (manejan diferentes señales).
- ▶ Así el procesador se independiza de las características físicas de los diferentes DRAM devices.
- ▶ Un controlador de memoria se conecta a uno o mas DIMMs,  $\implies$  controla varios DRAM devices.

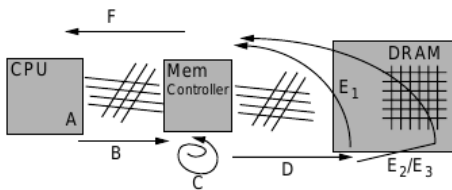
# Topología básica

- ▶ El/los procesador/es un sistema no se conecta/n directamente a los dispositivos de memoria DRAM, sino que intermedia/n uno o mas controladores de memoria.
- ▶ Un procesador puede conectarse a uno o mas controladores, o varios procesadores pueden utilizar un único controlador.
- ▶ Las redes de conexionado (bus, crossbar, etc.) entre el procesador y el controlador de memoria son diferentes de las que conectan al controlador con la DRAM (manejan diferentes señales.
- ▶ Así el procesador se independiza de las características físicas de los diferentes DRAM devices.
- ▶ Un controlador de memoria se conecta a uno o mas DIMMs,  $\implies$  controla varios DRAM devices.

# Topología básica

- ▶ El/los procesador/es un sistema no se conecta/n directamente a los dispositivos de memoria DRAM, sino que intermedia/n uno o mas controladores de memoria.
- ▶ Un procesador puede conectarse a uno o mas controladores, o varios procesadores pueden utilizar un único controlador.
- ▶ Las redes de conexionado (bus, crossbar, etc.) entre el procesador y el controlador de memoria son diferentes de las que conectan al controlador con la DRAM (manejan diferentes señales.
- ▶ Así el procesador se independiza de las características físicas de los diferentes DRAM devices.
- ▶ Un controlador de memoria se conecta a uno o mas DIMMs,  $\implies$  controla varios DRAM devices.

## Diagrama general y transacciones



A: Transaction request may be delayed in Queue  
B: Transaction request sent to Memory Controller  
C: Transaction converted to Command Sequences  
(may be queued)

D: Command/s Sent to DRAM

E<sub>1</sub>: Requires only a **CAS** or

E<sub>2</sub>: Requires **RAS + CAS** or

E<sub>3</sub>: Requires **PRE + RAS + CAS**

F: Transaction sent back to CPU

$$\text{DRAM Latency} = A + B + C + D + E + F$$

- ▶ El procesador ordena y encola los requerimientos de acceso a memoria y los envía al controlador.
- ▶ El controlador los encola hasta que la DRAM esté lista y se hayan resuelto los requerimientos en curso.
- ▶ La interfaz del controlador de DRAM con los DIMM o con los DRAM devices, es mucho mas compleja que la de por ejemplo una memoria estática. En principio convierte dirección física en códigos de CAS y RAS. El detalle a continuación...

# Longitud de ráfaga (busrt) Programable

- ▶ **Objetivo:** Mejorar el ancho de banda de las transferencias de memoria.
- ▶ Los SDRAM disponen de un Registro de Modo en el que se programa entre otras cosas la cantidad de lecturas o escrituras en ráfaga (burst) que se llevarán a cabo una vez activada la línea CAS.
- ▶ En lugar conmutar la señal CAS para transmitir en cada flanco un paquete de datos, como las BEDO, las SDRAM conmutan una vez la línea CAS, y se transmiten tantos paquetes de datos en cada flanco ascendente de la señal de clock como indique su Registro de Modo.
- ▶ Al no tener que conmutar la línea CAS el controlador de memoria, el controlador puede enviar requerimientos por el bus de control a otros DRAM Devices aumentando el ancho de banda del bus.
- ▶ Una vez terminado el burst en este DRAM Device, habrá otros Devices ya precargados y disponibles para transmitir datos en ráfaga.

# Longitud de ráfaga (busrt) Programable

- ▶ Objetivo: Mejorar el ancho de banda de las transferencias de memoria.
- ▶ Los SDRAM disponen de un Registro de Modo en el que se programa entre otras cosas la cantidad de lecturas o escrituras en ráfaga (burst) que se llevarán a cabo una vez activada la línea CAS.
- ▶ En lugar conmutar la señal CAS para transmitir en cada flanco un paquete de datos, como las BEDO, las SDRAM conmutan una vez la línea CAS, y se transmiten tantos paquetes de datos en cada flanco ascendente de la señal de clock como indique su Registro de Modo.
- ▶ Al no tener que conmutar la línea CAS el controlador de memoria, el controlador puede enviar requerimientos por el bus de control a otros DRAM Devices aumentando el ancho de banda del bus.
- ▶ Una vez terminado el burst en este DRAM Device, habrá otros Devices ya precargados y disponibles para transmitir datos en ráfaga.

# Longitud de ráfaga (busrt) Programable

- ▶ Objetivo: Mejorar el ancho de banda de las transferencias de memoria.
- ▶ Los SDRAM disponen de un Registro de Modo en el que se programa entre otras cosas la cantidad de lecturas o escrituras en ráfaga (burst) que se llevarán a cabo una vez activada la línea CAS.
- ▶ En lugar conmutar la señal CAS para transmitir en cada flanco un paquete de datos, como las BEDO, las SDRAM conmutan una vez la línea CAS, y se transmiten tantos paquetes de datos en cada flanco ascendente de la señal de clock como indique su Registro de Modo.
- ▶ Al no tener que conmutar la línea CAS el controlador de memoria, el controlador puede enviar requerimientos por el bus de control a otros DRAM Devices aumentando el ancho de banda del bus.
- ▶ Una vez terminado el burst en este DRAM Device, habrá otros Devices ya precargados y disponibles para transmitir datos en ráfaga.

# Longitud de ráfaga (busrt) Programable

- ▶ Objetivo: Mejorar el ancho de banda de las transferencias de memoria.
- ▶ Los SDRAM disponen de un Registro de Modo en el que se programa entre otras cosas la cantidad de lecturas o escrituras en ráfaga (burst) que se llevarán a cabo una vez activada la línea CAS.
- ▶ En lugar conmutar la señal CAS para transmitir en cada flanco un paquete de datos, como las BEDO, las SDRAM conmutan una vez la línea CAS, y se transmiten tantos paquetes de datos en cada flanco ascendente de la señal de clock como indique su Registro de Modo.
- ▶ Al no tener que conmutar la línea CAS el controlador de memoria, el controlador puede enviar requerimientos por el bus de control a otros DRAM Devices aumentando el ancho de banda del bus.
- ▶ Una vez terminado el burst en este DRAM Device, habrá otros Devices ya precargados y disponibles para transmitir datos en ráfaga.



# Longitud de ráfaga (busrt) Programable

- ▶ Objetivo: Mejorar el ancho de banda de las transferencias de memoria.
- ▶ Los SDRAM disponen de un Registro de Modo en el que se programa entre otras cosas la cantidad de lecturas o escrituras en ráfaga (burst) que se llevarán a cabo una vez activada la línea CAS.
- ▶ En lugar conmutar la señal CAS para transmitir en cada flanco un paquete de datos, como las BEDO, las SDRAM conmutan una vez la línea CAS, y se transmiten tantos paquetes de datos en cada flanco ascendente de la señal de clock como indique su Registro de Modo.
- ▶ Al no tener que conmutar la línea CAS el controlador de memoria, el controlador puede enviar requerimientos por el bus de control a otros DRAM Devices aumentando el ancho de banda del bus.
- ▶ Una vez terminado el burst en este DRAM Device, habrá otros Devices ya precargados y disponibles para transmitir datos en ráfaga.

# CAS Latency Programmable

- ▶ CAS Latency o Delay: Cantidad de ciclos de clock que transcurren desde que se activa CAS hasta que el dato está disponible.
- ▶ Se lo solía referir también como tiempo de acceso.
- ▶ Las SDRAM permiten programar esta cantidad de ciclo de clock en su Registro de Modo.
- ▶ De este modo se facilita la coexistencia en un mismo sistema de memoria de SDRAM devices de diferentes características en lo que a tiempo de acceso se refiere.

# CAS Latency Programmable

- ▶ CAS Latency o Delay: Cantidad de ciclos de clock que transcurren desde que se activa CAS hasta que el dato está disponible.
- ▶ Se lo solía referir también como tiempo de acceso.
- ▶ Las SDRAM permiten programar esta cantidad de ciclo de clock en su Registro de Modo.
- ▶ De este modo se facilita la coexistencia en un mismo sistema de memoria de SDRAM devices de diferentes características en lo que a tiempo de acceso se refiere.

# CAS Latency Programmable

- ▶ CAS Latency o Delay: Cantidad de ciclos de clock que transcurren desde que se activa CAS hasta que el dato está disponible.
- ▶ Se lo solía referir también como tiempo de acceso.
- ▶ Las SDRAM permiten programar esta cantidad de ciclo de clock en su Registro de Modo.
- ▶ De este modo se facilita la coexistencia en un mismo sistema de memoria de SDRAM devices de diferentes características en lo que a tiempo de acceso se refiere.

# CAS Latency Programmable

- ▶ CAS Latency o Delay: Cantidad de ciclos de clock que transcurren desde que se activa CAS hasta que el dato está disponible.
- ▶ Se lo solía referir también como tiempo de acceso.
- ▶ Las SDRAM permiten programar esta cantidad de ciclo de clock en su Registro de Modo.
- ▶ De este modo se facilita la coexistencia en un mismo sistema de memoria de SDRAM devices de diferentes características en lo que a tiempo de acceso se refiere.

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Dual Edge Clocking

- ▶ Las DDR SDRAM conservan las características de las SDRAM agregando nuevas prestaciones.
- ▶ Las transacciones de bits de direcciones y control se llevan a cabo sincronizando con el flanco ascendente de la señal de clock igual que en el caso de las SDRAM.
- ▶ Las transmisiones de datos por su parte usan un clock dual edge.
- ▶ A igual tamaño de ráfaga, las DDR requieren la mitad de ciclos de clock para acceder a los datos.

# Dual Edge Clocking

- ▶ Las DDR SDRAM conservan las características de las SDRAM agregando nuevas prestaciones.
- ▶ Las transacciones de bits de direcciones y control se llevan a cabo sincronizando con el flanco ascendente de la señal de clock igual que en el caso de las SDRAM.
- ▶ Las transmisiones de datos por su parte usan un clock dual edge.
- ▶ A igual tamaño de ráfaga, las DDR requieren la mitad de ciclos de clock para acceder a los datos.

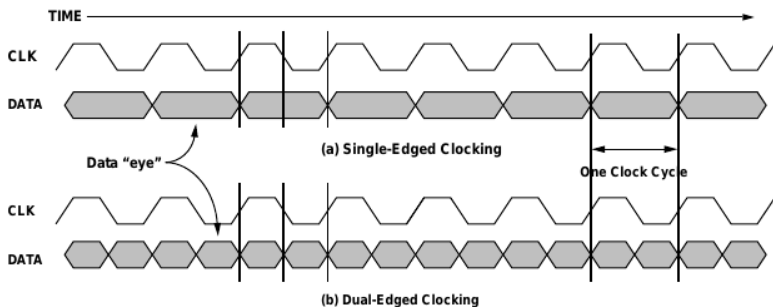


# Dual Edge Clocking

- ▶ Las DDR SDRAM conservan las características de las SDRAM agregando nuevas prestaciones.
- ▶ Las transacciones de bits de direcciones y control se llevan a cabo sincronizando con el flanco ascendente de la señal de clock igual que en el caso de las SDRAM.
- ▶ Las transmisiones de datos por su parte usan un clock dual edge.
- ▶ A igual tamaño de ráfaga, las DDR requieren la mitad de ciclos de clock para acceder a los datos.

# Dual Edge Clcking

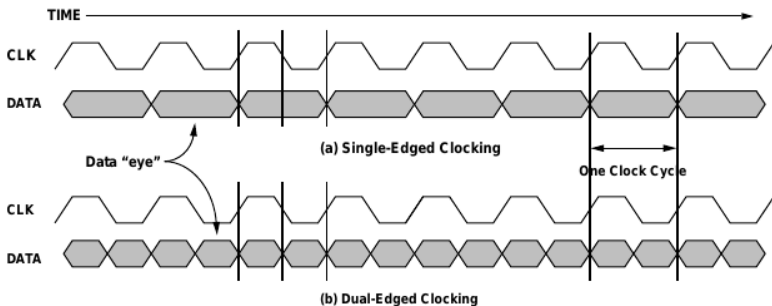
- ▶ Las DDR SDRAM conservan las características de las SDRAM agregando nuevas prestaciones.
- ▶ Las transacciones de bits de direcciones y control se llevan a cabo sincronizando con el flanco ascendente de la señal de clock igual que en el caso de las SDRAM.
- ▶ Las transmisiones de datos por su parte usan un clock dual edge.



- ▶ A igual tamaño de ráfaga, las DDR requieren la mitad de ciclos de clock para acceder a los datos.

# Dual Edge Clcking

- ▶ Las DDR SDRAM conservan las características de las SDRAM agregando nuevas prestaciones.
- ▶ Las transacciones de bits de direcciones y control se llevan a cabo sincronizando con el flanco ascendente de la señal de clock igual que en el caso de las SDRAM.
- ▶ Las transmisiones de datos por su parte usan un clock dual edge.



- ▶ A igual tamaño de ráfaga, las DDR requieren la mitad de ciclos de clock para acceder a los datos.

# Dual Edge Clocking

- ▶ Al duplicarse la velocidad de acceso, el tamaño de la ventana válida (también llamada “ojo de datos”) es la mitad que en una SDRAM.
- ▶ Se requiere asegurar precisión tanto en el vuelco de los datos al bus como en el muestreo a la entrada de la DRAM cuando ésta es escrita.
- ▶ Ambas operaciones tienen diferente relación con el flanco de la señal de clock.
- ▶ En la lectura del DRAM device, el flanco dispara el vuelco de los datos hacia el bus. Se refiere como alineada al flanco de los datos.
- ▶ En la escritura del DRAM device, se necesita alinearse al centro de los datos para asegurarse que la habilitación de los circuitos de entrada del DRAM device encuentren señales válidas en el bus de datos

# Dual Edge Clocking

- ▶ Al duplicarse la velocidad de acceso, el tamaño de la ventana válida (también llamada “ojo de datos”) es la mitad que en una SDRAM.
- ▶ Se requiere asegurar precisión tanto en el vuelco de los datos al bus como en el muestreo a la entrada de la DRAM cuando ésta es escrita.
- ▶ Ambas operaciones tienen diferente relación con el flanco de la señal de clock.
- ▶ En la lectura del DRAM device, el flanco dispara el vuelco de los datos hacia el bus. Se refiere como alineada al flanco de los datos.
- ▶ En la escritura del DRAM device, se necesita alinearse al centro de los datos para asegurarse que la habilitación de los circuitos de entrada del DRAM device encuentren señales válidas en el bus de datos

# Dual Edge Clocking

- ▶ Al duplicarse la velocidad de acceso, el tamaño de la ventana válida (también llamada “ojo de datos”) es la mitad que en una SDRAM.
- ▶ Se requiere asegurar precisión tanto en el vuelco de los datos al bus como en el muestreo a la entrada de la DRAM cuando ésta es escrita.
- ▶ Ambas operaciones tienen diferente relación con el flanco de la señal de clock.
- ▶ En la lectura del DRAM device, el flanco dispara el vuelco de los datos hacia el bus. Se refiere como alineada al flanco de los datos.
- ▶ En la escritura del DRAM device, se necesita alinearse al centro de los datos para asegurarse que la habilitación de los circuitos de entrada del DRAM device encuentren señales válidas en el bus de datos

# Dual Edge Clocking

- ▶ Al duplicarse la velocidad de acceso, el tamaño de la ventana válida (también llamada “ojo de datos”) es la mitad que en una SDRAM.
- ▶ Se requiere asegurar precisión tanto en el vuelco de los datos al bus como en el muestreo a la entrada de la DRAM cuando ésta es escrita.
- ▶ Ambas operaciones tienen diferente relación con el flanco de la señal de clock.
- ▶ En la lectura del DRAM device, el flanco dispara el vuelco de los datos hacia el bus. Se refiere como alineada al flanco de los datos.
- ▶ En la escritura del DRAM device, se necesita alinearse al centro de los datos para asegurarse que la habilitación de los circuitos de entrada del DRAM device encuentren señales válidas en el bus de datos

# Dual Edge Clocking

- ▶ Al duplicarse la velocidad de acceso, el tamaño de la ventana válida (también llamada “ojo de datos”) es la mitad que en una SDRAM.
- ▶ Se requiere asegurar precisión tanto en el vuelco de los datos al bus como en el muestreo a la entrada de la DRAM cuando ésta es escrita.
- ▶ Ambas operaciones tienen diferente relación con el flanco de la señal de clock.
- ▶ En la lectura del DRAM device, el flanco dispara el vuelco de los datos hacia el bus. Se refiere como alineada al flanco de los datos.
- ▶ En la escritura del DRAM device, se necesita alinearse al centro de los datos para asegurarse que la habilitación de los circuitos de entrada del DRAM device encuentren señales válidas en el bus de datos



# Señal adicional: DQS

- ▶ A medida que aumenta la velocidad, los datos en las entradas pueden desfasarse en el tiempo respecto de otras señales que corren libremente como el **Clock**.
- ▶ Pueden existir en las entradas de un DRAM Device desfasajes de las señales de datos de acuerdo a variaciones de temperatura, en la tensión, o en la carga del dispositivo.
- ▶ Estos efectos se acentúan en los dispositivos DDR. Por lo tanto se necesita usar señales de strobe, generadas por el controlador de memoria, en lugar de señales libres como **Clock**.

# Señal adicional: DQS

- ▶ A medida que aumenta la velocidad, los datos en las entradas pueden desfasarse en el tiempo respecto de otras señales que corren libremente como el **Clock**.
- ▶ Pueden existir en las entradas de un DRAM Device desfasajes de las señales de datos de acuerdo a variaciones de temperatura, en la tensión, o en la carga del dispositivo.
- ▶ Estos efectos se acentúan en los dispositivos DDR. Por lo tanto se necesita usar señales de strobe, generadas por el controlador de memoria, en lugar de señales libres como **Clock**.

# Señal adicional: DQS

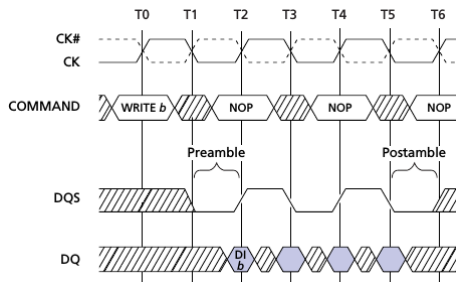
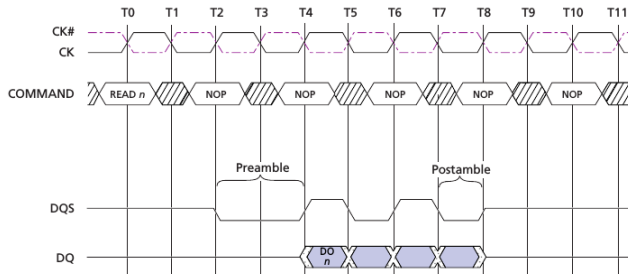
- ▶ A medida que aumenta la velocidad, los datos en las entradas pueden desfasarse en el tiempo respecto de otras señales que corren libremente como el **Clock**.
- ▶ Pueden existir en las entradas de un DRAM Device desfasajes de las señales de datos de acuerdo a variaciones de temperatura, en la tensión, o en la carga del dispositivo.
- ▶ Estos efectos se acentúan en los dispositivos DDR. Por lo tanto se necesita usar señales de strobe, generadas por el controlador de memoria, en lugar de señales libres como **Clock**.

# Señal adicional: DQS

- ▶ A medida que aumenta la velocidad, los datos en las entradas pueden desfasarse en el tiempo respecto de otras señales que corren libremente como el **Clock**.
- ▶ Pueden existir en las entradas de un DRAM Device desfasajes de las señales de datos de acuerdo a variaciones de temperatura, en la tensión, o en la carga del dispositivo.
- ▶ Estos efectos se acentúan en los dispositivos DDR. Por lo tanto se necesita usar señales de strobe, generadas por el controlador de memoria, en lugar de señales libres como **Clock**.

Por ello, los controladores de DRAM que soportan DDR tienen una señal DQS, que para en operaciones READ se genera con su flanco alineado al de la señal de datos y para WRITE se genera centrada en la señal de datos

# Señal adicional: DQS



# On-chip Delay-Locked Loop (DLL)

- ▶ El DLL se encarga de sincronizar la señal DQS y los datos salientes del DDR SDRAM Device con la señal de clock proveniente del controlador de memoria .
- ▶ El DLL sincroniza las lecturas del DDR SDRAM Device, ya que las de escritura se efectúan al ritmo de la señal DQS que es manejada por el controlador de memoria.
- ▶ El controlador de memoria maneja dos clocks internos: Uno que se sincroniza con el clock del sistema, y otro desfasado  $90^\circ$  para sincronizar las escrituras a los DRAM Devices.
- ▶ Como DQS está en fase con los datos, no puede ser utilizada por el controlador de memoria para muestrear directamente los datos. Por eso utiliza DQS para sincronizar los datos salientes del DDR SDRAM Device con los clocks del controlador de memoria.

# On-chip Delay-Locked Loop (DLL)

- ▶ El DLL se encarga de sincronizar la señal DQS y los datos salientes del DDR SDRAM Device con la señal de clock proveniente del controlador de memoria .
- ▶ El DLL sincroniza las lecturas del DDR SDRAM Device, ya que las de escritura se efectúan al ritmo de la señal DQS que es manejada por el controlador de memoria.
- ▶ El controlador de memoria maneja dos clocks internos: Uno que se sincroniza con el clock del sistema, y otro desfasado  $90^\circ$  para sincronizar las escrituras a los DRAM Devices.
- ▶ Como DQS está en fase con los datos, no puede ser utilizada por el controlador de memoria para muestrear directamente los datos. Por eso utiliza DQS para sincronizar los datos salientes del DDR SDRAM Device con los clocks del controlador de memoria.

# On-chip Delay-Locked Loop (DLL)

- ▶ El DLL se encarga de sincronizar la señal DQS y los datos salientes del DDR SDRAM Device con la señal de clock proveniente del controlador de memoria .
- ▶ El DLL sincroniza las lecturas del DDR SDRAM Device, ya que las de escritura se efectúan al ritmo de la señal DQS que es manejada por el controlador de memoria.
- ▶ El controlador de memoria maneja dos clocks internos: Uno que se sincroniza con el clock del sistema, y otro desfasado  $90^\circ$  para sincronizar las escrituras a los DRAM Devices.
- ▶ Como DQS está en fase con los datos, no puede ser utilizada por el controlador de memoria para muestrear directamente los datos. Por eso utiliza DQS para sincronizar los datos salientes del DDR SDRAM Device con los clocks del controlador de memoria.



# On-chip Delay-Locked Loop (DLL)

- ▶ El DLL se encarga de sincronizar la señal DQS y los datos salientes del DDR SDRAM Device con la señal de clock proveniente del controlador de memoria .
- ▶ El DLL sincroniza las lecturas del DDR SDRAM Device, ya que las de escritura se efectúan al ritmo de la señal DQS que es manejada por el controlador de memoria.
- ▶ El controlador de memoria maneja dos clocks internos: Uno que se sincroniza con el clock del sistema, y otro desfasado  $90^\circ$  para sincronizar las escrituras a los DRAM Devices.
- ▶ Como DQS está en fase con los datos, no puede ser utilizada por el controlador de memoria para muestrear directamente los datos. Por eso utiliza DQS para sincronizar los datos salientes del DDR SDRAM Device con los clocks del controlador de memoria.

# On-chip Delay-Locked Loop (DLL)

- ▶ El clock desfasado  $90^\circ$  sincroniza los datos entrantes debido a que el DLL garantiza un mínimo entrelazado entre el el clock global y los datos salientes.
- ▶ Debido a los delays que físicamente generan el receptor de clock, las etapas amplificadoras, los alambres, pads etc, desde la perspectiva del bus, la señal de datos estará desfasada respecto del clock del sistema.
- ▶ Sin embargo la existencia de un circuito DLL **asegura** un delay constante y del valor determinado por el DLL.
- ▶ La diferencia entre ambas situaciones se muestra en el siguiente slide.

# On-chip Delay-Locked Loop (DLL)

- ▶ El clock desfasado  $90^\circ$  sincroniza los datos entrantes debido a que el DLL garantiza un mínimo entrelazado entre el el clock global y los datos salientes.
- ▶ Debido a los delays que físicamente generan el receptor de clock, las etapas amplificadoras, los alambres, pads etc, desde la perspectiva del bus, la señal de datos estará desfasada respecto del clock del sistema.
- ▶ Sin embargo la existencia de un circuito DLL **asegura** un delay constante y del valor determinado por el DLL.
- ▶ La diferencia entre ambas situaciones se muestra en el siguiente slide.

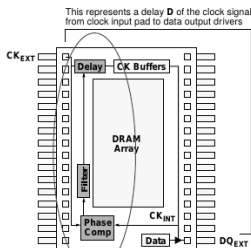
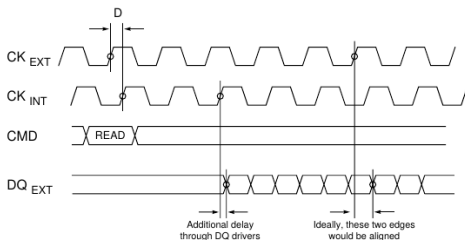
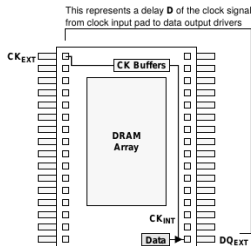
# On-chip Delay-Locked Loop (DLL)

- ▶ El clock desfasado  $90^\circ$  sincroniza los datos entrantes debido a que el DLL garantiza un mínimo entrelazado entre el el clock global y los datos salientes.
- ▶ Debido a los delays que físicamente generan el receptor de clock, las etapas amplificadoras, los alambres, pads etc, desde la perspectiva del bus, la señal de datos estará desfasada respecto del clock del sistema.
- ▶ Sin embargo la existencia de un circuito DLL **asegura** un delay constante y del valor determinado por el DLL.
- ▶ La diferencia entre ambas situaciones se muestra en el siguiente slide.

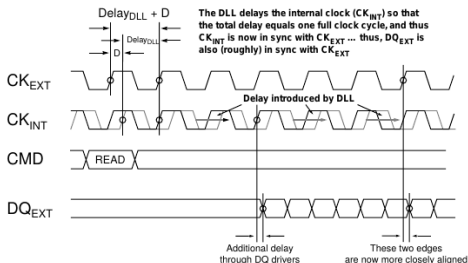
# On-chip Delay-Locked Loop (DLL)

- ▶ El clock desfasado  $90^\circ$  sincroniza los datos entrantes debido a que el DLL garantiza un mínimo entrelazado entre el el clock global y los datos salientes.
- ▶ Debido a los delays que físicamente generan el receptor de clock, las etapas amplificadoras, los alambres, pads etc, desde la perspectiva del bus, la señal de datos estará desfasada respecto del clock del sistema.
- ▶ Sin embargo la existencia de un circuito DLL **asegura** un delay constante y del valor determinado por el DLL.
- ▶ La diferencia entre ambas situaciones se muestra en el siguiente slide.

# On-chip Delay-Locked Loop (DLL)



The Phase Comparison, Loop Filter, and Variable Delay components constitute a DLL



# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

## • Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Lógica de control de un SDRAM Device

- ▶ En esencia los DRAM Devices son circuitos analógicos cuya temporización es naturalmente asincrónica.
- ▶ Los pasos que le toma a la circuitería de una DRAM para almacenar y recuperar datos en forma de carga en un capacitor a través de amplificadores de sensado, le insumen una latencia considerable, la cual se expresa normalmente en nseg., y no en ciclos de clock.
- ▶ Por eso de acuerdo al diseño y al proceso de fabricación cada DRAM tiene un juego de parámetros de timing diferentes.
- ▶ La solución que encontró la industria es proveerles una interfaz sincrónica.
- ▶ Para lidiar con los diferentes juegos de temporización se incluyen uno o varios registro de modo (dependiendo del device).



# Lógica de control de un SDRAM Device

- ▶ En esencia los DRAM Devices son circuitos analógicos cuya temporización es naturalmente asincrónica.
- ▶ Los pasos que le toma a la circuitería de una DRAM para almacenar y recuperar datos en forma de carga en un capacitor a través de amplificadores de sensado, le insumen una latencia considerable, la cual se expresa normalmente en nseg., y no en ciclos de clock.
- ▶ Por eso de acuerdo al diseño y al proceso de fabricación cada DRAM tiene un juego de parámetros de timing diferentes.
- ▶ La solución que encontró la industria es proveerles una interfaz sincrónica.
- ▶ Para lidiar con los diferentes juegos de temporización se incluyen uno o varios registro de modo (dependiendo del device).

# Lógica de control de un SDRAM Device

- ▶ En esencia los DRAM Devices son circuitos analógicos cuya temporización es naturalmente asincrónica.
- ▶ Los pasos que le toma a la circuitería de una DRAM para almacenar y recuperar datos en forma de carga en un capacitor a través de amplificadores de sensado, le insumen una latencia considerable, la cual se expresa normalmente en nseg., y no en ciclos de clock.
- ▶ Por eso de acuerdo al diseño y al proceso de fabricación cada DRAM tiene un juego de parámetros de timing diferentes.
- ▶ La solución que encontró la industria es proveerles una interfaz sincrónica.
- ▶ Para lidiar con los diferentes juegos de temporización se incluyen uno o varios registro de modo (dependiendo del device).

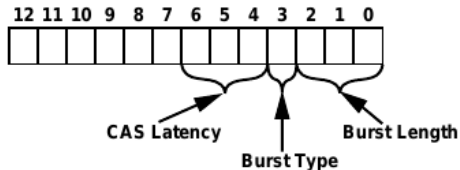
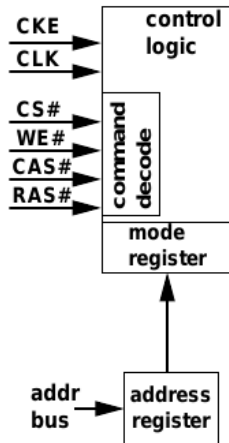
# Lógica de control de un SDRAM Device

- ▶ En esencia los DRAM Devices son circuitos analógicos cuya temporización es naturalmente asincrónica.
- ▶ Los pasos que le toma a la circuitería de una DRAM para almacenar y recuperar datos en forma de carga en un capacitor a través de amplificadores de sensado, le insumen una latencia considerable, la cual se expresa normalmente en nseg., y no en ciclos de clock.
- ▶ Por eso de acuerdo al diseño y al proceso de fabricación cada DRAM tiene un juego de parámetros de timing diferentes.
- ▶ La solución que encontró la industria es proveerles una interfaz sincrónica.
- ▶ Para lidiar con los diferentes juegos de temporización se incluyen uno o varios registro de modo (dependiendo del device).

# Lógica de control de un SDRAM Device

- ▶ En esencia los DRAM Devices son circuitos analógicos cuya temporización es naturalmente asincrónica.
- ▶ Los pasos que le toma a la circuitería de una DRAM para almacenar y recuperar datos en forma de carga en un capacitor a través de amplificadores de sensado, le insumen una latencia considerable, la cual se expresa normalmente en nseg., y no en ciclos de clock.
- ▶ Por eso de acuerdo al diseño y al proceso de fabricación cada DRAM tiene un juego de parámetros de timing diferentes.
- ▶ La solución que encontró la industria es proveerles una interfaz sincrónica.
- ▶ Para lidiar con los diferentes juegos de temporización se incluyen uno o varios registro de modo (dependiendo del device).

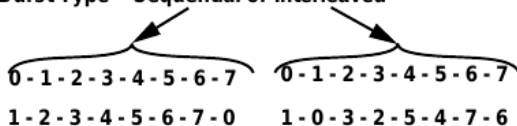
# Registro de modo de un SDRAM Device



**Burst Length = 1, 2, 4, 8, or Page mode**

**CAS Latency = 2, 3 (4, 5, etc. in special versions)**

**Burst Type = Sequential or Interleaved**



# Registro de modo de un SDRAM Device

## ► Posee tres campos

- 1 **CAS latency**: En las hojas de datos se lo llama **CL**. De acuerdo al valor de este campo de bits, el SDRAM Device devolverá el dato dos o tres ciclos de clock luego de activada CAS
- 2 **Burst Type**: Determina el orden en el que el SDRAM Device devuelve los datos.
- 3 **Burst Length**: Determina el número de columnas de una fila entera que el SDRAM Device devuelve por cada comando CAS: 1, 2, 4, u 8 bits.

► Los DDR SDRAM Devices y los D-RDRAM Devices, tienen mas Registros de Modos en los que se pueden configurar mas parámetros de control.

# Registro de modo de un SDRAM Device

## ► Posee tres campos

- 1 **CAS latency**: En las hojas de datos se lo llama **CL**. De acuerdo al valor de este campo de bits, el SDRAM Device devolverá el dato dos o tres ciclos de clock luego de activada CAS
- 2 **Burst Type**: Determina el orden en el que el SDRAM Device devuelve los datos.
- 3 **Burst Length**: Determina el número de columnas de una fila entera que el SDRAM Device devuelve por cada comando CAS: 1, 2, 4, u 8 bits.

► Los DDR SDRAM Devices y los D-RDRAM Devices, tienen mas Registros de Modos en los que se pueden configurar mas parámetros de control.

# Registro de modo de un SDRAM Device

► Posee tres campos

- 1 **CAS latency**: En las hojas de datos se lo llama **CL**. De acuerdo al valor de este campo de bits, el SDRAM Device devolverá el dato dos o tres ciclos de clock luego de activada CAS
- 2 **Burst Type**: Determina el orden en el que el SDRAM Device devuelve los datos.
- 3 **Burst Length**: Determina el número de columnas de una fila entera que el SDRAM Device devuelve por cada comando CAS: 1, 2, 4, u 8 bits.

► Los DDR SDRAM Devices y los D-RDRAM Devices, tienen mas Registros de Modos en los que se pueden configurar mas parámetros de control.



# Registro de modo de un SDRAM Device

► Posee tres campos

- 1 **CAS latency**: En las hojas de datos se lo llama **CL**. De acuerdo al valor de este campo de bits, el SDRAM Device devolverá el dato dos o tres ciclos de clock luego de activada CAS
- 2 **Burst Type**: Determina el orden en el que el SDRAM Device devuelve los datos.
- 3 **Burst Length**: Determina el número de columnas de una fila entera que el SDRAM Device devuelve por cada comando CAS: 1, 2, 4, u 8 bits.

► Los DDR SDRAM Devices y los D-RDRAM Devices, tienen mas Registros de Modos en los que se pueden configurar mas parámetros de control.

# Registro de modo de un SDRAM Device

► Posee tres campos

- 1 **CAS latency**: En las hojas de datos se lo llama **CL**. De acuerdo al valor de este campo de bits, el SDRAM Device devolverá el dato dos o tres ciclos de clock luego de activada CAS
- 2 **Burst Type**: Determina el orden en el que el SDRAM Device devuelve los datos.
- 3 **Burst Length**: Determina el número de columnas de una fila entera que el SDRAM Device devuelve por cada comando CAS: 1, 2, 4, u 8 bits.

► Los DDR SDRAM Devices y los D-RDRAM Devices, tienen mas Registros de Modos en los que se pueden configurar mas parámetros de control.

# Diferentes configuraciones para el mismo tamaño

- ▶ Las SDRAM se clasifican de acuerdo al número de bits contenidos en el Device. Pero independientemente de este dato básico se puede organizar de diferentes maneras, de acuerdo con el tamaño de la palabra de datos que se pretenda leer o escribir en el Device.
- ▶ A pesar de los cambios de configuración, el número de filas de la SDRAM del Device, se mantiene constante, mientras que el número de columnas por fila decrece a medida que se agranda la cantidad de bits que entrega el Device en un mismo CAS.

# Diferentes configuraciones para el mismo tamaño

- ▶ Las SDRAM se clasifican de acuerdo al número de bits contenidos en el Device. Pero independientemente de este dato básico se puede organizar de diferentes maneras, de acuerdo con el tamaño de la palabra de datos que se pretenda leer o escribir en el Device.

Configuración	64Mbit x 4	32Mbit x 8	16Mbit x 16
Nº Bancos	4	4	4
Nº Filas	8192	8192	8192
Nº Columnas	2048	1024	512
Ancho Data Bus	4	8	16

- ▶ A pesar de los cambios de configuración, el número de filas de la SDRAM del Device, se mantiene constante, mientras que el número de columnas por fila decrece a medida que se agranda la cantidad de bits que entrega el Device en un mismo CAS.

# Diferentes configuraciones para el mismo tamaño

- ▶ Las SDRAM se clasifican de acuerdo al número de bits contenidos en el Device. Pero independientemente de este dato básico se puede organizar de diferentes maneras, de acuerdo con el tamaño de la palabra de datos que se pretenda leer o escribir en el Device.

Configuración	64Mbit x 4	32Mbit x 8	16Mbit x 16
Nº Bancos	4	4	4
Nº Filas	8192	8192	8192
Nº Columnas	2048	1024	512
Ancho Data Bus	4	8	16

- ▶ A pesar de los cambios de configuración, el número de filas de la SDRAM del Device, se mantiene constante, mientras que el número de columnas por fila decrece a medida que se agranda la cantidad de bits que entrega el Device en un mismo CAS.

# Diferentes configuraciones para el mismo tamaño

- ▶ Para capacidades mayores de memoria no siempre se cumple que el tamaño de fila se mantenga constante. De hecho para una DRAM de 1 Gbit normalmente se tiene una configuración como la siguiente
- ▶ Estas diferencias en las configuraciones implican diferente cantidad de bits por línea, lo cual deriva en diferentes cantidades de circuitos que se activan en un comando CAS variando las características de performance y consumo para diferentes configuraciones en una misma generación de Dispositivos.

# Diferentes configuraciones para el mismo tamaño

- ▶ Para capacidades mayores de memoria no siempre se cumple que el tamaño de fila se mantenga constante. De hecho para una DRAM de 1 Gbit normalmente se tiene una configuración como la siguiente

Configuración	256Mbit x 4	128Mbit x 8	64Mbit x 16
Nº Bancos	8	8	8
Nº Filas	16384	16384	8192
Nº Columnas	2048	1024	1024
Ancho Data Bus	4	8	16

- ▶ Estas diferencias en las configuraciones implican diferente cantidad de bits por línea, lo cual deriva en diferentes cantidades de circuitos que se activan en un comando CAS variando las características de performance y consumo para diferentes configuraciones en una misma generación de Dispositivos.

# Diferentes configuraciones para el mismo tamaño

- ▶ Para capacidades mayores de memoria no siempre se cumple que el tamaño de fila se mantenga constante. De hecho para una DRAM de 1 Gbit normalmente se tiene una configuración como la siguiente

Configuración	256Mbit x 4	128Mbit x 8	64Mbit x 16
Nº Bancos	8	8	8
Nº Filas	16384	16384	8192
Nº Columnas	2048	1024	1024
Ancho Data Bus	4	8	16

- ▶ Estas diferencias en las configuraciones implican diferente cantidad de bits por línea, lo cual deriva en diferentes cantidades de circuitos que se activan en un comando CAS variando las características de performance y consumo para diferentes configuraciones en una misma generación de Dispositivos.



# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Funciones que se resuelven

- ▶ Un protocolo de acceso a memoria define los comandos y restricciones de temporización que un controlador de memoria debe utilizar para manejar sus transferencias de datos con la memoria DRAM
- ▶ Nos proponemos analizar un protocolo de acceso genérico aplicable a SDRAM y DDRx SDRAM.
- ▶ Examinar en detalle cualquier protocolo de acceso a una SDRAM es sumamente complejo.
- ▶ Esta complejidad proviene de la gran variedad de comandos en los sistemas de memoria modernos.
- ▶ Sin embargo es posible definir un set de comandos básicos que es posible encontrar en la mayoría de los sistemas SDRAM modernos.

# Funciones que se resuelven

- ▶ Un protocolo de acceso a memoria define los comandos y restricciones de temporización que un controlador de memoria debe utilizar para manejar sus transferencias de datos con la memoria DRAM
- ▶ Nos proponemos analizar un protocolo de acceso genérico aplicable a SDRAM y DDRx SDRAM.
- ▶ Examinar en detalle cualquier protocolo de acceso a una SDRAM es sumamente complejo.
- ▶ Esta complejidad proviene de la gran variedad de comandos en los sistemas de memoria modernos.
- ▶ Sin embargo es posible definir un set de comandos básicos que es posible encontrar en la mayoría de los sistemas SDRAM modernos.

# Funciones que se resuelven

- ▶ Un protocolo de acceso a memoria define los comandos y restricciones de temporización que un controlador de memoria debe utilizar para manejar sus transferencias de datos con la memoria DRAM
- ▶ Nos proponemos analizar un protocolo de acceso genérico aplicable a SDRAM y DDRx SDRAM.
- ▶ Examinar en detalle cualquier protocolo de acceso a una SDRAM es sumamente complejo.
- ▶ Esta complejidad proviene de la gran variedad de comandos en los sistemas de memoria modernos.
- ▶ Sin embargo es posible definir un set de comandos básicos que es posible encontrar en la mayoría de los sistemas SDRAM modernos.

# Funciones que se resuelven

- ▶ Un protocolo de acceso a memoria define los comandos y restricciones de temporización que un controlador de memoria debe utilizar para manejar sus transferencias de datos con la memoria DRAM
- ▶ Nos proponemos analizar un protocolo de acceso genérico aplicable a SDRAM y DDRx SDRAM.
- ▶ Examinar en detalle cualquier protocolo de acceso a una SDRAM es sumamente complejo.
- ▶ Esta complejidad proviene de la gran variedad de comandos en los sistemas de memoria modernos.
- ▶ Sin embargo es posible definir un set de comandos básicos que es posible encontrar en la mayoría de los sistemas SDRAM modernos.

# Funciones que se resuelven

- ▶ Un protocolo de acceso a memoria define los comandos y restricciones de temporización que un controlador de memoria debe utilizar para manejar sus transferencias de datos con la memoria DRAM
- ▶ Nos proponemos analizar un protocolo de acceso genérico aplicable a SDRAM y DDRx SDRAM.
- ▶ Examinar en detalle cualquier protocolo de acceso a una SDRAM es sumamente complejo.
- ▶ Esta complejidad proviene de la gran variedad de comandos en los sistemas de memoria modernos.
- ▶ Sin embargo es posible definir un set de comandos básicos que es posible encontrar en la mayoría de los sistemas SDRAM modernos.

# Valores estándar de Timing

Parametro	Descripción
$t_{AL}$	<b>Added Latency</b> en accesos a columnas. Se usa en DDRx SDRAM devices para comandos CAS enviados.
$t_{BURST}$	<b>Duración del Burst de Datos</b> . Período de tiempo en que el <i>data burst</i> ocupa el bus de datos. Típicamente 4 u 8 <i>beats</i> de datos. En DDR SDRAM, 4 <i>beats</i> de datos consumen 2 ciclos completos de clock.
$t_{CAS}$	<b>Column Access Strobe latency</b> . Intervalo de Tiempo entre el comando de acceso a columna y el inicio del retorno de datos desde el/los DRAM device/s. Normalmente nombrado como $t_{CL}$ .
$t_{CCD}$	<b>Column-to-Column Delay</b> . Mínimo timing en comandos de columna, determinado por la longitud del <i>burst</i> interno ( <i>prefetch</i> ). Las lecturas de columnas utilizan múltiples <i>bursts</i> internos para formar un <i>burst</i> mas largo. $t_{CCD}$ es 2 <i>beats</i> (1 ciclo) en SDRAM DDR, 4 (2 ciclos) en DDR2, y 8 (4 ciclos) en DDR3.
$t_{CMD}$	<b>Tiempo de Transporte de Comando</b> . Tiempo durante el cual el comando ocupa el bus de Comandos y es transportado desde el Controlador de DRAM hasta el/los DRAM device/s.
$t_{CWD}$	<b>Column Write Delay</b> . Tiempo entre el envío de un comando <i>column-write</i> y el vuelco de datos al bus por parte del controlador de DRAM.
$t_{FAW}$	<b>Four (row) bank Activation Window</b> . Ranura de tiempo en la cual se pueden activar un máximo de cuatro bancos. Limita el perfil de corriente pico en DDR2 y DDR3 DRAMs con mas de 4 bancos.
$t_{OST}$	<b>ODT Switching Time</b> . Tiempo para conmutar el control ODT, de rango a rango.

# Valores estándar de Timing

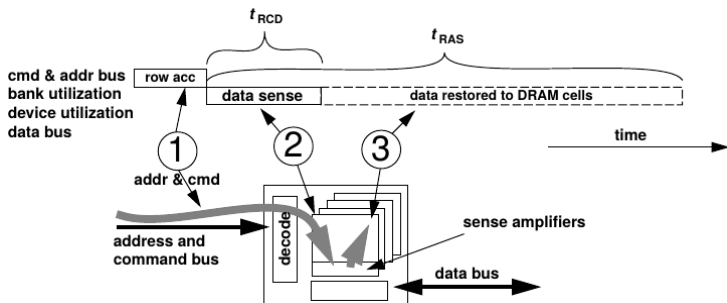
Parametro	Descripción
$t_{RAS}$	<b>Row Access Strobe.</b> Tiempo entre el comando <i>row access</i> y la regeneración de datos en un DRAM array. Un banco de DRAM no se puede precargar hasta el último $t_{RAS}$ luego de la activación del banco previa.
$t_{RC}$	<b>Row Cycle.</b> Tiempo entre accesos a diferentes filas en un banco. $t_{RC} = t_{RAS} + t_{RP}$ .
$t_{RCD}$	<b>Row to Column command Delay.</b> Tiempo entre un <i>row access</i> y los datos listos en los amplificadores de sensado.
$t_{RFC}$	<b>Refresh Cycle time.</b> Tiempo entre los comandos <i>Refresh</i> y <i>Activation</i> .
$t_{RP}$	<b>Row Precharge.</b> Tiempo que le toma a un DRAM array precargarse para otro <i>row access</i> .
$t_{RRD}$	<b>Row activation to Row activation Delay.</b> Tiempo mínimo entre dos comandos de activación de fila en el mismo DRAM device. Limita el perfil de corriente pico.
$t_{RTP}$	<b>Read to Precharge.</b> Tiempo entre un comando de lectura y otro de precarga.
$t_{RTRS}$	<b>Rank-to-rank switching time.</b> Un ciclo completo en DDR SDRAM. No se usa en sistemas SDRAM ni RDRAM Directo.
$t_{WR}$	<b>Write Recovery time.</b> Mínimo intervalo de tiempo entre el fin de una escritura burst y el inicio de un comando de precarga. Permite a los amplificadores de sensado restaurar la carga en las celdas.
$t_{WTR}$	<b>Write To Read delay time.</b> Tiempo mínimo entre el final de una escritura burst y el inicio de un comando <i>column read</i> . Permite a la E/S forzar a los amplificadores de sensado a saturar su salida antes de que se ejecute el comando Read.





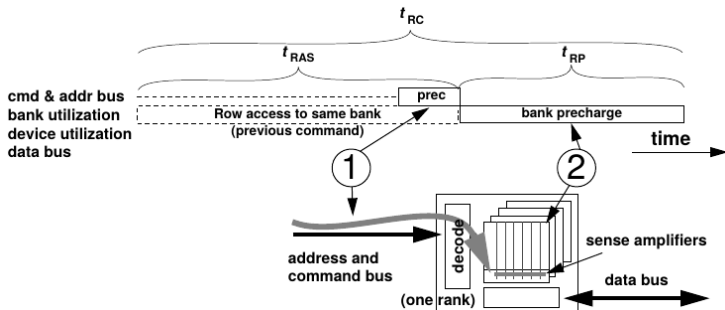


# Comando Row Access: Formato y timing.



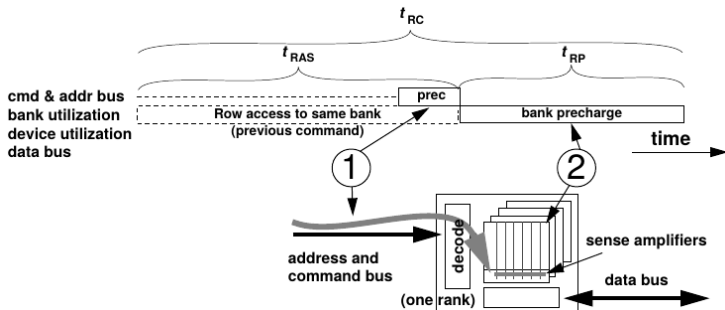
- ▶ Luego de  $t_{RCD}$  (**Row to Column command Delay**) los datos están listos en los amplificadores de sensado.
- ▶ El tiempo total entre el comando Row Access y la realimentación de los datos en el DRAM array es  $t_{RAS}$  (**Row Access Strobe latency**). Un banco DRAM no se puede precargar al menos hasta el último  $t_{RAS}$  posterior a la activación previa del banco.

# Comando Precharge: Formato y timing.



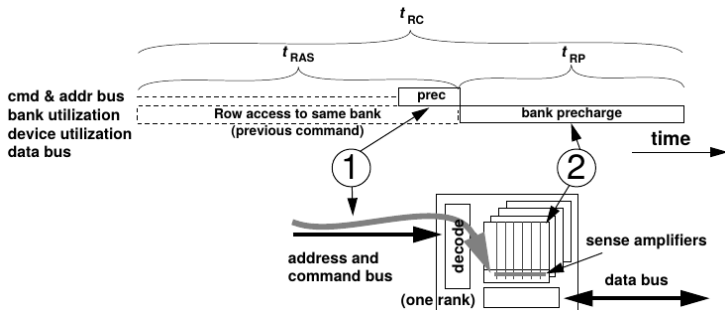
- El acceso de datos en una DRAM es un proceso de dos pasos:
  1. Un comando *Row Access* para mover los datos desde el array de celdas de memoria al array de amplificadores de sensado.

## Comando Precharge: Formato y timing.



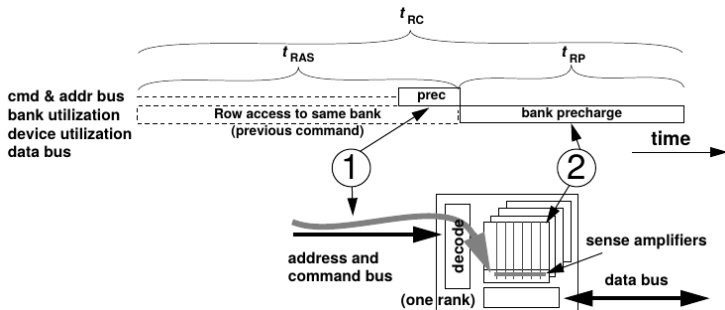
- ▶ El acceso de datos en una DRAM es un proceso de dos pasos:
  - 1 Un comando *Row Access* para mover los datos desde el array de celdas de memoria al array de amplificadores de sensado.
  - 2 Se cachean los datos en el amplificador de sensado para los *Column Access* posteriores

## Comando Precharge: Formato y timing.



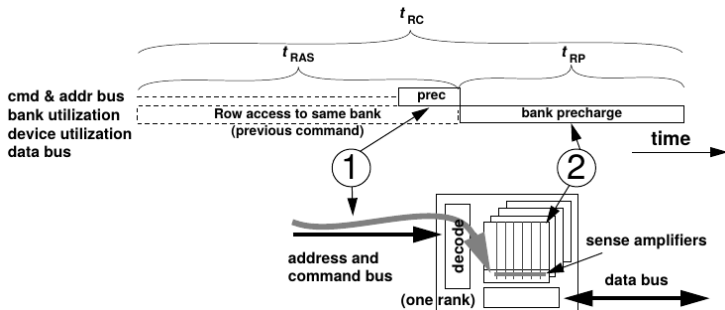
- ▶ El acceso de datos en una DRAM es un proceso de dos pasos:
  - 1 Un comando *Row Access* para mover los datos desde el array de celdas de memoria al array de amplificadores de sensado.
  - 2 Se cachean los datos en el amplificador de sensado para los *Column Access* posteriores

## Comando Precharge: Formato y timing.



- ▶ El acceso de datos en una DRAM es un proceso de dos pasos:
  - 1 Un comando *Row Access* para mover los datos desde el array de celdas de memoria al array de amplificadores de sensado.
  - 2 Se cachean los datos en el amplificador de sensado para los *Column Access* posteriores

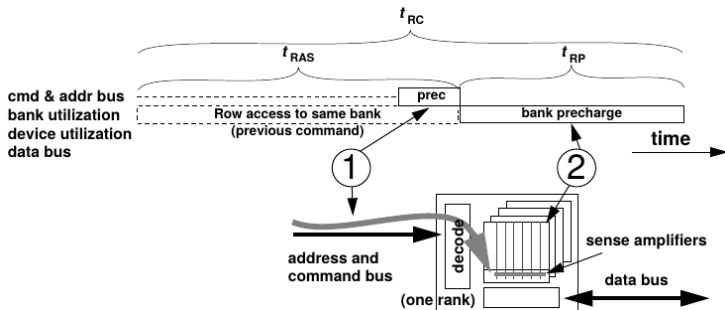
# Comando Precharge: Formato y timing.



- *Precharge*, completa la secuencia de *Row Access*, reseteando los amplificadores de sensado y las bitlines de modo de dejarlas listas para un nuevo *Row Access* al mismo array de celdas.
- El tiempo que toma precargar un DRAM array para otro *Row Access* es  $t_{RP}$  (**Row Precharge**).

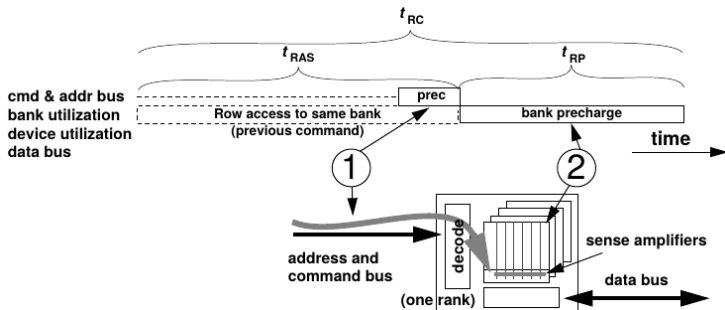


# Comando Precharge: Formato y timing.



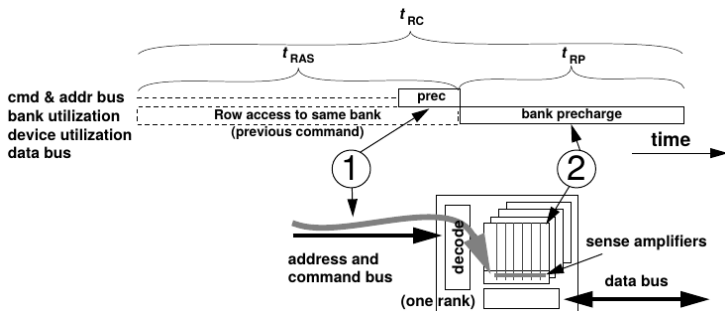
- *Precharge*, completa la secuencia de *Row Access*, reseteando los amplificadores de sensado y las bitlines de modo de dejarlas listas para un nuevo *Row Access* al mismo array de celdas.
- El tiempo que toma precargar un DRAM array para otro *Row Access* es  $t_{RP}$  (**Row Precharge**).

# Comando Precharge: Formato y timing.



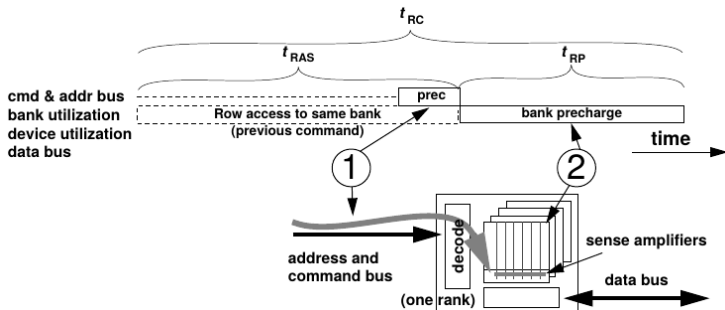
- *Precharge*, completa la secuencia de *Row Access*, reseteando los amplificadores de sensado y las bitlines de modo de dejarlas listas para un nuevo *Row Access* al mismo array de celdas.
- El tiempo que toma precargar un DRAM array para otro *Row Access* es  $t_{RP}$  (**Row Precharge**).

# Comando Precharge: Formato y timing.



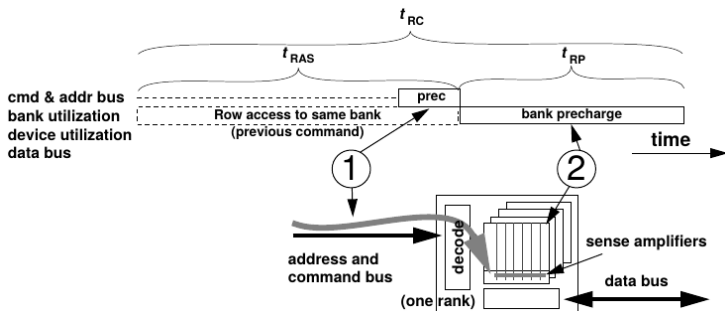
- $t_{RP}$  y  $t_{RAS}$  se combinan formando  $t_{RC}$  (**Row Cycle Time**), que indica el tiempo mínimo que necesita un DRAM device para poner disponibles en los amplificadores de sensado los datos de los arrays de celdas, restaurar los datos de esas celdas, y precargar los bitlines al nivel de tensión de referencia para un nuevo comando *Row Access*.

# Comando Precharge: Formato y timing.



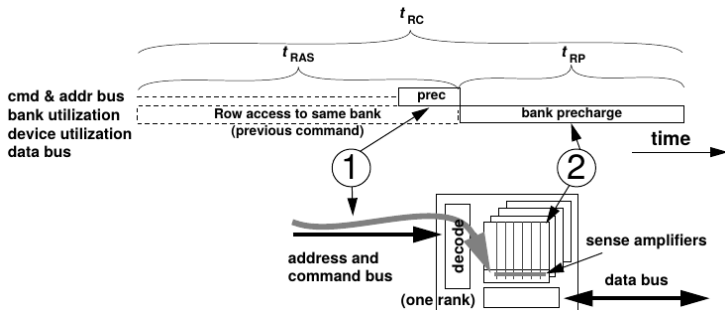
- $t_{RP}$  y  $t_{RAS}$  se combinan formando  $t_{RC}$  (**Row Cycle Time**), que indica el tiempo mínimo que necesita un DRAM device para poner disponibles en los amplificadores de sensado los datos de los arrays de celdas, restaurar los datos de esas celdas, y precargar los bitlines al nivel de tensión de referencia para un nuevo comando *Row Access*.

# Comando Precharge: Formato y timing.



- $t_{RC}$  es la limitación fundamental de la velocidad a la que se pueden obtener datos de filas diferentes del mismo banco de DRAM. También se lo conoce como *random row-cycle time* de un DRAM device.

# Comando Precharge: Formato y timing.



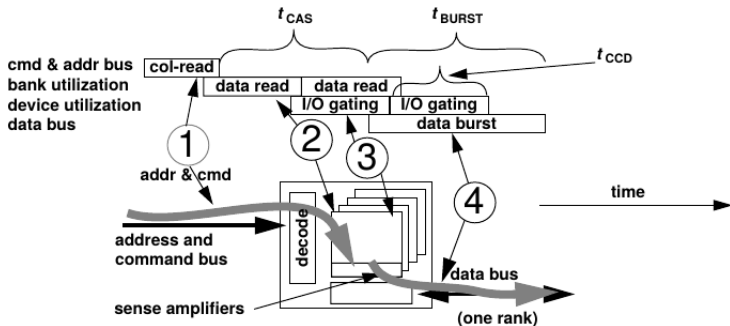
- $t_{RC}$  es la limitación fundamental de la velocidad a la que se pueden obtener datos de filas diferentes del mismo banco de DRAM. También se lo conoce como *random row-cycle time* de un DRAM device.

# Comando Column Read: Formato y timing

- ▶ Mueve los datos desde los amplificadores de sensado hasta el controlador de memoria.
- ▶  $t_{CAS}$  **Column Access Strobe latency**, también llamado  $t_{CL}$ , es el tiempo que le demanda al DRAM device poner el dato en el bus luego de recibir el comando Column Read.

# Comando Column Read: Formato y timing

- Mueve los datos desde los amplificadores de sensado hasta el controlador de memoria.

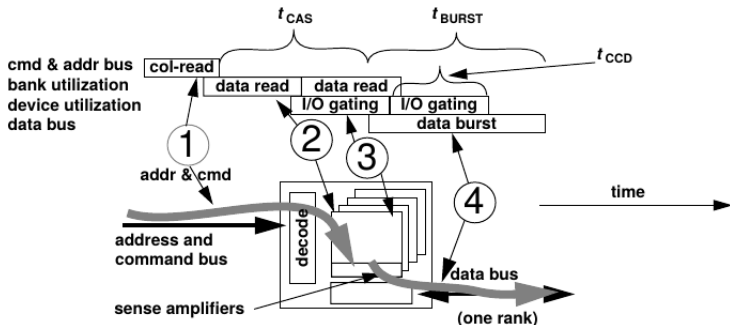


- $t_{CAS}$  **Column Access Strobe latency**, también llamado  $t_{CL}$ , es el tiempo que le demanda al DRAM device poner el dato en el bus luego de recibir el comando Column Read.



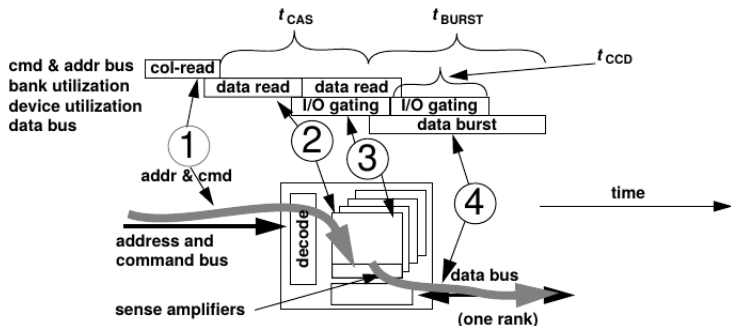
# Comando Column Read: Formato y timing

- Mueve los datos desde los amplificadores de sensado hasta el controlador de memoria.



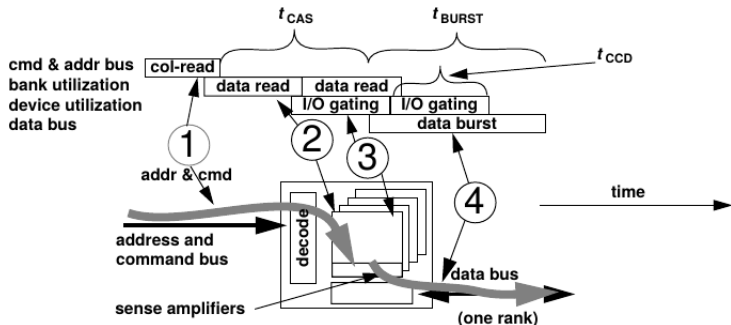
- $t_{CAS}$  **Column Access Strobe latency**, también llamado  $t_{CL}$ , es el tiempo que le demanda al DRAM device poner el dato en el bus luego de recibir el comando Column Read.

# Comando Column Read: Formato y timing



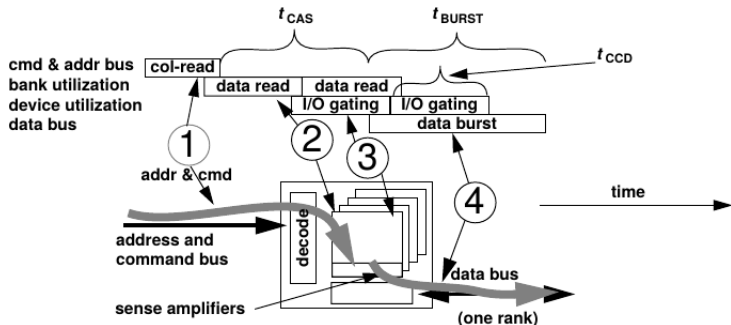
- ▶ Los DDR SDRAM devices mueven los datos internamente en pequeñas ráfagas (burst) a alta velocidad (n-prefetch).
- ▶ En el caso del gráfico se muestra un device que entrega datos en dos bursts rápidos (2n-prefetch), y los vuelca al bus de datos en un solo burst.

# Comando Column Read: Formato y timing



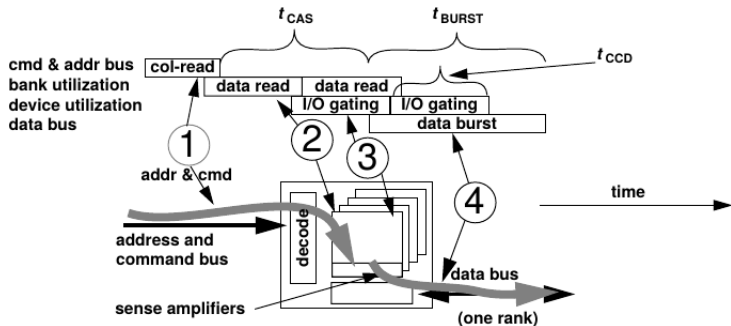
- ▶ Los DDR SDRAM devices mueven los datos internamente en pequeñas ráfagas (burst) a alta velocidad (n-prefetch).
- ▶ En el caso del gráfico se muestra un device que entrega datos en dos bursts rápidos (2n-prefetch), y los vuelca al bus de datos en un solo burst.

# Comando Column Read: Formato y timing



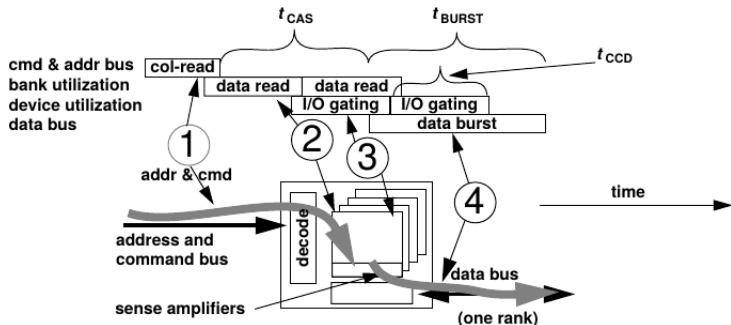
- ▶ Los DDR SDRAM devices mueven los datos internamente en pequeñas ráfagas (burst) a alta velocidad (n-prefetch).
- ▶ En el caso del gráfico se muestra un device que entrega datos en dos bursts rápidos (2n-prefetch), y los vuelca al bus de datos en un solo burst.

# Comando Column Read: Formato y timing.



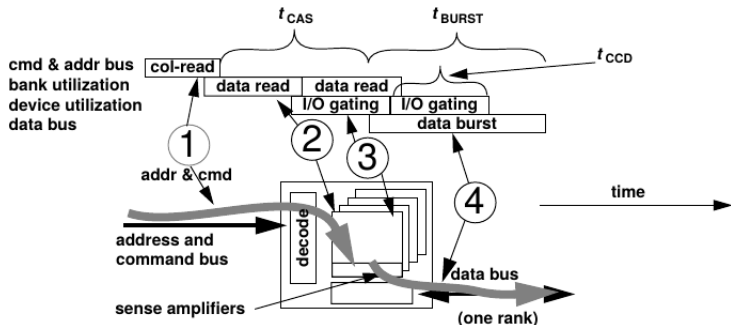
- $t_{CCD}$  **Column-to-Column Delay** es la duración de un pulso burst interno. Su valor mínimo está determinado por el valor de bits de prebúsqueda. En este ejemplo se usan 2 bits de prebúsqueda, por lo tanto,  $t_{CCD}$  será de un ciclo de clock.

# Comando Column Read: Formato y timing.



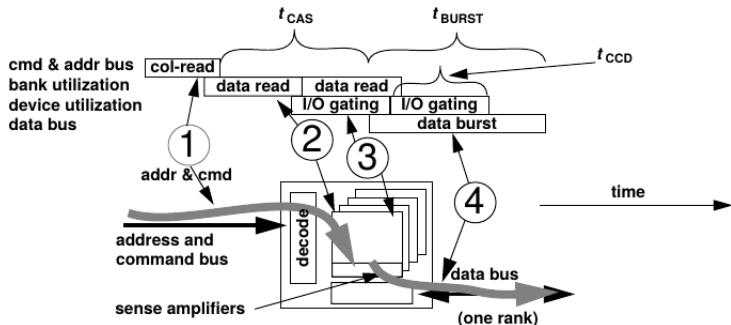
- $t_{CCD}$  **Column-to-Column Delay** es la duración de un pulso burst interno. Su valor mínimo está determinado por el valor de bits de prebúsqueda. En este ejemplo se usan 2 bits de prebúsqueda, por lo tanto,  $t_{CCD}$  será de un ciclo de clock.

# Comando Column Read: Formato y timing.



- ▶ La duración de un burst completo de datos en el bus es  $t_{BURST}$   
**Data burst duration.**
- ▶ Usar bits de prebúsqueda mejora el tiempo total de Column Read si se logra:  $t_{CCD} < t_{BURST}$

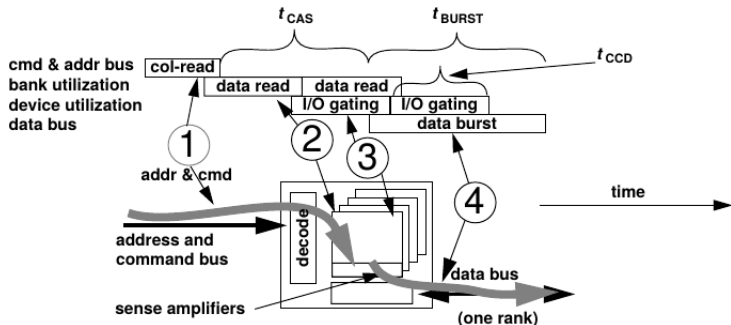
# Comando Column Read: Formato y timing.



- ▶ La duración de un burst completo de datos en el bus es  $t_{BURST}$   
**Data burst duration.**
- ▶ Usar bits de prebúsqueda mejora el tiempo total de Column Read si se logra:  $t_{CCD} < t_{BURST}$

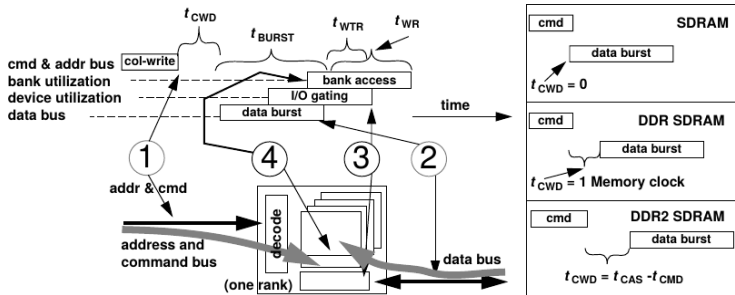


# Comando Column Read: Formato y timing.



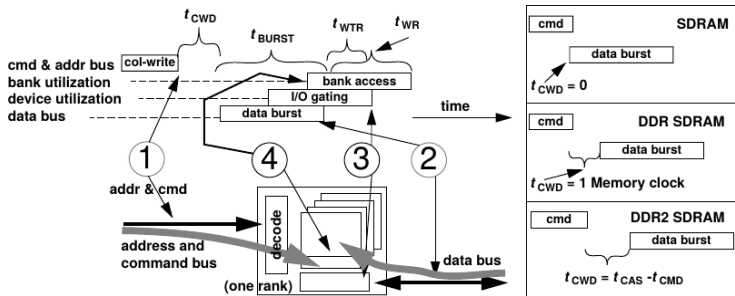
- ▶ La duración de un burst completo de datos en el bus es  $t_{BURST}$   
**Data burst duration.**
- ▶ Usar bits de prebúsqueda mejora el tiempo total de Column Read si se logra:  $t_{CCD} < t_{BURST}$

# Comando Column Write: Formato y timing.



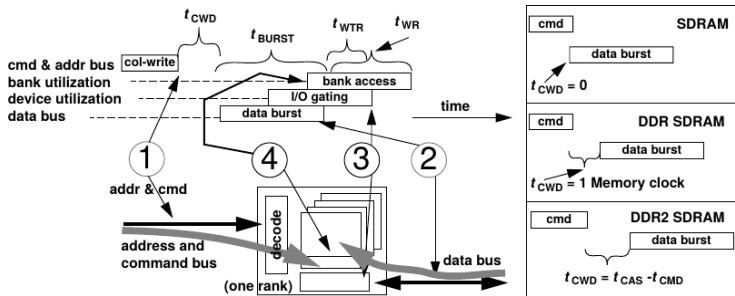
- Mueve los datos desde el controlador de memoria hasta los amplificadores de sensado.
- El controlador de memoria pone el comando Column Write en el bus de comandos, y luego de un tiempo  $t_{CWD}$  (**Column Write Delay**), pone los datos en el bus de datos. El valor de este tiempo depende de la tecnología de memoria (lado derecho de la figura).

# Comando Column Write: Formato y timing.



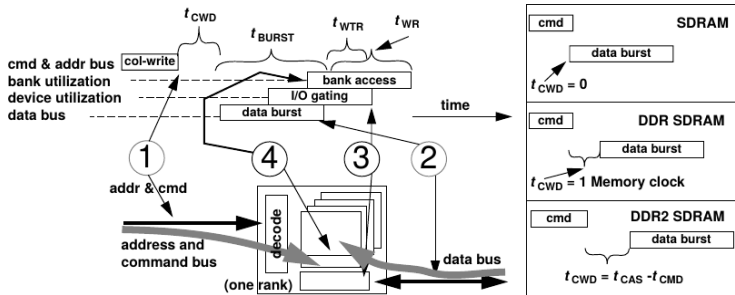
- Mueve los datos desde el controlador de memoria hasta los amplificadores de sensado.
- El controlador de memoria pone el comando Column Write en el bus de comandos, y luego de un tiempo  $t_{CWD}$  (**Column Write Delay**), pone los datos en el bus de datos. El valor de este tiempo depende de la tecnología de memoria (lado derecho de la figura).

# Comando Column Write: Formato y timing.



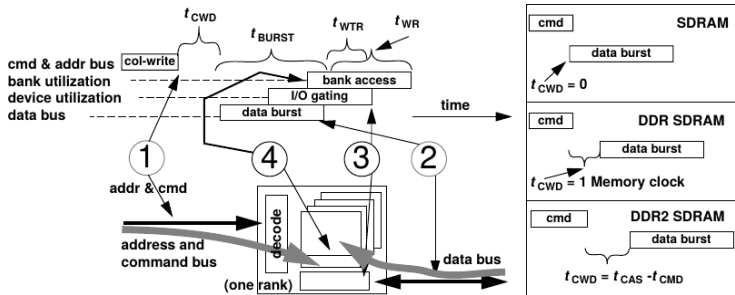
- Mueve los datos desde el controlador de memoria hasta los amplificadores de sensado.
- El controlador de memoria pone el comando Column Write en el bus de comandos, y luego de un tiempo  $t_{CWD}$  (**Column Write Delay**), pone los datos en el bus de datos. El valor de este tiempo depende de la tecnología de memoria (lado derecho de la figura).

# Comando Column Write: Formato y timing.



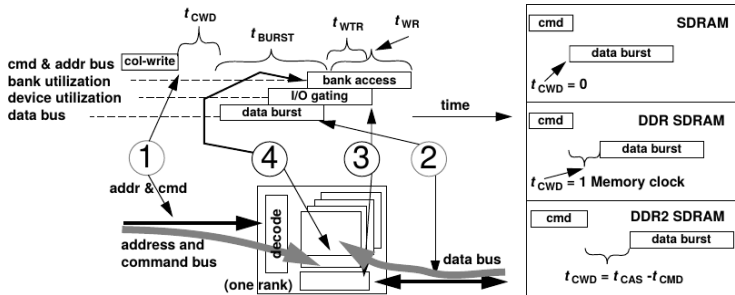
- ▶ La ráfaga (burst) completa de datos en el bus lleva un tiempo  $t_{BURST}$  (**Data burst duration**).
- ▶ El DRAM device emplea un tiempo  $t_{WTR}$  (**Write To Read delay time**) en liberar los recursos de gating de E/S. Este tiempo debe ser especialmente tenido en cuenta cuando al comando Write lo sigue un comando Read.

# Comando Column Write: Formato y timing.



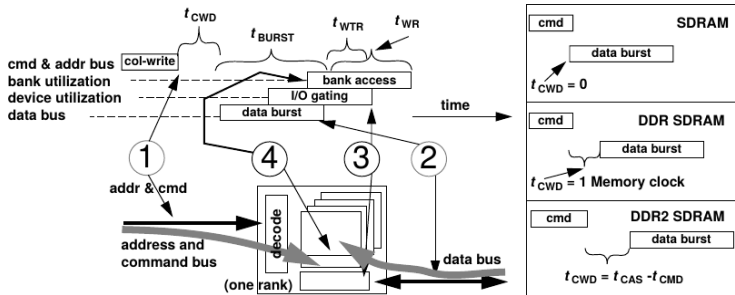
- ▶ La ráfaga (burst) completa de datos en el bus lleva un tiempo  $t_{BURST}$  (**Data burst duration**).
- ▶ El DRAM device emplea un tiempo  $t_{WTR}$  (**Write To Read delay time**) en liberar los recursos de gating de E/S. Este tiempo debe ser especialmente tenido en cuenta cuando al comando Write lo sigue un comando Read.

# Comando Column Write: Formato y timing.



- ▶ La ráfaga (burst) completa de datos en el bus lleva un tiempo  $t_{BURST}$  (**Data burst duration**).
- ▶ El DRAM device emplea un tiempo  $t_{WTR}$  (**Write To Read delay time**) en liberar los recursos de gating de E/S. Este tiempo debe ser especialmente tenido en cuenta cuando al comando Write lo sigue un comando Read.

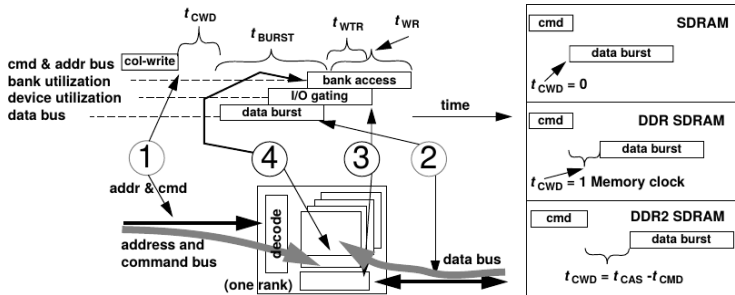
# Comando Column Write: Formato y timing.



- Los datos tomados por el gating de E/S se propagan hasta los gate arrays luego de un tiempo  $t_{WR}$  (**Write Recovery time**). Este tiempo debe ser especialmente tenido en cuenta si al comando Write lo sigue un comando Precharge.



# Comando Column Write: Formato y timing.



- Los datos tomados por el gating de E/S se propagan hasta los gate arrays luego de un tiempo  $t_{WR}$  (**Write Recovery time**). Este tiempo debe ser especialmente tenido en cuenta si al comando Write lo sigue un comando Precharge.

# Refresco de memoria

- ▶ Tiene por objeto asegurar la integridad de la información, ya que los capacitores inexorablemente encontrarán un camino de impedancia (alta pero no infinita) a través del cual descargarse.
- ▶ Refrescar implica leer para regenerar la carga. Esto consume ancho de banda en el bus que une el controlador y la memoria DRAM.
- ▶ Algunos controladores son diseñados para minimizar la complejidad del mecanismo de refresco, de memoria, otros para optimizar el uso de ancho de banda del bus en esta operación, y otros intentan minimizar el consumo de energía consumida en esta operación
- ▶ Por lo general todos los controladores contienen un Refresh Row Register al que utilizan para refrescar una fila completa de celdas con un solo comando.

# Refresco de memoria

- ▶ Tiene por objeto asegurar la integridad de la información, ya que los capacitores inexorablemente encontrarán un camino de impedancia (alta pero no infinita) a través del cual descargarse.
- ▶ Refrescar implica leer para regenerar la carga. Esto consume ancho de banda en el bus que une el controlador y la memoria DRAM.
- ▶ Algunos controladores son diseñados para minimizar la complejidad del mecanismo de refresco, de memoria, otros para optimizar el uso de ancho de banda del bus en esta operación, y otros intentan minimizar el consumo de energía consumida en esta operación
- ▶ Por lo general todos los controladores contienen un Refresh Row Register al que utilizan para refrescar una fila completa de celdas con un solo comando.

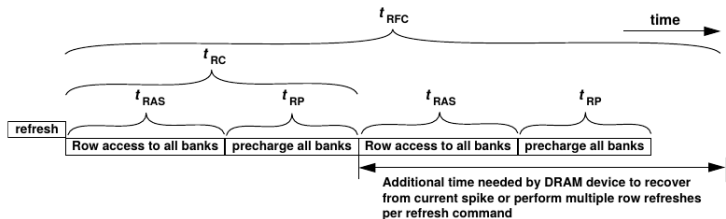
# Refresco de memoria

- ▶ Tiene por objeto asegurar la integridad de la información, ya que los capacitores inexorablemente encontrarán un camino de impedancia (alta pero no infinita) a través del cual descargarse.
- ▶ Refrescar implica leer para regenerar la carga. Esto consume ancho de banda en el bus que une el controlador y la memoria DRAM.
- ▶ Algunos controladores son diseñados para minimizar la complejidad del mecanismo de refresco, de memoria, otros para optimizar el uso de ancho de banda del bus en esta operación, y otros intentan minimizar el consumo de energía consumida en esta operación
- ▶ Por lo general todos los controladores contienen un Refresh Row Register al que utilizan para refrescar una fila completa de celdas con un solo comando.

# Refresco de memoria

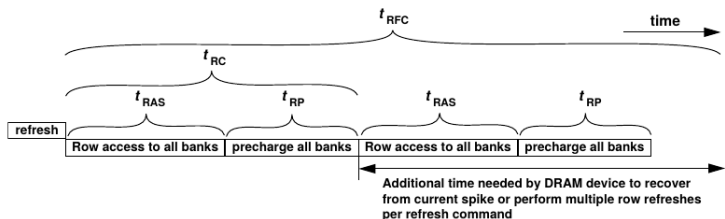
- ▶ Tiene por objeto asegurar la integridad de la información, ya que los capacitores inexorablemente encontrarán un camino de impedancia (alta pero no infinita) a través del cual descargarse.
- ▶ Refrescar implica leer para regenerar la carga. Esto consume ancho de banda en el bus que une el controlador y la memoria DRAM.
- ▶ Algunos controladores son diseñados para minimizar la complejidad del mecanismo de refresco, de memoria, otros para optimizar el uso de ancho de banda del bus en esta operación, y otros intentan minimizar el consumo de energía consumida en esta operación
- ▶ Por lo general todos los controladores contienen un Refresh Row Register al que utilizan para refrescar una fila completa de celdas con un solo comando.

# Refresco de memoria



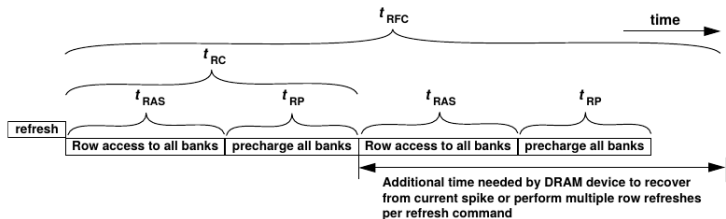
- ▶ El Refresh Row Register contiene siempre la Row Address de la última fila accedida.
- ▶ Cada vez que el Controlador de Memoria envía un comando Refresh, el DRAM Device incrementa este registro e inicia un row cycle en cada fila de cada banco cuya dirección coincida con la del Refresh Row Register,
- ▶  $t_{RFC}$  **Refresh Cycle time**. Intervalo de tiempo entre los comandos de Refresco y activación.

# Refresco de memoria



- ▶ El Refresh Row Register contiene siempre la Row Address de la última fila accedida.
- ▶ Cada vez que el Controlador de Memoria envía un comando Refresh, el DRAM Device incrementa este registro e inicia un row cycle en cada fila de cada banco cuya dirección coincida con la del Refresh Row Register,
- ▶  $t_{RFC}$  **Refresh Cycle time**. Intervalo de tiempo entre los comandos de Refresco y activación.

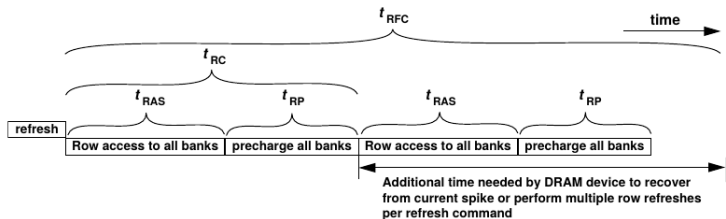
# Refresco de memoria



- ▶ El Refresh Row Register contiene siempre la Row Address de la última fila accedida.
- ▶ Cada vez que el Controlador de Memoria envía un comando Refresh, el DRAM Device incrementa este registro e inicia un row cycle en cada fila de cada banco cuya dirección coincida con la del Refresh Row Register,
- ▶  $t_{RFC}$  **Refresh Cycle time**. Intervalo de tiempo entre los comandos de Refresco y activación.



# Refresco de memoria



- ▶ El Refresh Row Register contiene siempre la Row Address de la última fila accedida.
- ▶ Cada vez que el Controlador de Memoria envía un comando Refresh, el DRAM Device incrementa este registro e inicia un row cycle en cada fila de cada banco cuya dirección coincida con la del Refresh Row Register,
- ▶  $t_{RFC}$  **Refresh Cycle time**. Intervalo de tiempo entre los comandos de Refresco y activación.

# Formato y timing de un comando Refresh.

Familia DRAM	Tensión	Capacidad	# bancos	# Filas	Tamaño Fila	Cuenta de Refresco	$t_{RC}$	$t_{RFC}$
DDR	2.5V	256Mb	4	8192	1kB	8192	60 ns	67 ns
		512Mb	4	8192	2kB	8192	55 ns	70 ns
DDR2	1.8V	256Mb	4	8192	1kB	8192	55 ns	75 ns
		512Mb	4	16384	1kB	8192	5 ns	105 ns
		1024Mb	8	16384	1kB	8192	54 ns	127.5 ns
		2048Mb	8	32768	1kB	8192	-	197.5 ns
		4096Mb	8	65536	1kB	8192	-	327.5 ns

- ▶ La tabla muestra tendencias en los tiempos de refresco.
- ▶ A mayor capacidad del banco mayor cantidad de celdas, mayor tiempo de refresco.
- ▶ Da la impresión que de DDR2 en adelante el challenge es mantener el número de comandos de refresco en períodos de 64 mseg a pesar de duplicarse en número de filas. Esto implica para el controlador de memoria enviar 8192 comandos Refresh en 64 mseg.
- ▶ Los devices de alta capacidad tienen mas de 8192 filas con 2, 4, u 8 en filas en cada device.

# Formato y timing de un comando Refresh.

Familia DRAM	Tensión	Capacidad	# bancos	# Filas	Tamaño Fila	Cuenta de Refresco	$t_{RC}$	$t_{RFC}$
DDR	2.5V	256Mb	4	8192	1kB	8192	60 ns	67 ns
		512Mb	4	8192	2kB	8192	55 ns	70 ns
DDR2	1.8V	256Mb	4	8192	1kB	8192	55 ns	75 ns
		512Mb	4	16384	1kB	8192	5 ns	105 ns
		1024Mb	8	16384	1kB	8192	54 ns	127.5 ns
		2048Mb	8	32768	1kB	8192	-	197.5 ns
		4096Mb	8	65536	1kB	8192	-	327.5 ns

- ▶ La tabla muestra tendencias en los tiempos de refresco.
- ▶ A mayor capacidad del banco mayor cantidad de celdas, mayor tiempo de refresco.
- ▶ Da la impresión que de DDR2 en adelante el challenge es mantener el número de comandos de refresco en períodos de 64 mseg a pesar de duplicarse en número de filas. Esto implica para el controlador de memoria enviar 8192 comandos Refresh en 64 mseg.
- ▶ Los devices de alta capacidad tienen mas de 8192 filas con 2, 4, u 8 en filas en cada device.

# Formato y timing de un comando Refresh.

Familia DRAM	Tensión	Capacidad	# bancos	# Filas	Tamaño Fila	Cuenta de Refresco	$t_{RC}$	$t_{RFC}$
DDR	2.5V	256Mb	4	8192	1kB	8192	60 ns	67 ns
		512Mb	4	8192	2kB	8192	55 ns	70 ns
DDR2	1.8V	256Mb	4	8192	1kB	8192	55 ns	75 ns
		512Mb	4	16384	1kB	8192	5 ns	105 ns
		1024Mb	8	16384	1kB	8192	54 ns	127.5 ns
		2048Mb	8	32768	1kB	8192	-	197.5 ns
		4096Mb	8	65536	1kB	8192	-	327.5 ns

- ▶ La tabla muestra tendencias en los tiempos de refresco.
- ▶ A mayor capacidad del banco mayor cantidad de celdas, mayor tiempo de refresco.
- ▶ Da la impresión que de DDR2 en adelante el challenge es mantener el número de comandos de refresco en períodos de 64 mseg a pesar de duplicarse en número de filas. Esto implica para el controlador de memoria enviar 8192 comandos Refresh en 64 mseg.
- ▶ Los devices de alta capacidad tienen mas de 8192 filas con 2, 4, u 8 en filas en cada device.

# Formato y timing de un comando Refresh.

Familia DRAM	Tensión	Capacidad	# bancos	# Filas	Tamaño Fila	Cuenta de Refresco	$t_{RC}$	$t_{RFC}$
DDR	2.5V	256Mb	4	8192	1kB	8192	60 ns	67 ns
		512Mb	4	8192	2kB	8192	55 ns	70 ns
DDR2	1.8V	256Mb	4	8192	1kB	8192	55 ns	75 ns
		512Mb	4	16384	1kB	8192	5 ns	105 ns
		1024Mb	8	16384	1kB	8192	54 ns	127.5 ns
		2048Mb	8	32768	1kB	8192	-	197.5 ns
		4096Mb	8	65536	1kB	8192	-	327.5 ns

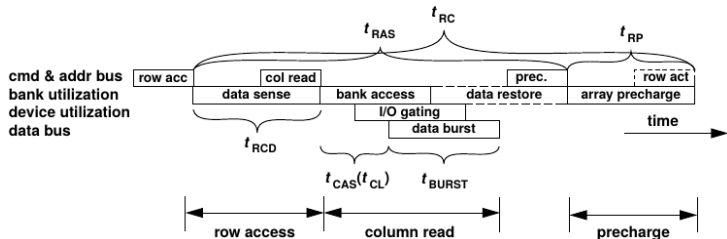
- ▶ La tabla muestra tendencias en los tiempos de refresco.
- ▶ A mayor capacidad del banco mayor cantidad de celdas, mayor tiempo de refresco.
- ▶ Da la impresión que de DDR2 en adelante el challenge es mantener el número de comandos de refresco en períodos de 64 mseg a pesar de duplicarse en número de filas. Esto implica para el controlador de memoria enviar 8192 comandos Refresh en 64 mseg.
- ▶ Los devices de alta capacidad tienen mas de 8192 filas con 2, 4, u 8 en filas en cada device.

# Formato y timing de un comando Refresh.

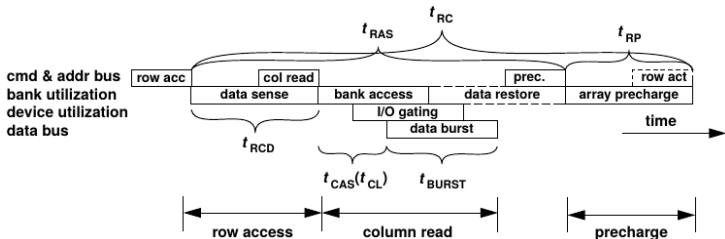
Familia DRAM	Tensión	Capacidad	# bancos	# Filas	Tamaño Fila	Cuenta de Refresco	$t_{RC}$	$t_{RFC}$
DDR	2.5V	256Mb	4	8192	1kB	8192	60 ns	67 ns
		512Mb	4	8192	2kB	8192	55 ns	70 ns
DDR2	1.8V	256Mb	4	8192	1kB	8192	55 ns	75 ns
		512Mb	4	16384	1kB	8192	5 ns	105 ns
		1024Mb	8	16384	1kB	8192	54 ns	127.5 ns
		2048Mb	8	32768	1kB	8192	-	197.5 ns
		4096Mb	8	65536	1kB	8192	-	327.5 ns

- ▶ La tabla muestra tendencias en los tiempos de refresco.
- ▶ A mayor capacidad del banco mayor cantidad de celdas, mayor tiempo de refresco.
- ▶ Da la impresión que de DDR2 en adelante el challenge es mantener el número de comandos de refresco en períodos de 64 mseg a pesar de duplicarse en número de filas. Esto implica para el controlador de memoria enviar 8192 comandos Refresh en 64 mseg.
- ▶ Los devices de alta capacidad tienen mas de 8192 filas con 2, 4, u 8 en filas en cada device.

# Ciclo de Lectura



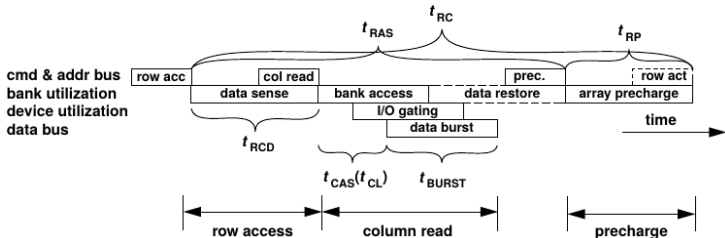
# Ciclo de Lectura



- ▶ El comando Row Access habilita muchos cientos de bits de datos al array de amplificadores de sensado dentro de un banco.
- ▶ Luego Column Access selecciona decenas o algunos pocos cientos de esos bits.
- ▶ Si el acceso a memoria se efectúa en direcciones secuenciales el device no necesita precargar nuevamente la fila, con el consiguiente ahorro de tiempo de acceso y de energía.
- ▶ Nuevamente el principio de vecindad en el acceso a memoria regula la eficiencia del sistema.

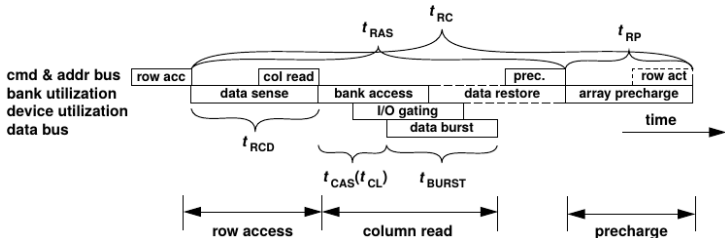


# Ciclo de Lectura



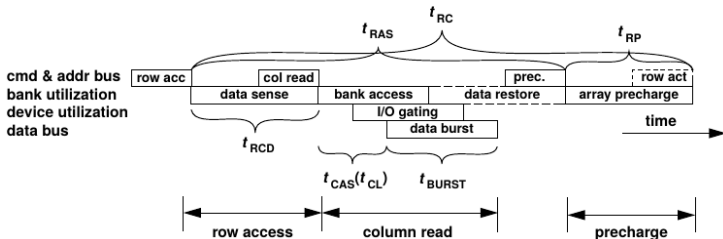
- ▶ El comando Row Access habilita muchos cientos de bits de datos al array de amplificadores de sensado dentro de un banco.
- ▶ Luego Column Access selecciona decenas o algunos pocos cientos de esos bits.
- ▶ Si el acceso a memoria se efectúa en direcciones secuenciales el device no necesita precargar nuevamente la fila, con el consiguiente ahorro de tiempo de acceso y de energía.
- ▶ Nuevamente el principio de vecindad en el acceso a memoria regula la eficiencia del sistema.

# Ciclo de Lectura



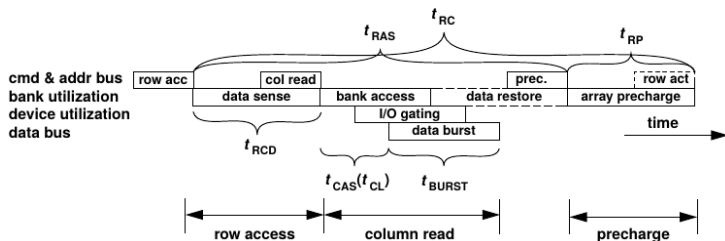
- ▶ El comando Row Access habilita muchos cientos de bits de datos al array de amplificadores de sensado dentro de un banco.
- ▶ Luego Column Access selecciona decenas o algunos pocos cientos de esos bits.
- ▶ Si el acceso a memoria se efectúa en direcciones secuenciales el device no necesita precargar nuevamente la fila, con el consiguiente ahorro de tiempo de acceso y de energía.
- ▶ Nuevamente el principio de vecindad en el acceso a memoria regula la eficiencia del sistema.

# Ciclo de Lectura



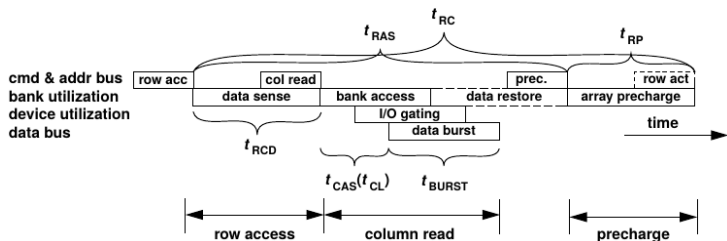
- ▶ El comando Row Access habilita muchos cientos de bits de datos al array de amplificadores de sensado dentro de un banco.
- ▶ Luego Column Access selecciona decenas o algunos pocos cientos de esos bits.
- ▶ Si el acceso a memoria se efectúa en direcciones secuenciales el device no necesita precargar nuevamente la fila, con el consiguiente ahorro de tiempo de acceso y de energía.
- ▶ Nuevamente el principio de vecindad en el acceso a memoria regula la eficiencia del sistema.

# Ciclo de Lectura



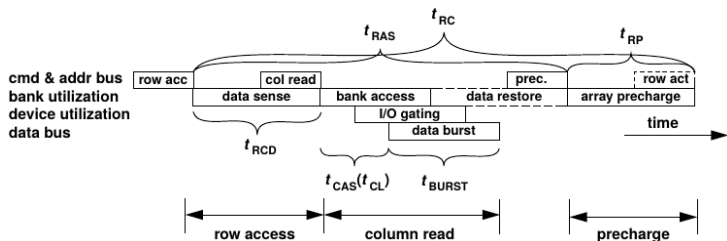
- ▶ Transcurrido  $t_{RCD}$  los datos de la fila seleccionada son precargados por los amplificadores de sensado.
- ▶ El controlador de memoria envía el comando Column Read y los datos se transfieren al bus y se recargan en las celdas.
- ▶ Transcurrido  $t_{RAS}$  las celdas están listas para un comando Precharge que resetee las bitlines y los amplificadores de sensado.

# Ciclo de Lectura



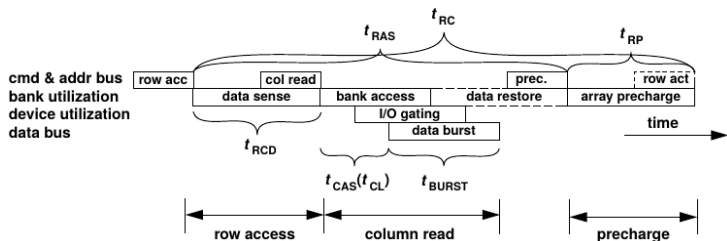
- ▶ Transcurrido  $t_{RCD}$  los datos de la fila seleccionada son precargados por los amplificadores de sensado.
- ▶ El controlador de memoria envía el comando Column Read y los datos se transfieren al bus y se recargan en las celdas.
- ▶ Transcurrido  $t_{RAS}$  las celdas están listas para un comando Precharge que resetee las bitlines y los amplificadores de sensado.

# Ciclo de Lectura



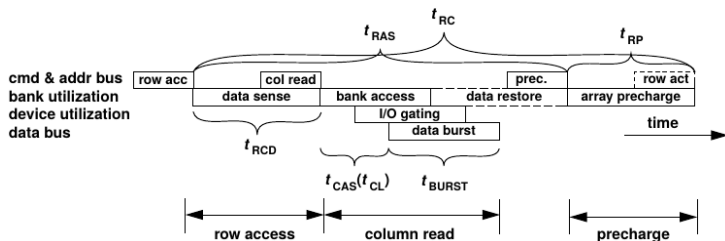
- ▶ Transcurrido  $t_{RCD}$  los datos de la fila seleccionada son precargados por los amplificadores de sensado.
- ▶ El controlador de memoria envía el comando Column Read y los datos se transfieren al bus y se recargan en las celdas.
- ▶ Transcurrido  $t_{RAS}$  las celdas están listas para un comando Precharge que resetee las bitlines y los amplificadores de sensado.

# Ciclo de Lectura



- ▶ Transcurrido  $t_{RCD}$  los datos de la fila seleccionada son precargados por los amplificadores de sensado.
- ▶ El controlador de memoria envía el comando Column Read y los datos se transfieren al bus y se recargan en las celdas.
- ▶ Transcurrido  $t_{RAS}$  las celdas están listas para un comando Precharge que resetee las bitlines y los amplificadores de sensado.

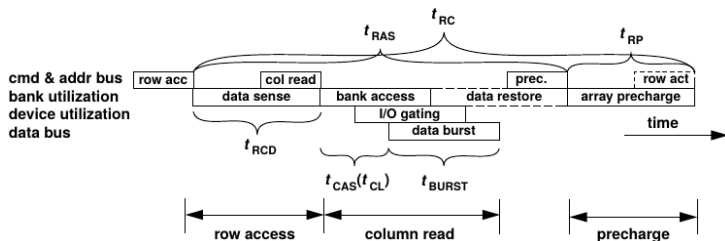
# Ciclo de Lectura



- Sistemas close-page: son los que precargan los bancos para acceder a una fila diferente luego del  $t_{RAS}$ .
- Sistemas open-page: son los que mantienen la fila activa y los amplificadores de sensado luego del  $t_{RAS}$  especulando con accesos secuenciales desde el procesador (principio de vecindad).

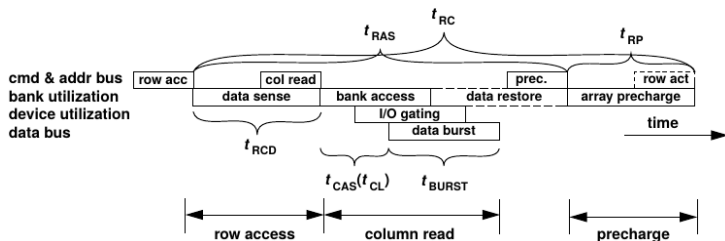


# Ciclo de Lectura



- **Sistemas close-page:** son los que precargan los bancos para acceder a una fila diferente luego del  $t_{RAS}$ .
- **Sistemas open-page:** son los que mantienen la fila activa y los amplificadores de sensado luego del  $t_{RAS}$  especulando con accesos secuenciales desde el procesador (principio de vecindad).

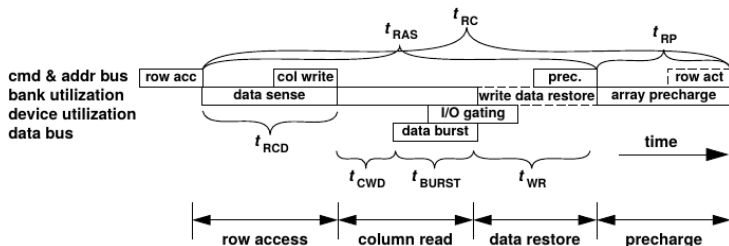
# Ciclo de Lectura



- ▶ **Sistemas close-page:** son los que precargan los bancos para acceder a una fila diferente luego del  $t_{RAS}$ .
- ▶ **Sistemas open-page:** son los que mantienen la fila activa y los amplificadores de sensado luego del  $t_{RAS}$  especulando con accesos secuenciales desde el procesador (principio de vecindad).

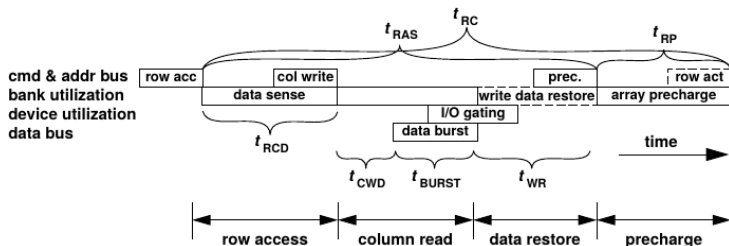


# Ciclo de Escritura



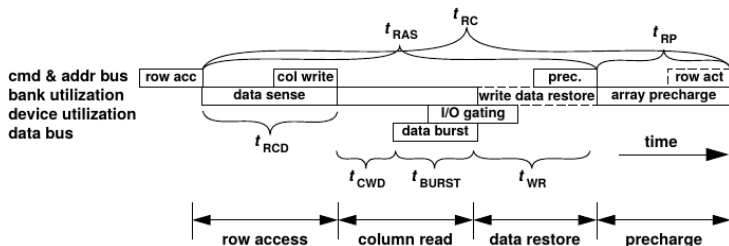
- ▶ En los dispositivos DRAM modernos el tiempo de ciclo de una fila está determinado por el tiempo de escritura.
- ▶ Es el tiempo que necesita un DRAM Device para proveer acceso a una fila en un determinado banco de celdas de DRAM.
- ▶ En el caso de un ciclo de escritura, los datos deben ser provistos por el controlador de memoria, viajan por el bus de datos, pasan por los multiplexores de disparo de E/S, saturan los amplificadores de sensado, y finalmente llegan a la celda.

# Ciclo de Escritura



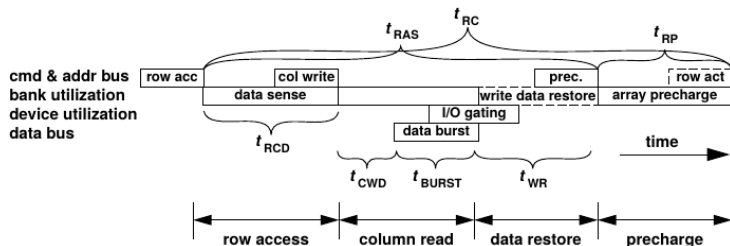
- ▶ En los dispositivos DRAM modernos el tiempo de ciclo de una fila está determinado por el tiempo de escritura.
- ▶ Es el tiempo que necesita un DRAM Device para proveer acceso a una fila en un determinado banco de celdas de DRAM.
- ▶ En el caso de un ciclo de escritura, los datos deben ser provistos por el controlador de memoria, viajan por el bus de datos, pasan por los multiplexores de disparo de E/S, saturan los amplificadores de sensado, y finalmente llegan a la celda.

# Ciclo de Escritura



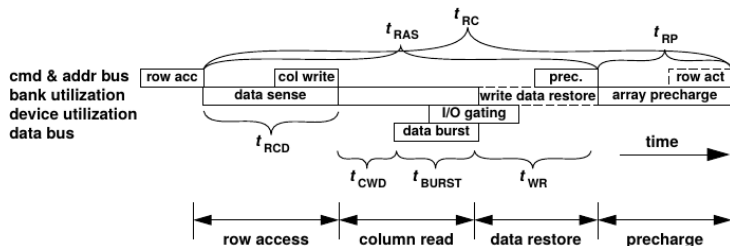
- ▶ En los dispositivos DRAM modernos el tiempo de ciclo de una fila está determinado por el tiempo de escritura.
- ▶ Es el tiempo que necesita un DRAM Device para proveer acceso a una fila en un determinado banco de celdas de DRAM.
- ▶ En el caso de un ciclo de escritura, los datos deben ser provistos por el controlador de memoria, viajan por el bus de datos, pasan por los multiplexores de disparo de E/S, saturan los amplificadores de sensado, y finalmente llegan a la celda.

# Ciclo de Escritura



- ▶ Esta secuencia se debe completar antes de enviar el comando de precarga que completa el ciclo de Escritura.
- ▶ De este modo el  $t_{RAS}$  debe ser suficientemente extenso como para abarcar  $t_{RCD}$ ,  $t_{CWD}$ ,  $t_{CCD}$  y  $t_{WR}$ .
- ▶ En la práctica debe setearse en un valor por lo menos igual a la suma de los otros cuatro.

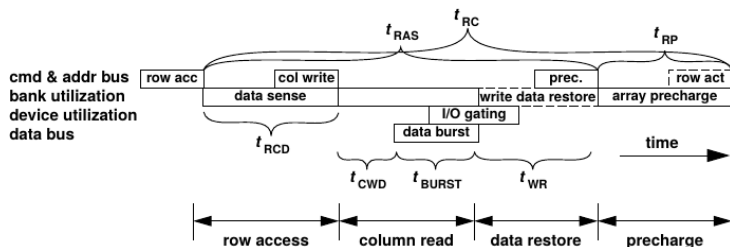
# Ciclo de Escritura



- ▶ Esta secuencia se debe completar antes de enviar el comando de precarga que completa el ciclo de Escritura.
- ▶ De este modo el  $t_{RAS}$  debe ser suficientemente extenso como para abarcar  $t_{RCD}$ ,  $t_{CWD}$ ,  $t_{CCD}$  y  $t_{WR}$ .
- ▶ En la práctica debe setearse en un valor por lo menos igual a la suma de los otros cuatro.

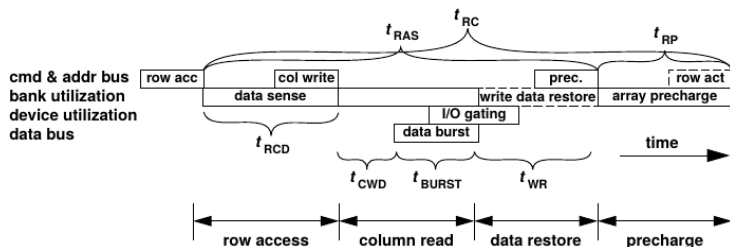


# Ciclo de Escritura



- ▶ Esta secuencia se debe completar antes de enviar el comando de precarga que completa el ciclo de Escritura.
- ▶ De este modo el  $t_{RAS}$  debe ser suficientemente extenso como para abarcar  $t_{RCD}$ ,  $t_{CWD}$ ,  $t_{CCD}$  y  $t_{WR}$ .
- ▶ En la práctica debe setearse en un valor por lo menos igual a la suma de los otros cuatro.

# Ciclo de Escritura



- ▶ Esta secuencia se debe completar antes de enviar el comando de precarga que completa el ciclo de Escritura.
- ▶ De este modo el  $t_{RAS}$  debe ser suficientemente extenso como para abarcar  $t_{RCD}$ ,  $t_{CWD}$ ,  $t_{CCD}$  y  $t_{WR}$ .
- ▶ En la práctica debe setearse en un valor por lo menos igual a la suma de los otros cuatro.

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- **Introducción General**
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Porque usamos un controlador de memoria

- ▶ Maneja la transferencia de datos hacia y desde la memoria asegurando la compatibilidad con el protocolo.
- ▶ Independiza al procesador de las características eléctricas de los DRAM Devices que conforman el sistema de memoria.
- ▶ Asegura el cumplimiento de los requerimientos de timing de los DRAM Devices independientemente del procesador base del sistema.
- ▶ Permite arbitrar el acceso a un único subsistema de memoria por parte de múltiples cores de procesamiento, aun si estos son heterogéneos.

# Porque usamos un controlador de memoria

- ▶ Maneja la transferencia de datos hacia y desde la memoria asegurando la compatibilidad con el protocolo.
- ▶ Independiza al procesador de las características eléctricas de los DRAM Devices que conforman el sistema de memoria.
- ▶ Asegura el cumplimiento de los requerimientos de timing de los DRAM Devices independientemente del procesador base del sistema.
- ▶ Permite arbitrar el acceso a un único subsistema de memoria por parte de múltiples cores de procesamiento, aun si estos son heterogéneos.

# Porque usamos un controlador de memoria

- ▶ Maneja la transferencia de datos hacia y desde la memoria asegurando la compatibilidad con el protocolo.
- ▶ Independiza al procesador de las características eléctricas de los DRAM Devices que conforman el sistema de memoria.
- ▶ Asegura el cumplimiento de los requerimientos de timing de los DRAM Devices independientemente del procesador base del sistema.
- ▶ Permite arbitrar el acceso a un único subsistema de memoria por parte de múltiples cores de procesamiento, aun si estos son heterogéneos.

# Porque usamos un controlador de memoria

- ▶ Maneja la transferencia de datos hacia y desde la memoria asegurando la compatibilidad con el protocolo.
- ▶ Independiza al procesador de las características eléctricas de los DRAM Devices que conforman el sistema de memoria.
- ▶ Asegura el cumplimiento de los requerimientos de timing de los DRAM Devices independientemente del procesador base del sistema.
- ▶ Permite arbitrar el acceso a un único subsistema de memoria por parte de múltiples cores de procesamiento, aun si estos son heterogéneos.

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

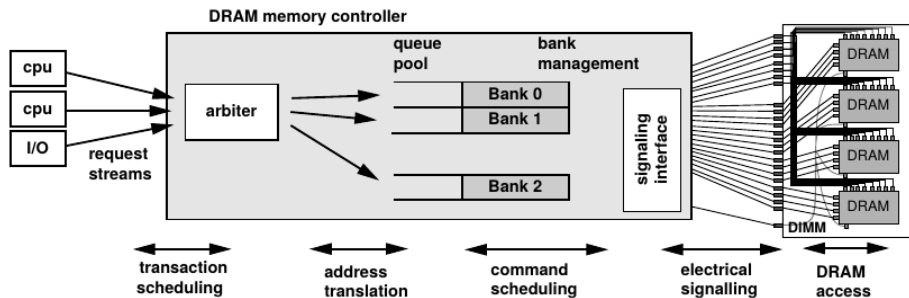
- Introducción General
- **Arquitectura**

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB



# Arquitectura de un Controlador Genérico



# Arbitración de requerimientos

- ▶ Los costos en acceso a memoria siguen aumentando relativamente los costos computacionales de muchos algoritmos.
- ▶ La arbitración de requerimientos es un aspecto crítico para la performance del sistema de memoria.
- ▶ Resuelve en base a las prioridades de los requerimientos de memoria de parte de los cores y de dispositivos de E/S con un DMA controller por ejemplo, cual es el que debe transmitirse a los DRAM devices.
- ▶ No siempre pasa el de mayor prioridad. Puede darse que un requerimiento de baja prioridad arribe al controlador en el momento en que la fila que contiene sus datos está activa. En tal caso algunos controladores resuelven primero este requerimiento a pesar de su prioridad menor en virtud de la performance general.

# Arbitración de requerimientos

- ▶ Los costos en acceso a memoria siguen aumentando relativamente los costos computacionales de muchos algoritmos.
- ▶ La arbitración de requerimientos es un aspecto crítico para la performance del sistema de memoria.
- ▶ Resuelve en base a las prioridades de los requerimientos de memoria de parte de los cores y de dispositivos de E/S con un DMA controller por ejemplo, cual es el que debe transmitirse a los DRAM devices.
- ▶ No siempre pasa el de mayor prioridad. Puede darse que un requerimiento de baja prioridad arribe al controlador en el momento en que la fila que contiene sus datos está activa. En tal caso algunos controladores resuelven primero este requerimiento a pesar de su prioridad menor en virtud de la performance general.

# Arbitración de requerimientos

- ▶ Los costos en acceso a memoria siguen aumentando relativamente los costos computacionales de muchos algoritmos.
- ▶ La arbitración de requerimientos es un aspecto crítico para la performance del sistema de memoria.
- ▶ Resuelve en base a las prioridades de los requerimientos de memoria de parte de los cores y de dispositivos de E/S con un DMA controller por ejemplo, cual es el que debe transmitirse a los DRAM devices.
- ▶ No siempre pasa el de mayor prioridad. Puede darse que un requerimiento de baja prioridad arribe al controlador en el momento en que la fila que contiene sus datos está activa. En tal caso algunos controladores resuelven primero este requerimiento a pesar de su prioridad menor en virtud de la performance general.

# Arbitración de requerimientos

- ▶ Los costos en acceso a memoria siguen aumentando relativamente los costos computacionales de muchos algoritmos.
- ▶ La arbitración de requerimientos es un aspecto crítico para la performance del sistema de memoria.
- ▶ Resuelve en base a las prioridades de los requerimientos de memoria de parte de los cores y de dispositivos de E/S con un DMA controller por ejemplo, cual es el que debe transmitirse a los DRAM devices.
- ▶ No siempre pasa el de mayor prioridad. Puede darse que un requerimiento de baja prioridad arribe al controlador en el momento en que la fila que contiene sus datos está activa. En tal caso algunos controladores resuelven primero este requerimiento a pesar de su prioridad menor en virtud de la performance general.

# Arbitración de requerimientos

- ▶ Los costos en acceso a memoria siguen aumentando relativamente los costos computacionales de muchos algoritmos.
- ▶ La arbitración de requerimientos es un aspecto crítico para la performance del sistema de memoria.
- ▶ Resuelve en base a las prioridades de los requerimientos de memoria de parte de los cores y de dispositivos de E/S con un DMA controller por ejemplo, cual es el que debe transmitirse a los DRAM devices.
- ▶ No siempre pasa el de mayor prioridad. Puede darse que un requerimiento de baja prioridad arribe al controlador en el momento en que la fila que contiene sus datos está activa. En tal caso algunos controladores resuelven primero este requerimiento a pesar de su prioridad menor en virtud de la performance general.

# Traducción de Direcciones

- ▶ Una vez que una transacción ganó la arbitración pasa al controlador en donde se traduce la dirección física que proviene del bus del procesador, a la dirección del Bus del controlador de memoria: Row & Column.
- ▶ Además en función de la operación requerida desde el core o el device de E/S sobre esa dirección la etapa de traducción arma la secuencia de comandos necesaria para enviar a la DRAM de acuerdo al protocolo descripto.
- ▶ Una vez enviados a los DRAM devices, si éstos son relativamente modernos dispondrán de buffers de comandos como etapa previa a los amplificadores de sensado.
- ▶ La política de buffering es crítica en el diseño de cada DRAM Device en virtud de su efecto sobre la performance, o sobre el consumo. El análisis del trade on y trade off de estos aspectos son moneda corriente entre los diseñadores de estos dispositivos..

# Traducción de Direcciones

- ▶ Una vez que una transacción ganó la arbitración pasa al controlador en donde se traduce la dirección física que proviene del bus del procesador, a la dirección del Bus del controlador de memoria: Row & Column.
- ▶ Además en función de la operación requerida desde el core o el device de E/S sobre esa dirección la etapa de traducción arma la secuencia de comandos necesaria para enviar a la DRAM de acuerdo al protocolo descrito.
- ▶ Una vez enviados a los DRAM devices, si éstos son relativamente modernos dispondrán de buffers de comandos como etapa previa a los amplificadores de sensado.
- ▶ La política de buffering es crítica en el diseño de cada DRAM Device en virtud de su efecto sobre la performance, o sobre el consumo. El análisis del trade on y trade off de estos aspectos son moneda corriente entre los diseñadores de estos dispositivos.



# Traducción de Direcciones

- ▶ Una vez que una transacción ganó la arbitración pasa al controlador en donde se traduce la dirección física que proviene del bus del procesador, a la dirección del Bus del controlador de memoria: Row & Column.
- ▶ Además en función de la operación requerida desde el core o el device de E/S sobre esa dirección la etapa de traducción arma la secuencia de comandos necesaria para enviar a la DRAM de acuerdo al protocolo descripto.
- ▶ Una vez enviados a los DRAM devices, si éstos son relativamente modernos dispondrán de buffers de comandos como etapa previa a los amplificadores de sensado.
- ▶ La política de buffering es crítica en el diseño de cada DRAM Device en virtud de su efecto sobre la performance, o sobre el consumo. El análisis del trade on y trade off de estos aspectos son moneda corriente entre los diseñadores de estos dispositivos..

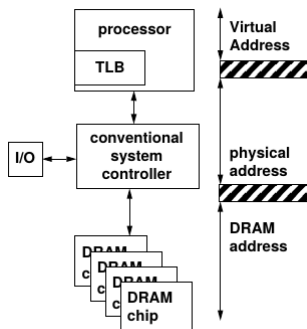
# Traducción de Direcciones

- ▶ Una vez que una transacción ganó la arbitración pasa al controlador en donde se traduce la dirección física que proviene del bus del procesador, a la dirección del Bus del controlador de memoria: Row & Column.
- ▶ Además en función de la operación requerida desde el core o el device de E/S sobre esa dirección la etapa de traducción arma la secuencia de comandos necesaria para enviar a la DRAM de acuerdo al protocolo descripto.
- ▶ Una vez enviados a los DRAM devices, si éstos son relativamente modernos dispondrán de buffers de comandos como etapa previa a los amplificadores de sensado.
- ▶ La política de buffering es crítica en el diseño de cada DRAM Device en virtud de su efecto sobre la performance, o sobre el consumo. El análisis del trade on y trade off de estos aspectos son moneda corriente entre los diseñadores de estos dispositivos..

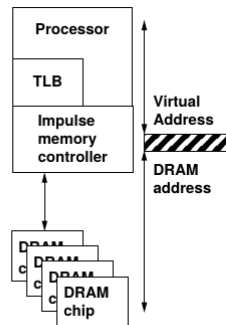
# Traducción de Direcciones

- ▶ Una vez que una transacción ganó la arbitración pasa al controlador en donde se traduce la dirección física que proviene del bus del procesador, a la dirección del Bus del controlador de memoria: Row & Column.
- ▶ Además en función de la operación requerida desde el core o el device de E/S sobre esa dirección la etapa de traducción arma la secuencia de comandos necesaria para enviar a la DRAM de acuerdo al protocolo descripto.
- ▶ Una vez enviados a los DRAM devices, si éstos son relativamente modernos dispondrán de buffers de comandos como etapa previa a los amplificadores de sensado.
- ▶ La política de buffering es crítica en el diseño de cada DRAM Device en virtud de su efecto sobre la performance, o sobre el consumo. El análisis del trade on y trade off de estos aspectos son moneda corriente entre los diseñadores de estos dispositivos..

# Traducción de direcciones



Conventional System Architecture



Impulse Memory System Architecture

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

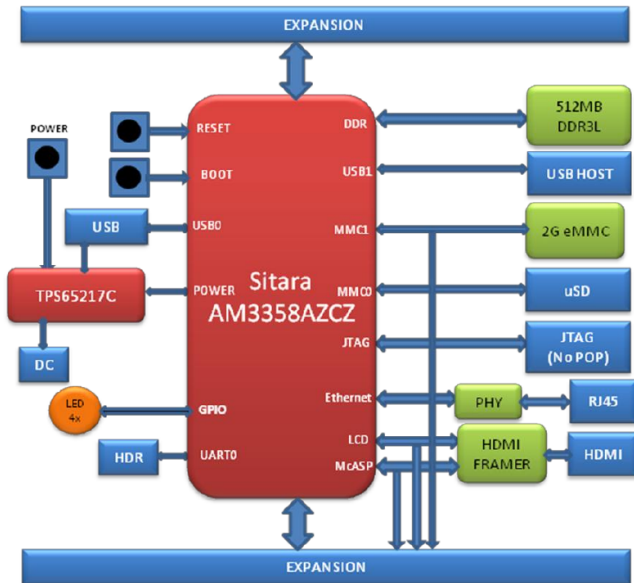
## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- **Beagle Bone Black**
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB

# Diagrama General de la BBB



# Subsistema de Memoria

- ▶ **512MB DDR3L** Utiliza un solo device DRAM de 256Mbx16 DDR3L 4Gb (512MB). El chip de memoria es el MT41K256M16HA-125 de Micron. Opera a 400MHz manteniendo una velocidad efectiva de 800MHZ sobre el bus DDR3L, alcanzando así un ancho de banda de memoria de 1.6GB/S.
- ▶ **4KB EEPROM** Mantiene la información del board: board name, serial number, revisión, etc. .
- ▶ **2GB Embedded MMC** Se conecta por default al port MMC1 del procesador, permitiendo accesos de 8 bits de ancho. Se puede cambiar al port MMC0, al slot de la SD.

# Subsistema de Memoria

- ▶ **512MB DDR3L** Utiliza un solo device DRAM de 256Mbx16 DDR3L 4Gb (512MB). El chip de memoria es el MT41K256M16HA-125 de Micron. Opera a 400MHz manteniendo una velocidad efectiva de 800MHZ sobre el bus DDR3L, alcanzando así un ancho de banda de memoria de 1.6GB/S.
- ▶ **4KB EEPROM** Mantiene la información del board: board name, serial number, revisión, etc. .
- ▶ **2GB Embedded MMC** Se conecta por default al port MMC1 del procesador, permitiendo accesos de 8 bits de ancho. Se puede cambiar al port MMC0, al slot de la SD.



# Subsistema de Memoria

- ▶ **512MB DDR3L** Utiliza un solo device DRAM de 256Mbx16 DDR3L 4Gb (512MB). El chip de memoria es el MT41K256M16HA-125 de Micron. Opera a 400MHz manteniendo una velocidad efectiva de 800MHZ sobre el bus DDR3L, alcanzando así un ancho de banda de memoria de 1.6GB/S.
- ▶ **4KB EEPROM** Mantiene la información del board: board name, serial number, revisión, etc. .
- ▶ **2GB Embedded MMC** Se conecta por default al port MMC1 del procesador, permitiendo accesos de 8 bits de ancho. Se puede cambiar al port MMC0, al slot de la SD.

# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

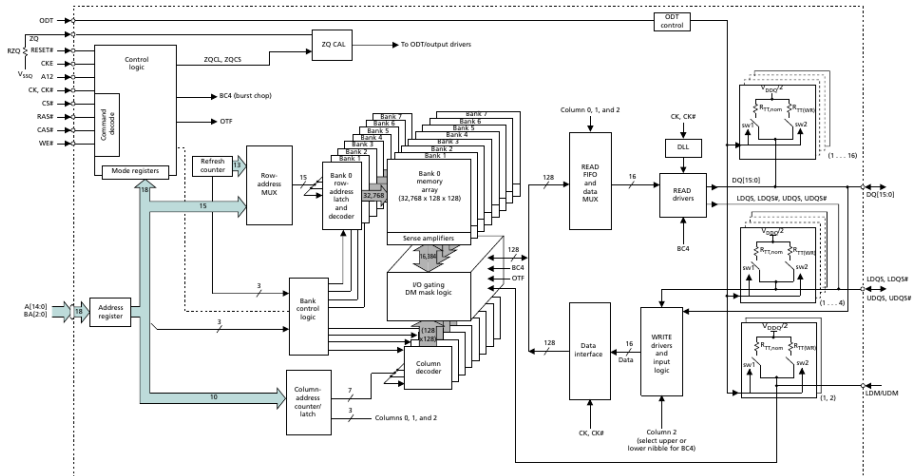
## 6 Controladores de Memoria

- Introducción General
- Arquitectura

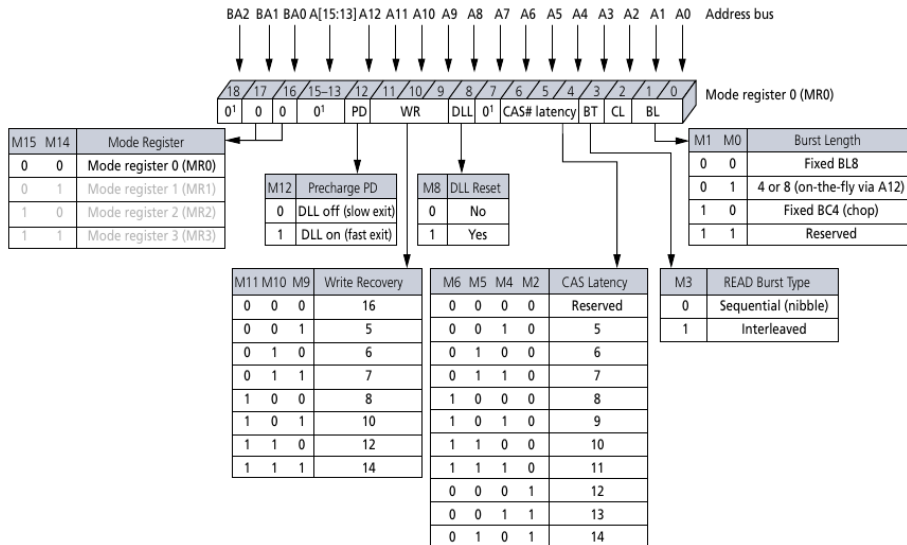
## 7 Casos Prácticos

- Beagle Bone Black
- **Memorias DDR en la BBB**
- Controlador de DDRn SDRAM en la BBB

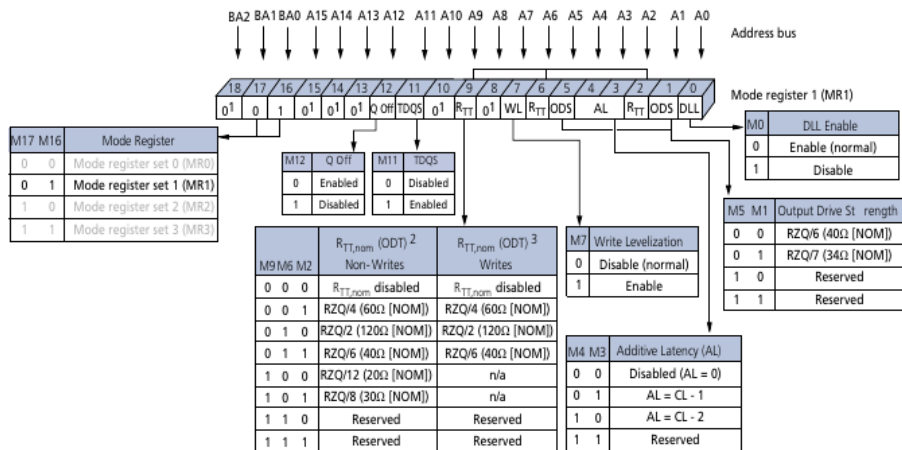
# Organización de la DRAM MT41K256M16 de Micron



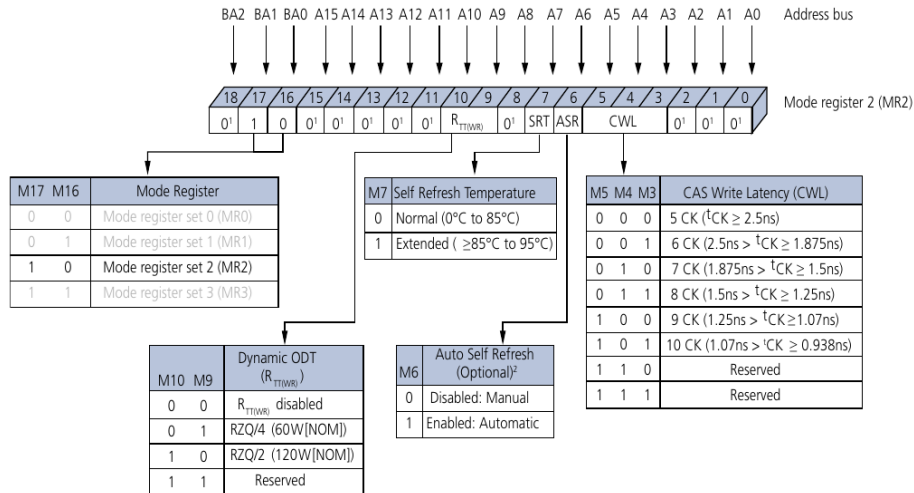
# Registro de Modo 0 del MT41K256M16



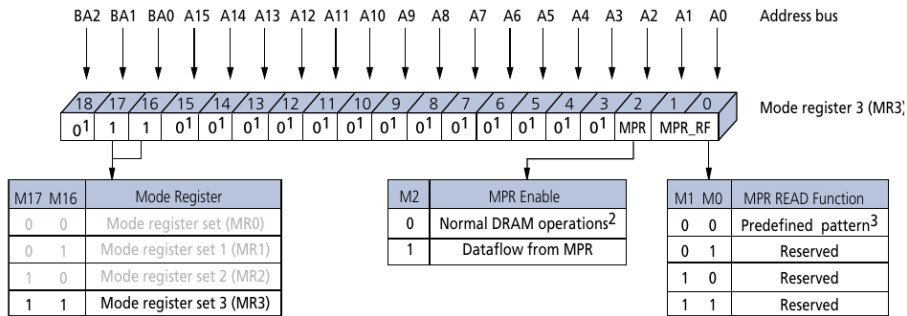
# Registro de Modo 1 del MT41K256M16



# Registro de Modo 2 del MT41K256M16



# Registro de Modo 3 del MT41K256M16



# Temario

## 1 Memorias Dinámicas

- Introducción
- Organización interna

## 2 Arquitecturas DRAM

- Evolución
- Throughput + Latency

## 3 Standards

- Estado del arte
- JEDEC SDRAM
- JEDEC DDR SDRAM

## 4 Configuración

- Configuración del DRAM Device

## 5 Protocolo de acceso

- Fundamentos básicos

## 6 Controladores de Memoria

- Introducción General
- Arquitectura

## 7 Casos Prácticos

- Beagle Bone Black
- Memorias DDR en la BBB
- Controlador de DDRn SDRAM en la BBB



# Registros Memory Mapped

- ▶ En general los Cores ARM tienen E/S memmory mapped.
- ▶ El controlador de memoria (EMIF) tiene un set de registros de 32 bits mapeados a partir de la dirección 0x4C000000)
- ▶ En rigor el espacio asignado al controlador es 0x4C000000 a 0x4CFF

# Registros Memory Mapped

- ▶ En general los Cores ARM tienen E/S memmory mapped.
- ▶ El controlador de memoria (EMIF) tiene un set de registros de 32 bits mapeados a partir de la dirección 0x4C000000)
- ▶ En rigor el espacio asignado al controlador es 0x4C000000 a 0x4CFF

# Registros Memory Mapped

- ▶ En general los Cores ARM tienen E/S memmory mapped.
- ▶ El controlador de memoria (EMIF) tiene un set de registros de 32 bits mapeados a partir de la dirección 0x4C000000)
- ▶ En rigor el espacio asignado al controlador es 0x4C000000 a 0x4CFF

# Registros Memory Mapped

- ▶ En general los Cores ARM tienen E/S memmory mapped.
- ▶ El controlador de memoria (EMIF) tiene un set de registros de 32 bits mapeados a partir de la dirección 0x4C000000)
- ▶ En rigor el espacio asignado al controlador es 0x4C000000 a 0x4CFF

# Cálculo de los valores a programar

	A	B	C	D	E	F	G	H
1	<b>AM335x DDR3 Timing Configuration Tool</b>							
2								
3		AM335x register name	AM335x register bit length	Memory datasheet symbol	Memory Datasheet value	unit	AM335x Setting (Decimal)	Comments
4		ICK		ICK	3.3	ns		
5	SDRAM_TIM_1	REG_T_RP	4	IRP	13.5	ns	4	typically taken from the speed bin tables
6		REG_T_RCD	4	IRCD	13.5	ns	4	typically taken from the speed bin tables
7		REG_T_WR	4	IVR	15	ns	4	
8		REG_T_RAS	5	IRAS	36	ns	10	IRAS should be >= IRCD
9		REG_T_RC	6	IRC	49.5	ns	14	
10		REG_T_RRD	3	IRRD	4	tCK	3	use the value given in CK units
11	SDRAM_TIM_2	REG_T_WTR	3	IVTR	4	tCK	3	use the value given in CK units
12		REG_T_0P	3	ICP	3	tCK	2	use the value given in CK units
13		REG_T_ODT	3	ODTLon	3	tCK	3	typically in terms of CWL. First determine CWL.
14		REG_T_XSNR	9	IXS	170	ns	511	usually IRFC+10
15		REG_T_XSRD	10	IXSDLL	512	tCK	511	usually in terms of tDLLK
16		REG_T_RTP	3	IRTP	4	tCK	3	use the value given in CK units
17	SDRAM_TIM_3	REG_T_CKE	3	ICKE	3	tCK	2	use the value given in CK units
18		REG_T_PDLL_UL	4				5	set to fixed value of 5
19		REG_T_ZQCS	6	IZQCS	64	tCK	63	
20		REG_T_RFC	9	IRFC	160	ns	48	
21				IREF				
22		REG_T_RAS_MAX	4	IRASmax			15	for DDR3, must be set to 15
23								
24								

## Herramienta de configuración

Generalmente los fabricantes proporcionan una herramienta para calcular los valores a setear en los campos de bits de los registros del controlador de memoria. En este caso, en la columna coloreada en amarillo se ingresan los valores de temporización que figuran en la hoja de datos de la memoria que se decida colocar, y se calcula automáticamente el valor a programar para esa memoria.

# Cálculo de los valores a programar

AM335x DDR3 Timing Configuration Tool							
	AM335x register name	AM335x register bit length	Memory datasheet symbol	Memory Datasheet value	unit	AM335x Setting (Decimal)	Comments
	ICK		ICK	3.3	ns		
	REG_T_RP	4	IRP	13.5	ns		typically taken from the speed bin tables
	REG_T_RCD	4	IRCD	13.5	ns		typically taken from the speed bin tables
	REG_T_WR	4	IWR	15	ns		
	REG_T_RAS	5	IRAS	36	ns		IRAS should be >= IRCD
	REG_T_RC	6	IRC	49.5	ns		
	REG_T_RRD	3	IRRD	4	tCK		use the value given in CK units
	REG_T_WTR	3	IWTR	4	tCK		use the value given in CK units
	REG_T_XP	3	IXP	3	tCK		use the value given in CK units
	REG_T_ODT	3	ODTLen		tCK		typically in terms of CWL. First determine CWL
	REG_T_XSNR	9	IXS	170	ns		usually IRFC+10
	REG_T_XSRD	10	IXSDLL	512	tCK		usually in terms of tDLLK
	REG_T_RTP	3	IRTP	4	tCK		use the value given in CK units
	REG_T_CKE	3	ICKE	3	tCK		use the value given in CK units
	REG_T_PDLL_UL	4					set to fixed value of 5
	REG_T_ZQCS	6	IZQCS	64	tCK		
	REG_T_RFC	9	IRFC	160	ns		
			IREF				
	REG_T_RAS_MAX	4	IRASmax				for DDR3, must be set to 15

## Herramienta de configuración

Generalmente los fabricantes proporcionan una herramienta para calcular los valores a setear en los campos de bits de los registros del controlador de memoria. En este caso, En la columna coloreada en amarillo se ingresan los valores de temporización que figuran en la hoja de datos de la memoria que se decida colocar, y se calcula automáticamente el valor a programar para esa memoria.

# Tiempos

I <sub>DD</sub> Parameter	DDR3L -800		DDR3L -1066		DDR3L -1333		DDR3L -1600		DDR3L -1866	DDR3L -2133	Unit
	-25E	-25	-187E	-187	-15E	-15	-125E	-125	-107	-093	
	5-5-5	6-6-6	7-7-7	8-8-8	9-9-9	10-10-10	10-10-10	11-11-11	13-13-13	14-14-14	
t <sub>CK</sub> (MIN) I <sub>DD</sub>	2.5		1.875		1.5		1.25		1.07	0.938	ns
CL I <sub>DD</sub>	5	6	7	8	9	10	10	11	13	14	CK
t <sub>RCD</sub> (MIN) I <sub>DD</sub>	5	6	7	8	9	10	10	11	13	14	CK
t <sub>RC</sub> (MIN) I <sub>DD</sub>	20	21	27	28	33	34	38	39	45	50	CK
t <sub>RAS</sub> (MIN) I <sub>DD</sub>	15	15	20	20	24	24	28	28	32	36	CK
t <sub>RP</sub> (MIN)	5	6	7	8	9	10	10	11	13	14	CK
t <sub>FAW</sub>	x4, x8	16	16	20	20	20	24	24	26	27	CK
	x16	20	20	27	27	30	30	32	32	33	CK
t <sub>RRD</sub> I <sub>DD</sub>	x4, x8	4	4	4	4	4	5	5	5	6	CK
	x16	4	4	6	6	5	5	6	6	7	CK
t <sub>RFC</sub>	1Gb	44	44	59	59	74	74	88	88	103	CK
	2Gb	64	64	86	86	107	107	128	128	150	CK
	4Gb	104	104	139	139	174	174	208	208	243	CK
	8Gb	140	140	187	187	234	234	280	280	328	CK

# Tiempos

## Electrical Characteristics and AC Operating Conditions

Parameter	Symbol	DDR3L-800		DDR3L-1066		DDR3L-1333		DDR3L-1600		Unit	Notes
		Min	Max	Min	Max	Min	Max	Min	Max		
DQS, DQS# Low-Z time (RL - 1)	<sup>1</sup> LZDQS	-800	400	-600	300	-500	250	-450	225	ps	22, 23
DQS, DQS# High-Z time (RL + BL/2)	<sup>1</sup> HZDQS	-	400	-	300	-	250	-	225	ps	22, 23
DQS, DQS# differential READ preamble	<sup>1</sup> RPRE	0.9	Note 24	0.9	Note 24	0.9	Note 24	0.9	Note 24	CK	23, 24
DQS, DQS# differential READ postamble	<sup>1</sup> RPST	0.3	Note 27	0.3	Note 27	0.3	Note 27	0.3	Note 27	CK	23, 27
Command and Address Timing											
DLL locking time	<sup>1</sup> DLK	512	-	512	-	512	-	512	-	CK	28
CTRL, CMD, ADDR setup to CK, CK#	Base (specification)	<sup>1</sup> IS (AC160)	215	-	140	-	80	-	60	ps	29, 30, 44
	V <sub>REF</sub> @ 1 V/ns		375	-	300	-	240	-	220	ps	20, 30
CTRL, CMD, ADDR setup to CK, CK#	Base (specification)	<sup>1</sup> IS (AC135)	365	-	290	-	205	-	185	ps	29, 30, 44
	V <sub>REF</sub> @ 1 V/ns		500	-	425	-	340	-	320	ps	20, 30
CTRL, CMD, ADDR setup to CK, CK#	Base (specification)	<sup>1</sup> IH (DC90)	285	-	210	-	150	-	130	ps	29, 30, 44
	V <sub>REF</sub> @ 1 V/ns		375	-	300	-	240	-	220	ps	20, 30
Minimum CTRL, CMD, ADDR pulse width	<sup>1</sup> PW	900	-	780	-	620	-	560	-	ps	41
ACTIVATE to internal READ or WRITE delay	<sup>1</sup> RCD	See Speed Bin Tables for <sup>1</sup> RCD									ns 31
PRECHARGE command period	<sup>1</sup> RP	See Speed Bin Tables for <sup>1</sup> RP									ns 31
ACTIVATE-to-PRECHARGE command period	<sup>1</sup> RAS	See Speed Bin Tables for <sup>1</sup> RAS									ns 31, 32
ACTIVATE-to-ACTIVATE command period	<sup>1</sup> RC	See Speed Bin Tables for <sup>1</sup> RC									ns 31, 43
ACTIVATE-to-ACTIVATE minimum command period	x4/x8 (1KB page size)	<sup>1</sup> RRD	MIN = greater of 4CK or 10ns		MIN = greater of 4CK or 7.5ns		MIN = greater of 4CK or 6ns		MIN = greater of 4CK or 6ns		CK 31
	x16 (2KB page size)		MIN = greater of 4CK or 10ns		MIN = greater of 4CK or 7.5ns						CK 31
Four ACTIVATE windows	x4/x8 (1KB page size)	<sup>1</sup> FAW	40	-	37.5	-	30	-	30	-	ns 31
	x16 (2KB page size)		50	-	50	-	45	-	40	-	ns 31
Write recovery time	<sup>1</sup> WR	MIN = 15ns; MAX = N/A									ns 31, 32, 33, 34
Delay from start of internal WRITE transaction to internal READ command	<sup>1</sup> WTR	MIN = greater of 4CK or 7.5ns; MAX = N/A									CK 31, 34
READ-to-PRECHARGE time	<sup>1</sup> RTP	MIN = greater of 4CK or 7.5ns; MAX = N/A									CK 31, 32



# Tiempos

## Electrical Characteristics and AC Operating Conditions

Parameter		Symbol	DDR3L-800		DDR3L-1066		DDR3L-1333		DDR3L-1600		Unit	Notes
			Min	Max	Min	Max	Min	Max	Min	Max		
WRITE with auto precharge command to power-down entry	BL8 (OTF, MRS)	<sup>1</sup> WRAP-DEN	MIN = WL + 4 + WR + 1								CK	
	BC4OTF											
	BC4MRS	<sup>1</sup> WRAP-DEN	MIN = WL + 2 + WR + 1								CK	
Power-Down Exit Timing												
DLL on, any valid command, or DLL off to commands not requiring locked DLL		<sup>1</sup> XP	MIN = greater of 3CK or 7.5ns; MAX = N/A				MIN = greater of 3CK or 6ns; MAX = N/A				CK	
Precharge power-down with DLL off to commands requiring a locked DLL		<sup>1</sup> XPDLL	MIN = greater of 10CK or 24ns; MAX = N/A								CK	28
ODT Timing												
R <sub>TT</sub> synchronous turn-on delay		ODTLon	CWL + AL - 2CK								CK	38
R <sub>TT</sub> synchronous turn-off delay		ODTLoff	CWL + AL - 2CK								CK	40
R <sub>TT</sub> turn-on from ODTL on reference		<sup>1</sup> AON	-400	400	-300	300	-250	250	-225	225	ps	23, 38
R <sub>TT</sub> turn-off from ODTL off reference		<sup>1</sup> AOF	0.3	0.7	0.3	0.7	0.3	0.7	0.3	0.7	CK	39, 40
Asynchronous R <sub>TT</sub> turn-on delay (power-down with DLL off)		<sup>1</sup> AONPD	MIN = 2; MAX = 8.5								ns	38
Asynchronous R <sub>TT</sub> turn-off delay (power-down with DLL off)		<sup>1</sup> AOFPD	MIN = 2; MAX = 8.5								ns	40
ODT HIGH time with WRITE command and BL8		ODTH8	MIN = 6; MAX = N/A								CK	
ODT HIGH time without WRITE command or with WRITE command and BC4		ODTH4	MIN = 4; MAX = N/A								CK	
Dynamic ODT Timing												
R <sub>TT(nom)</sub> to R <sub>TT(WR)</sub> change skew		ODTLcnw	WL - 2CK								CK	
R <sub>TT(WR)</sub> to R <sub>TT(nom)</sub> change skew - BC4		ODTLcnw4	4CK + ODTLoff								CK	
R <sub>TT(WR)</sub> to R <sub>TT(nom)</sub> change skew - BL8		ODTLcnw8	6CK + ODTLoff								CK	
R <sub>TT</sub> dynamic change skew		<sup>1</sup> ADC	0.3	0.7	0.3	0.7	0.3	0.7	0.3	0.7	CK	39
Write Leveling Timing												
First DQS, DQS# rising edge		<sup>1</sup> WLMRD	40	-	40	-	40	-	40	-	CK	
DQS, DQS# delay		<sup>1</sup> WLDQSEN	25	-	25	-	25	-	25	-	CK	
Write leveling setup from rising CK, CK# crossing to rising DQS, DQS# crossing		<sup>1</sup> WLS	325	-	245	-	195	-	165	-	ps	

# Tiempos

## Electrical Characteristics and AC Operating Conditions

Parameter		Symbol	DDR3L-800		DDR3L-1066		DDR3L-1333		DDR3L-1600		Unit	Notes
			Min	Max	Min	Max	Min	Max	Min	Max		
CAS#-to-CAS# command delay		<sup>1</sup> t <sub>CCD</sub>	MIN = 4CK; MAX = N/A								CK	
Auto precharge write recovery + precharge time		<sup>1</sup> t <sub>DAL</sub>	MIN = WR + <sup>1</sup> t <sub>RP</sub> /CK (AVG); MAX = N/A								CK	
MODE REGISTER SET command cycle time		<sup>1</sup> t <sub>MRD</sub>	MIN = 4CK; MAX = N/A								CK	
MODE REGISTER SET command update delay		<sup>1</sup> t <sub>MOD</sub>	MIN = greater of 12CK or 15ns; MAX = N/A								CK	
MULTIPURPOSE REGISTER READ burst end to mode register set for multipurpose register exit		<sup>1</sup> t <sub>MPRR</sub>	MIN = 1CK; MAX = N/A								CK	
Calibration Timing												
ZQCL command: Long calibration time	POWER-UP and RE-SET operation	<sup>1</sup> t <sub>Zqinit</sub>	512	–	512	–	512	–	512	–	CK	
	Normal operation	<sup>1</sup> t <sub>Zqoper</sub>	256	–	256	–	256	–	256	–	CK	
ZQCS command: Short calibration time		<sup>1</sup> t <sub>ZQCS</sub>	64	–	64	–	64	–	64	–	CK	
Initialization and Reset Timing												
Exit reset from CKE HIGH to a valid command		<sup>1</sup> t <sub>XPR</sub>	MIN = greater of 5CK or <sup>1</sup> t <sub>RFC</sub> + 10ns; MAX = N/A								CK	
Begin power supply ramp to power supplies stable		<sup>1</sup> t <sub>VDDPR</sub>	MIN = N/A; MAX = 200								ms	
RESET# LOW to power supplies stable		<sup>1</sup> t <sub>RPS</sub>	MIN = 0; MAX = 200								ms	
RESET# LOW to I/O and R <sub>TT</sub> High-Z		<sup>1</sup> t <sub>IOZ</sub>	MIN = N/A; MAX = 20								ns	35
Refresh Timing												
REFRESH-to-ACTIVATE or REFRESH command period		<sup>1</sup> t <sub>RFC</sub> – 1Gb	MIN = 110; MAX = 70,200								ns	
		<sup>1</sup> t <sub>RFC</sub> – 2Gb	MIN = 160; MAX = 70,200								ns	
		<sup>1</sup> t <sub>RFC</sub> – 4Gb	MIN = 260; MAX = 70,200								ns	
		<sup>1</sup> t <sub>RFC</sub> – 8Gb	MIN = 350; MAX = 70,200								ns	
Maximum refresh period	T <sub>C</sub> ≤ 85°C	–	64 (1X)								ms	36
	T <sub>C</sub> > 85°C	–	32 (2X)								ms	36
Maximum average periodic refresh	T <sub>C</sub> ≤ 85°C	<sup>1</sup> t <sub>REFI</sub>	7.8 (64ms/8192)								µs	36
	T <sub>C</sub> > 85°C	–	3.9 (32ms/8192)								µs	36
Self Refresh Timing												
Exit self refresh to commands not requiring locked DLL		<sup>1</sup> t <sub>XS</sub>	MIN = greater of 5CK or <sup>1</sup> t <sub>RFC</sub> + 10ns; MAX = N/A								CK	

# Tiempos

## Electrical Characteristics and AC Operating Conditions

Parameter		Symbol	DDR3L-800		DDR3L-1066		DDR3L-1333		DDR3L-1600		Unit	Notes
			Min	Max	Min	Max	Min	Max	Min	Max		
Exit self refresh to commands requiring a locked DLL		<b><sup>1</sup>XSDLL</b>	MIN = <sup>1</sup> DLLK (MIN); MAX = N/A								CK	28
Minimum CKE low pulse width for self refresh entry to self refresh exit timing		<sup>1</sup> CKESR	MIN = <sup>1</sup> CKE (MIN) + CK; MAX = N/A								CK	
Valid clocks after self refresh entry or power-down entry		<sup>1</sup> CKSRE	MIN = greater of 5CK or 10ns; MAX = N/A								CK	
Valid clocks before self refresh exit, power-down exit, or reset exit		<sup>1</sup> CKSRX	MIN = greater of 5CK or 10ns; MAX = N/A								CK	
Power-Down Timing												
CKE MIN pulse width		<b><sup>1</sup>CKE (MIN)</b>	Greater of 3CK or 7.5ns	Greater of 3CK or 5.625ns	Greater of 3CK or 5.625ns	Greater of 3CK or 5ns	Greater of 3CK or 5ns	Greater of 3CK or 5ns	Greater of 3CK or 5ns	Greater of 3CK or 5ns	CK	
Command pass disable delay		<sup>1</sup> CPDED	MIN = 1; MAX = N/A								CK	
Power-down entry to power-down exit timing		<sup>1</sup> PD	MIN = <sup>1</sup> CKE (MIN); MAX = 9 * tREFI								CK	
Begin power-down period prior to CKE registered HIGH		<sup>1</sup> ANPD	WL - 1CK								CK	
Power-down entry period: ODT either synchronous or asynchronous		PDE	Greater of <sup>1</sup> ANPD or <sup>1</sup> RFC - REFRESH command to CKE LOW time								CK	
Power-down exit period: ODT either synchronous or asynchronous		PDX	<sup>1</sup> ANPD + <sup>1</sup> XPDLL								CK	
Power-Down Entry Minimum Timing												
ACTIVATE command to power-down entry		<sup>1</sup> ACTPDEN	MIN = 1								CK	
PRECHARGE/PRECHARGE ALL command to power-down entry		<sup>1</sup> PRPDEN	MIN = 1								CK	
REFRESH command to power-down entry		<sup>1</sup> REFPDEN	MIN = 1								CK	37
MRS command to power-down entry		<sup>1</sup> MRSPDEN	MIN = <sup>1</sup> MOD (MIN)								CK	
READ/READ with auto precharge command to power-down entry		<sup>1</sup> RDPDEN	MIN = RL + 4 + 1								CK	
WRITE command to power-down entry	BL8 (OTF, MRS) BC4OTF	<sup>1</sup> WRPDEN	MIN = WL + 4 + <sup>1</sup> WR/CK (AVG)								CK	
	BC4MRS	<sup>1</sup> WRPDEN	MIN = WL + 2 + <sup>1</sup> WR/CK (AVG)								CK	

# Tiempos

DDR3-1066 Speed Bin		-187E		-187		Unit	Notes	
CL- <sup>t</sup> RCD- <sup>t</sup> RP		7-7-7		8-8-8				
Parameter	Symbol	Min	Max	Min	Max			
Internal READ command to first data	<sup>t</sup> AA	13.125	–	15	–	ns		
ACTIVATE to internal READ or WRITE delay time	<sup>t</sup> RCD	13.125	–	15	–	ns		
PRECHARGE command period	<sup>t</sup> RP	13.125	–	15	–	ns		
ACTIVATE-to-ACTIVATE or REFRESH command period	<sup>t</sup> RC	50.625	–	52.5	–	ns		
ACTIVATE-to-PRECHARGE command period	<sup>t</sup> RAS	37.5	9 x <sup>t</sup> REFI	37.5	9 x <sup>t</sup> REFI	ns	1	
CL = 5	CWL = 5	<sup>t</sup> CK (AVG)	3.0	3.3	3.0	3.3	ns	2
	CWL = 6	<sup>t</sup> CK (AVG)	Reserved		Reserved		ns	3
CL = 6	CWL = 5	<sup>t</sup> CK (AVG)	2.5	3.3	2.5	3.3	ns	2
	CWL = 6	<sup>t</sup> CK (AVG)	Reserved		Reserved		ns	3
CL = 7	CWL = 5	<sup>t</sup> CK (AVG)	Reserved		Reserved		ns	3
	CWL = 6	<sup>t</sup> CK (AVG)	1.875	<2.5	Reserved		ns	2, 3
CL = 8	CWL = 5	<sup>t</sup> CK (AVG)	Reserved		Reserved		ns	3
	CWL = 6	<sup>t</sup> CK (AVG)	1.875	<2.5	1.875	<2.5	ns	2
Supported CL settings		5, 6, 7, 8		5, 6, 8		CK		
Supported CWL settings		5, 6		5, 6		CK		

# Registro SDRAM\_TIM\_1

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-25	reg_t_rp	R/W	0h	Minimum number of DDR clock cycles from Precharge to Activate or Refresh, minus one.
24-21	reg_t_rcd	R/W	0h	Minimum number of DDR clock cycles from Activate to Read or Write, minus one.
20-17	reg_t_wr	R/W	0h	Minimum number of DDR clock cycles from last Write transfer to Pre-charge, minus one. The SDRAM initialization sequence will be started when the value of this field is changed from the previous value and the EMIF is in DDR2 mode.
16-12	reg_t_ras	R/W	0h	Minimum number of DDR clock cycles from Activate to Pre-charge, minus one. $\text{reg\_t\_ras} \geq \text{reg\_t\_rcd}$ .
11-6	reg_t_rc	R/W	0h	Minimum number of DDR clock cycles from Activate to Activate, minus one.
5-3	reg_t_rrd	R/W	0h	Minimum number of DDR clock cycles from Activate to Activate for a different bank, minus one. For an 8 bank DDR2 and DDR3, this field must be equal to $((\text{tFAW}/(4 \cdot \text{tCK}))-1)$ .
2-0	reg_t_wtr	R/W	0h	Minimum number of DDR clock cycles from last Write to Read, minus one.

# Registro SDRAM\_TIM\_2

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	reg_t_xp	R/W	0h	Minimum number of DDR clock cycles from Powerdown exit to any command other than a Read command, minus one. For DDR2 and LPDDR1, this field must satisfy greater of tXP or tCKE.
27-25	reg_t_odt	R/W	0h	Minimum number of DDR clock cycles from ODT enable to write data driven for DDR2 and DDR3. reg_t_odt must be equal to tAOND.
24-16	reg_t_xsnr	R/W	0h	Minimum number of DDR clock cycles from Self-Refresh exit to any command other than a Read command, minus one.
15-6	reg_t_xsrd	R/W	0h	Minimum number of DDR clock cycles from Self-Refresh exit to a Read command, minus one.
5-3	reg_t_rtp	R/W	0h	Minimum number of DDR clock cycles from the last Read command to a Pre-charge command for DDR2 and DDR3, minus one.
2-0	reg_t_cke	R/W	0h	Minimum number of DDR clock cycles between pad_cke_o changes, minus one.

# Registro SDRAM\_TIM\_3

Bit	Field	Type	Reset	Description
31-28	reg_t_pdll_ul	R/W	0h	Minimum number of DDR clock cycles for PHY DLL to unlock. A value of N will be equal to N x 128 clocks.
27-24	RESERVED	R	0h	
23-21	RESERVED	R/W	0h	Reserved.
20-15	reg_zq_zqcs	R/W	0h	Number of DDR clock clock cycles for a ZQCS command, minus one.
14-13	RESERVED	R/W	0h	Reserved.
12-4	reg_t_rfc	R/W	0h	Minimum number of DDR clock cycles from Refresh or Load Mode to Refresh or Activate, minus one.
3-0	reg_t_ras_max	R/W	0h	Maximum number of reg_refresh_rate intervals from Activate to Precharge command. This field must be equal to $((t_{RASmax} / t_{REFI}) - 1)$ rounded down to the next lower integer. This field is only applicable for mDDR. This field must be programmed to 0xF for other SDRAM types.