



Bridge of Life
Education

SOC Design Memory - DRAM

Topics

- DRAM Configuration
- DRAM Commands & Timing
- Low Power Features
- DRAM Controller and Performance Optimization

Modern DRAM Type

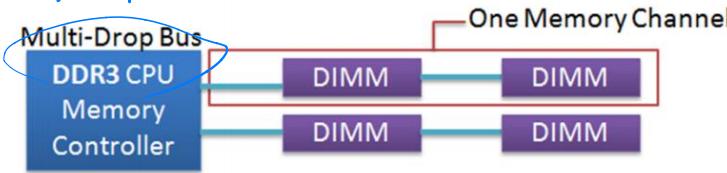
low power

DRAM Type	Banks per Rank	Bank Groups	3D-Stacked	Low-Power
DDR3	8			
DDR4	16	✓	<i>increased latency</i>	
GDDR5	16	✓	<i>increased area/power</i>	
HBM High-Bandwidth Memory	16		✓	
HMC Hybrid Memory Cube	256	<i>narrower rows, higher latency</i>		
Wide I/O	4		✓	✓
Wide I/O 2	8		✓	✓
LPDDR3	8			✓
LPDDR4	16			✓

DRAM Configuration

- Rank (CS), DIMM, Channel
- Bank, Bank Group
 - Number of banks: 8 – 16
 - Bank Group: 2-4
 - Separate activation, read, write or refresh underway in each of bank group
 - Increase memory bandwidth and efficiency for small granularity of access
- Page registers – keep track which page is activated, ready for read/write access

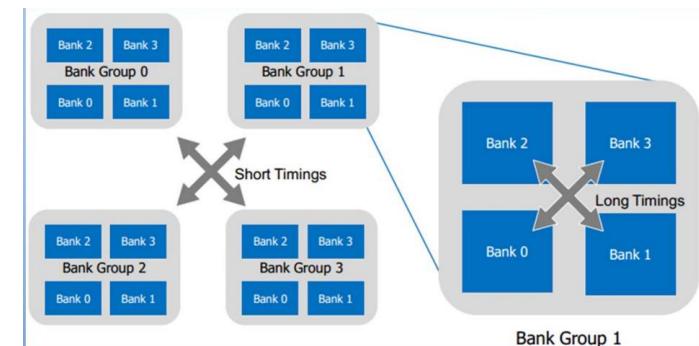
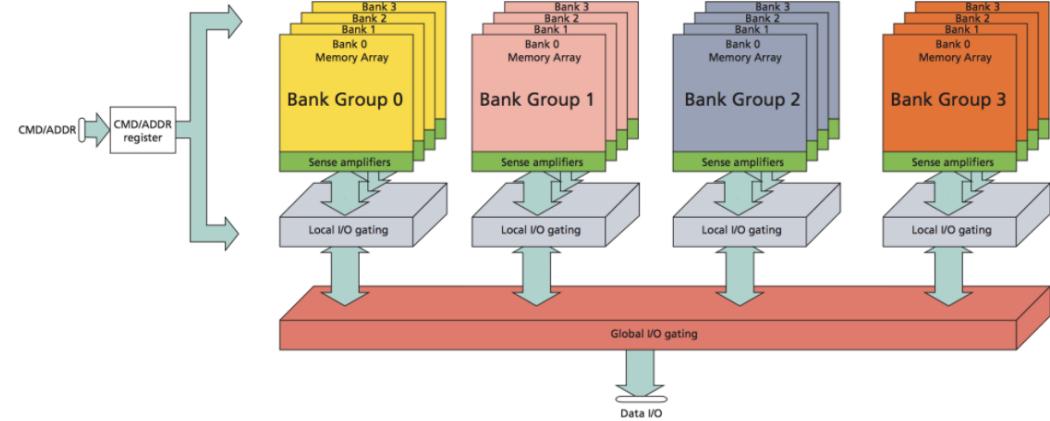
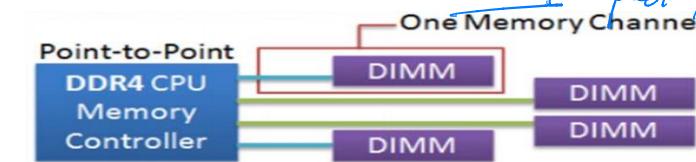
indep addr / Data



4 4
reflection speed



8 GB DDR4-2133 ECC 1.2 V RDIMMs



From the perceptive of performance

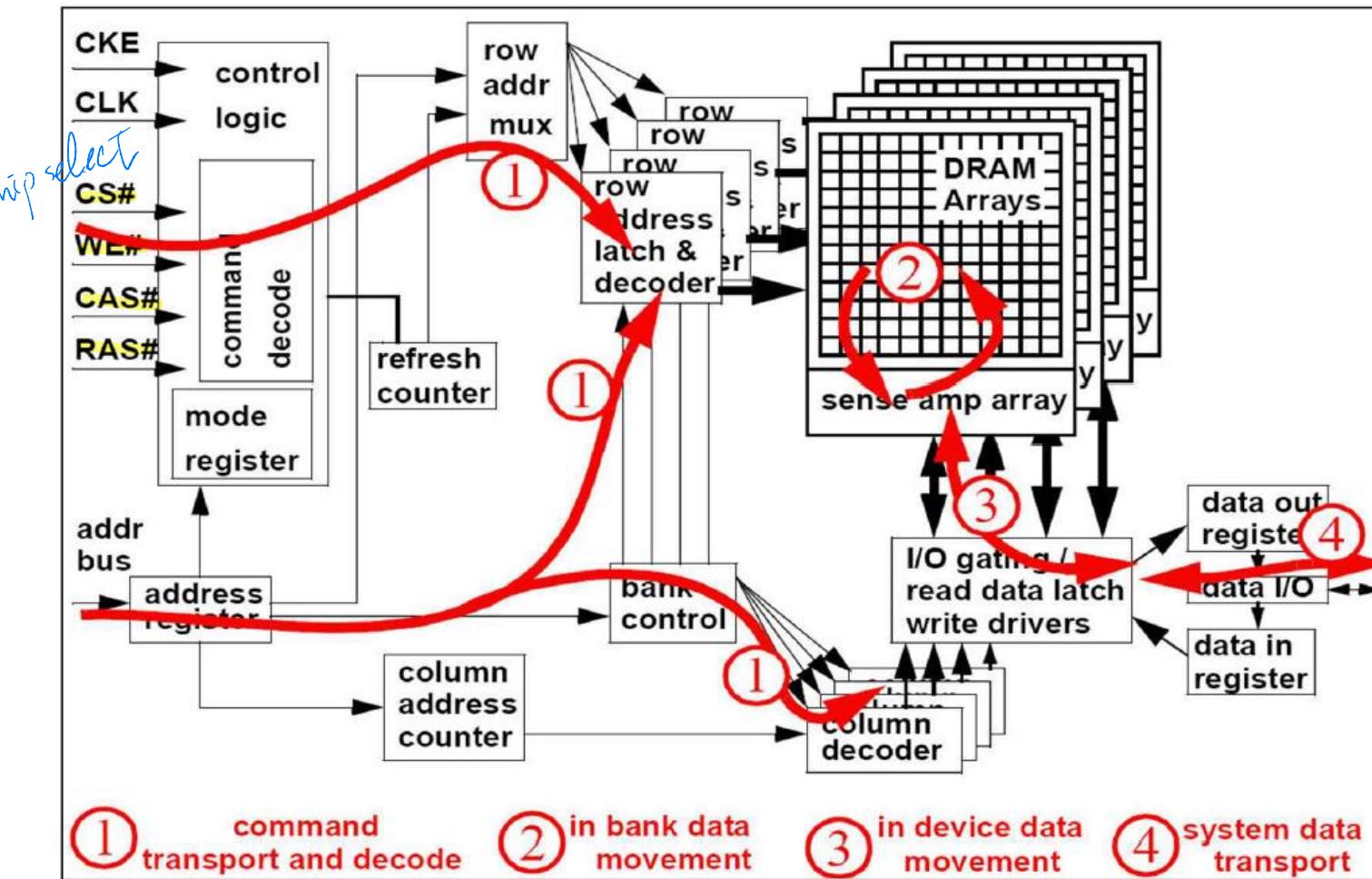
- Device Behavior
 - Multi-bank / page registers - bank interleave, open-page
 - Pipeline Burst access
 - Read/write turn-around
- Controller features
 - Interface to be compatible with memory device specification
 - Features for optimizing bus utilization

DRAM Configuration

Device 4Gb x 8

- Number of Row Address bits: A0-A14 = 15 bits => Total number of row = $2^{15} = 32K$
- Number of Column Address bits: A0-A9 = 10 bits => Number of columns per row = 1K
- Width of each column = 8 bits
- Number of Bank Groups = 4
- Number of Banks = 4
- Total DRAM Capacity =
 - Num.Rows x Num.Columns x Width.of.Column x Num.BankGroups x Num.Banks
 - $32K \times 1K \times 8 \times 4 \times 4 = 4Gb$
- **Page-size** (assume 64-bit DRAM channel)
 - Num.Columns x 64-bit = $1K \times 8B = 8KB$ → A_{13} for interleave
 - 2 bit

DRAM Access Sequence



- Five basic DRAM commands for four phases

- Row Access Command
- Column Read Command
- Column Write Command
- Precharge Command
- Refresh Command

ACTivate

READ/WRITE

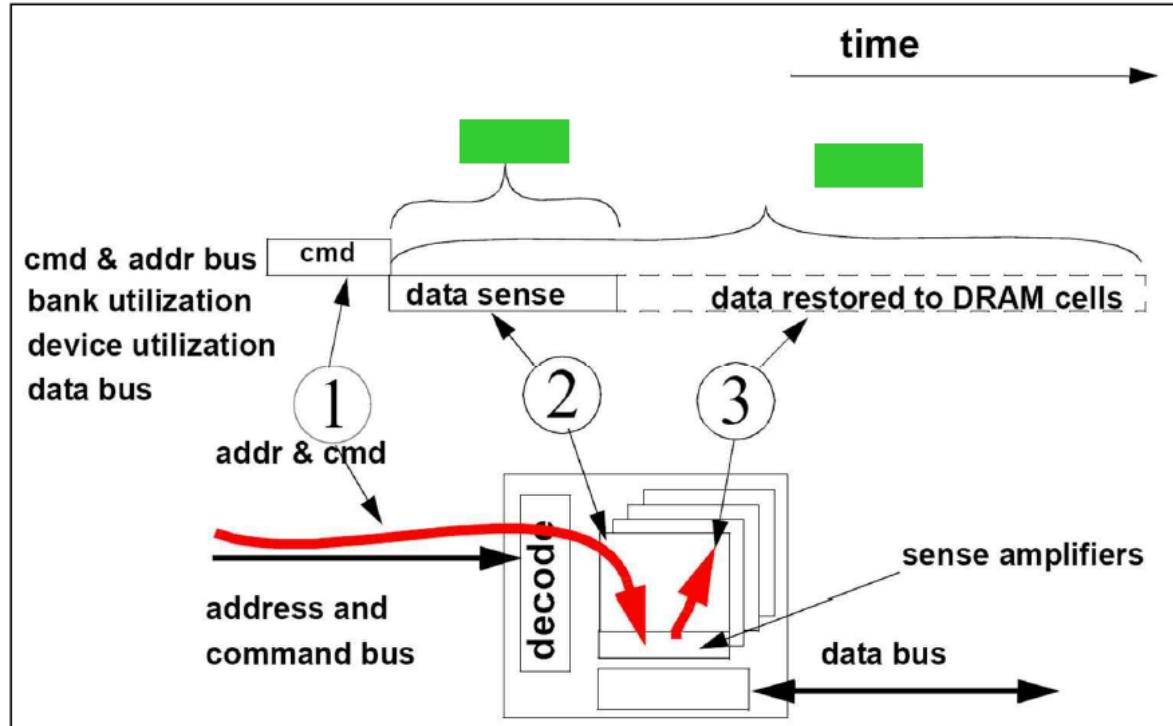
PREcharge

REFresh

Command Table

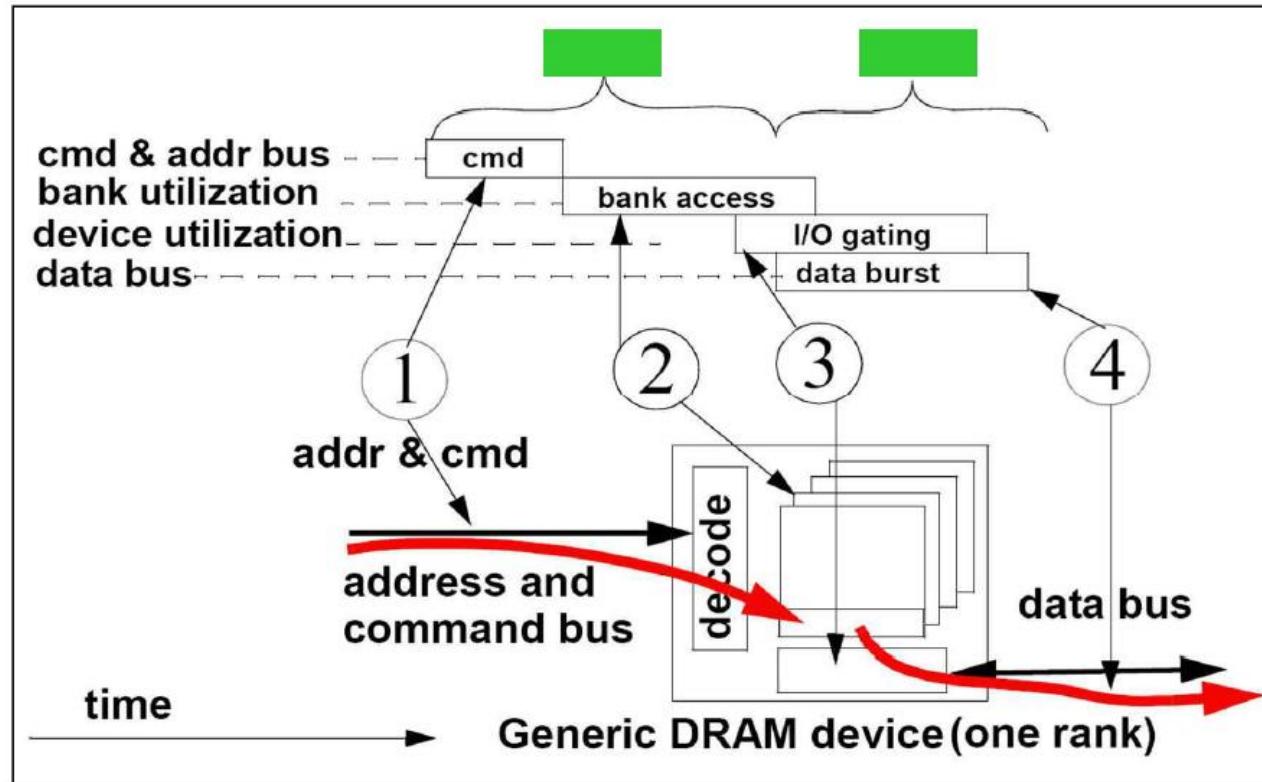
Name (Function)	CS#	RAS#	CAS#	WE#	DQM	ADDR	DQ	Notes
COMMAND INHIBIT (NOP)	H	X	X	X	X	X	X	
NO OPERATION (NOP)	L	H	H	H	X	X	X	
ACTIVE (select bank and activate row)	L	L	H	H	X	Bank/row	X	2
READ (select bank and column, and start READ burst)	L	H	L	H	L/H	Bank/col	X	3
WRITE (select bank and column, and start WRITE burst)	L	H	L	L	L/H	Bank/col	Valid	3
BURST TERMINATE	L	H	H	L	X	X	Active	4
PRECHARGE (Deactivate row in bank or banks)	L	L	H	L	X	Code	X	5
AUTO REFRESH or SELF REFRESH (enter self refresh mode)	L	L	L	H	X	X	X	6, 7
LOAD MODE REGISTER	L	L	L	L	X	Op-code	X	8
Write enable/output enable	X	X	X	X	L	X	Active	9
Write inhibit/output High-Z	X	X	X	X	H	X	High-Z	9

Row Access Command



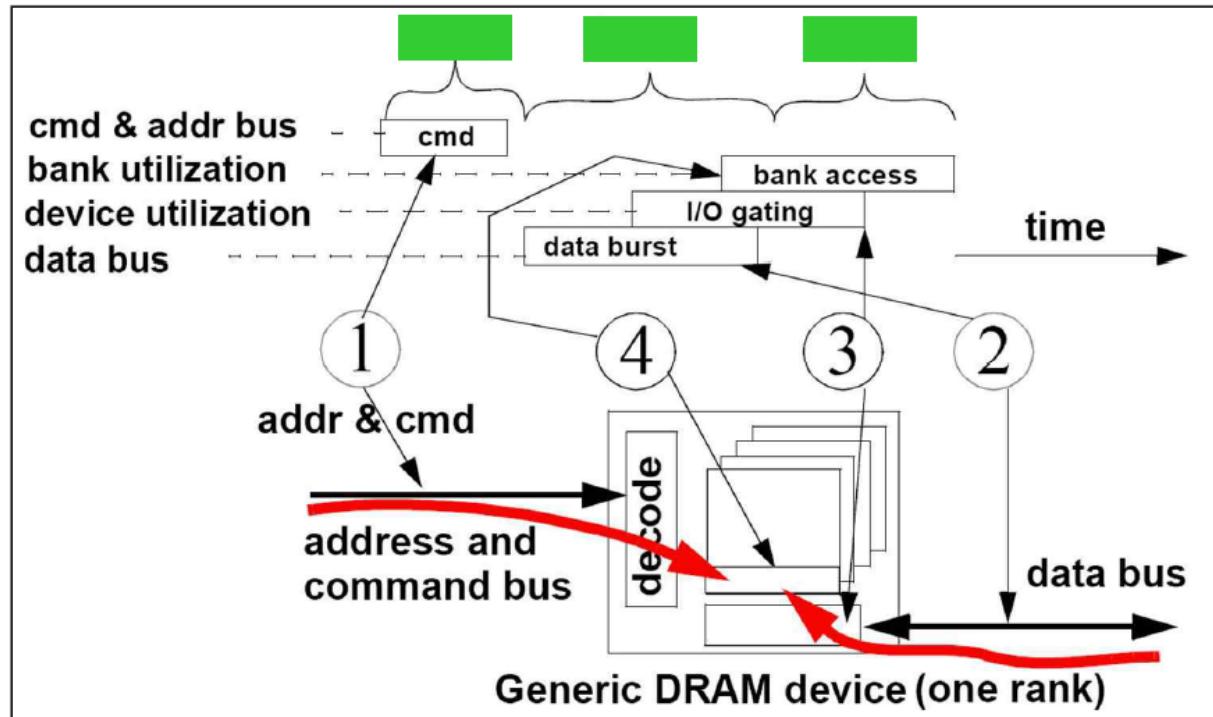
- Move data from cells in the DRAM array to the sense amp. and then restore the data back into the cells in the DRAM array as part of the same command
 - t_{RCD} (Row to column command delay): The time it takes to move data from DRAM cell to Sense Amps
 - t_{RAS} (Row Access Strobe): The time it takes to move data from to discharge and restore data from the row of DRAM cells

Column Read Command



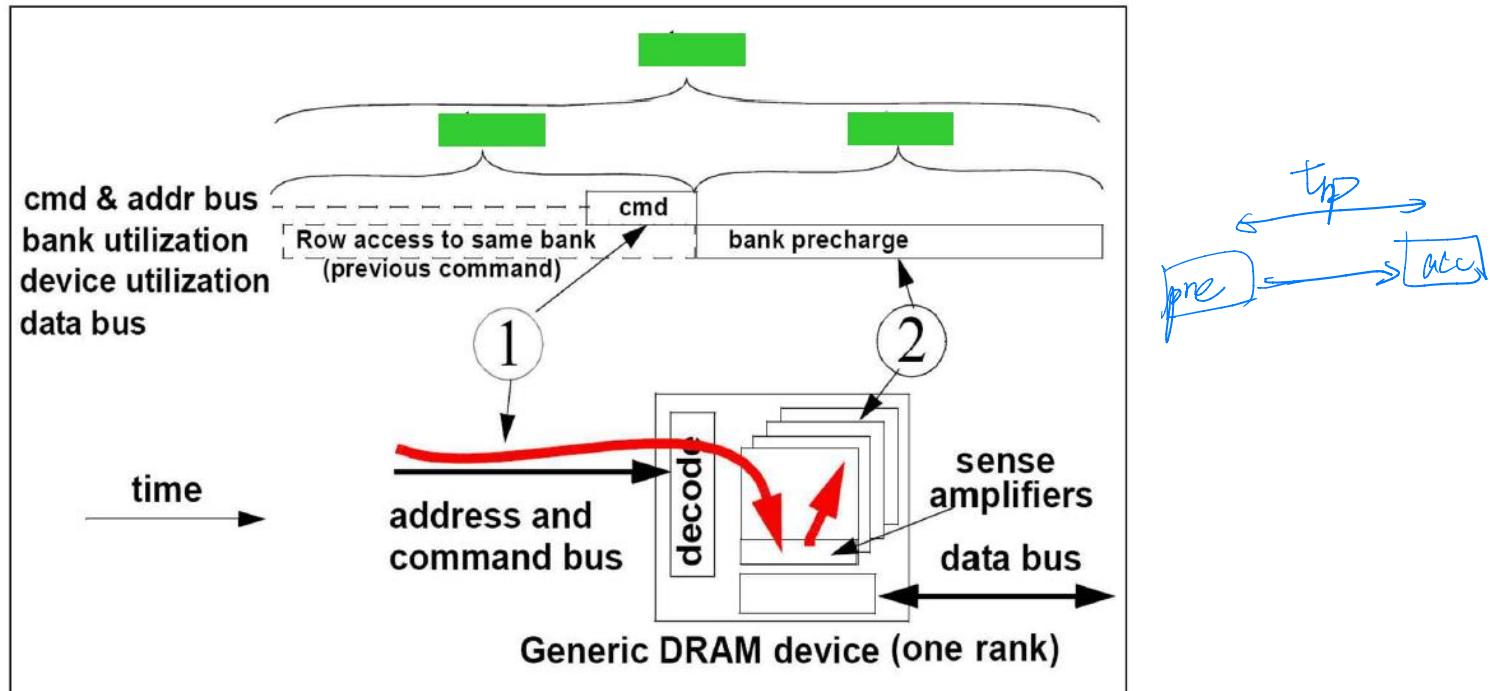
- Moves data from the array of sense amps. of a given bank of DRAM array through the data back to the memory controller
 - t_{CAS} (Column Access Strobe latency): The time it takes for the DRAM device to place the requested data onto the data bus
 - t_{Burst} : The duration of the data burst on the data bus for a single column-read command

Column Write Command



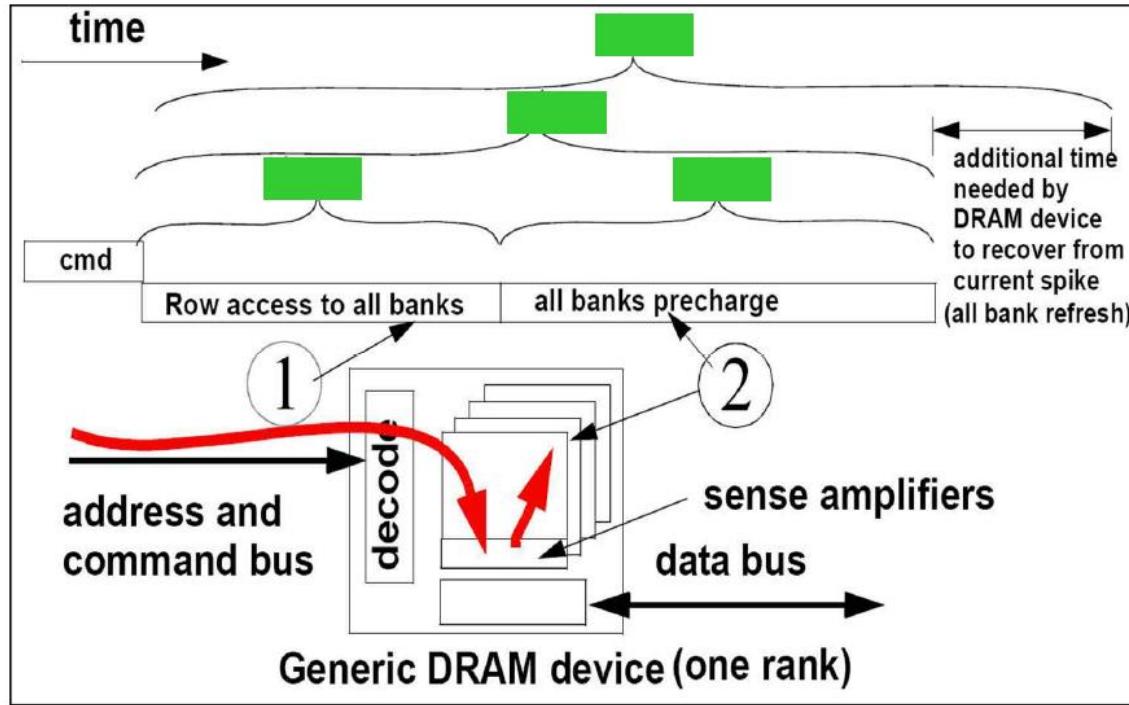
- Moves data from the memory controller to the sense amps. of the targeted bank
 - t_{CWD} (Column Write Delay): Timing between assertion of the column-write command on the command bus and the placement of the write data onto the data bus by the memory controller
 - t_{Burst} (Data Burst duration)
 - t_{WR} (Write Recovery time): The time it takes for the write data to propagate into the DRAM arrays

Precharge



- Completes the row access sequences as it **resets the sense amps**, and the bitlines and prepares them for another row access command to the same array of DRAM cells.
 - t_{RC} (Row Cycle)
 - t_{RAS} (Row Access Strobe)
 - **t_{RP}** (Row Precharge): Bitlines and sense amps. are properly precharged

Refresh



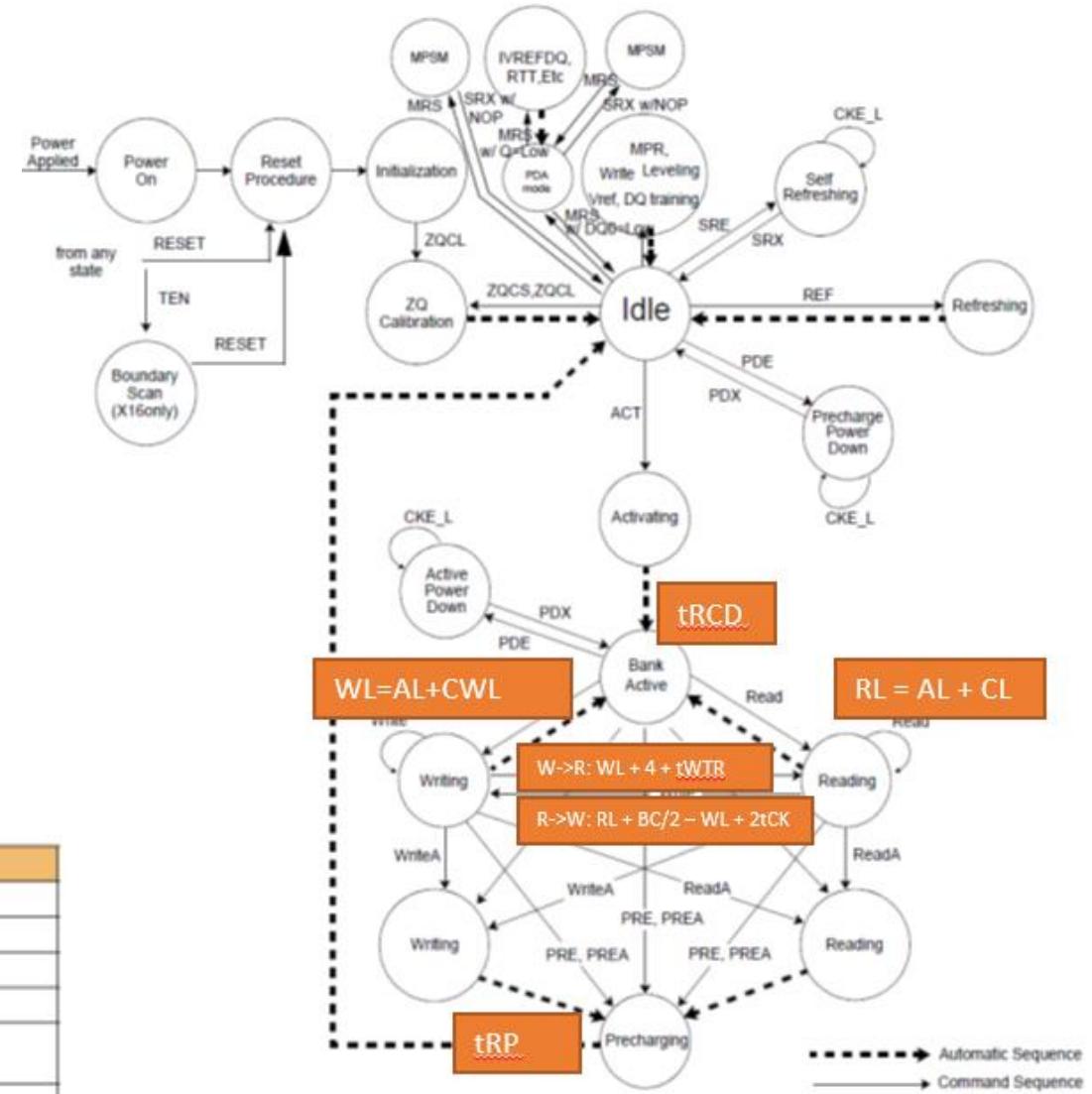
- Store complete value to DRAM cell
 - t_{RAS} (Row Access Strobe)
 - t_{RC} (Row Cycle)
 - t_{RP} (Row Precharge)
 - t_{RFC} (Refresh Cycle Time)

State Machine

Transaction flow

- Start Page :
 - accessed bank is idle
 - Idle -> Activate (tRCD) -> Read/Write
 - On Page:
 - accessed bank is already activated
 - Bank Active -> Read/write
 - Off Page:
 - accessed bank is different from current activated bank
 - Bank Active -> Precharge (tRP) -> Activate -> Read/Write

Abbreviation	Function	Abbreviation	Function	Abbreviation	Function
ACT	Activate	Read	RD,RDS4, RDS8	PDE	Enter Power-down
PRE	Precharge	Read A	RDA, RDAS4, RDAS8	PDX	Exit Power-down
PREA	PRECHARGE All	Write	WR, WRS4, WRS8 with/without CRC	SRE	Self-Refresh entry
MRS	Mode Register Set	Write A	WRA,WRAS4, WRAS8 with/without CRC	SRX	Self-Refresh exit
REF	Refresh, Fine granularity Refresh	RESET_n	Start RESET procedure	MPR	Multi Purpose Register
TEN	Boundary Scan Mode Enable				



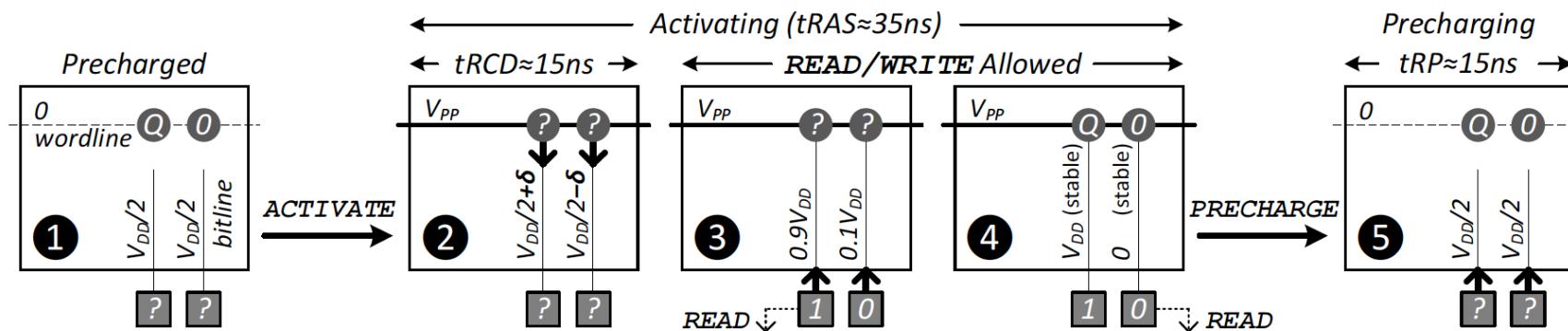


Figure 4. DRAM bank operation: Steps involved in serving a memory request [17] ($V_{PP} > V_{DD}$)

Category	RowCmd \leftrightarrow RowCmd			RowCmd \leftrightarrow ColCmd			ColCmd \leftrightarrow ColCmd			ColCmd \rightarrow DATA	
	tRC	$tRAS$	tRP	$tRCD$	$tRTP$	tWR^*	$tCCD$	$tRTW^\dagger$	$tWTR^*$	CL	CWL
Commands	A \rightarrow A	A \rightarrow P	P \rightarrow A	A \rightarrow R/W	R \rightarrow P	W $^*\rightarrow$ P	R(W) \rightarrow R(W)	R \rightarrow W	W $^*\rightarrow$ R	R \rightarrow DATA	W \rightarrow DATA
Scope	Bank	Bank	Bank	Bank	Bank	Bank	Channel	Rank	Rank	Bank	Bank
Value (ns)	~50		~35	13-15	13-15	~7.5	15	5-7.5	11-15	~7.5	13-15

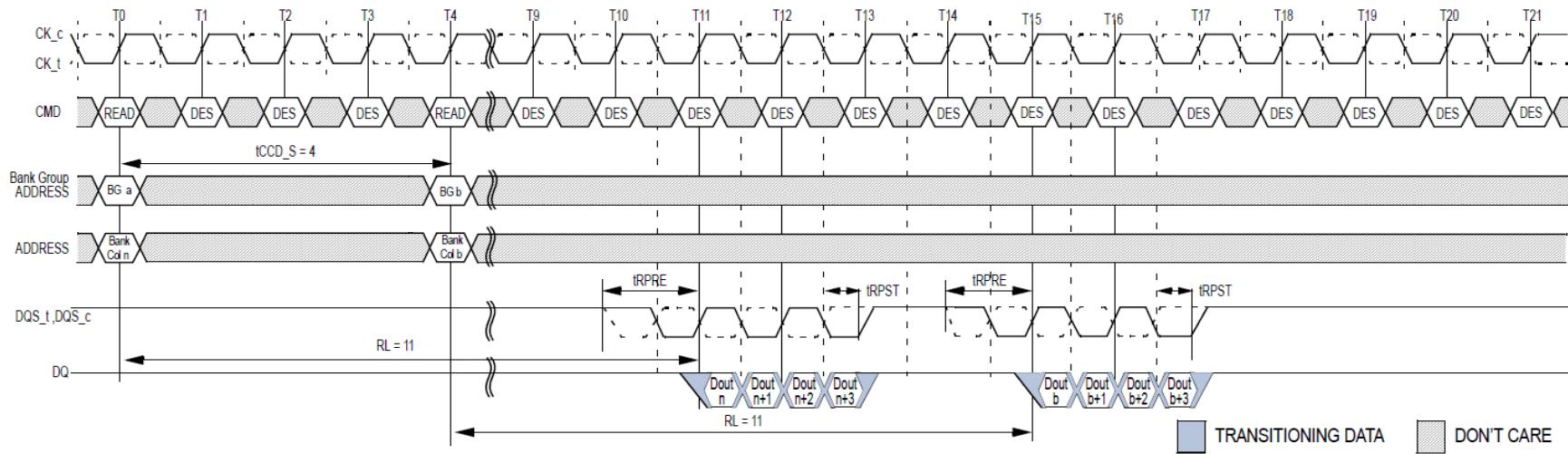
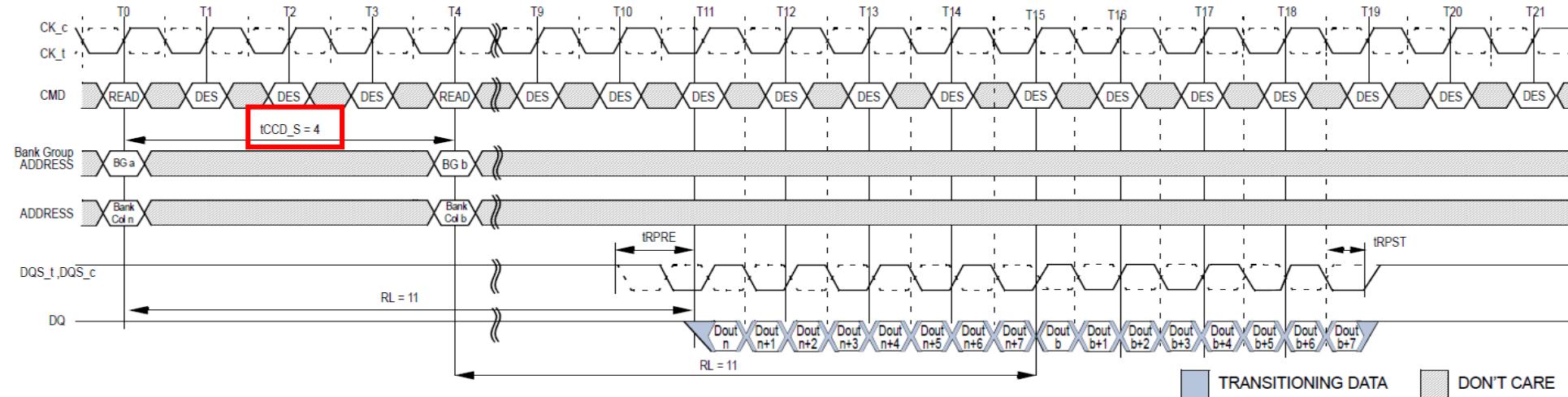
A: ACTIVATE– P: PRECHARGE– R: READ– W: WRITE

* Goes into effect after the last write *data*, not from the WRITE command

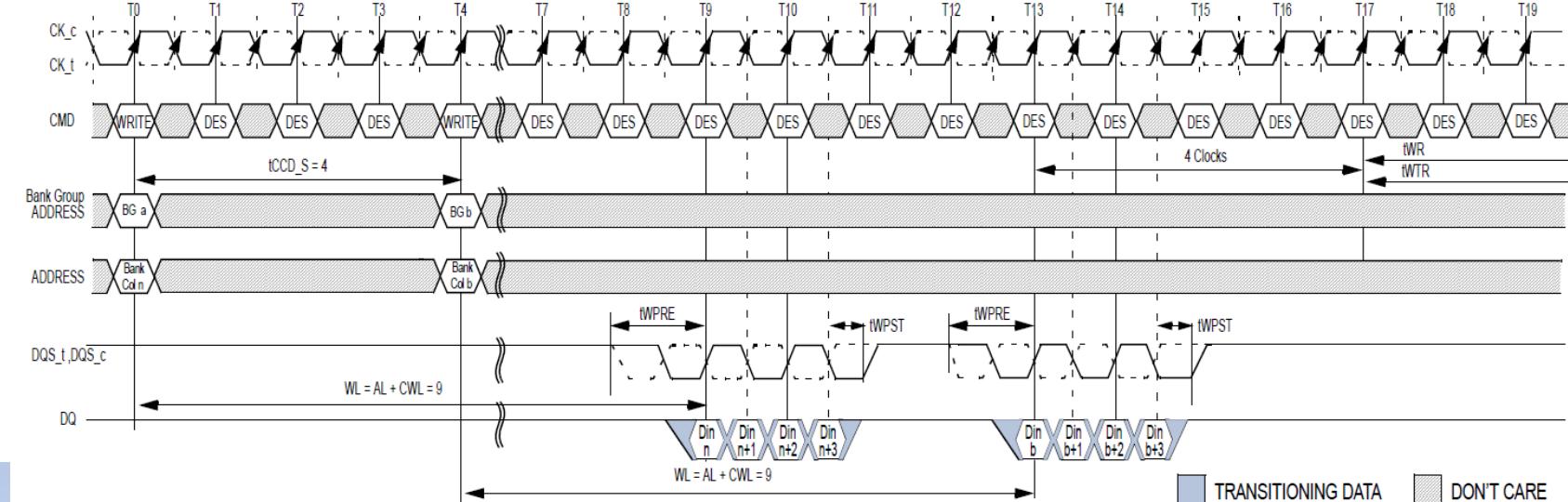
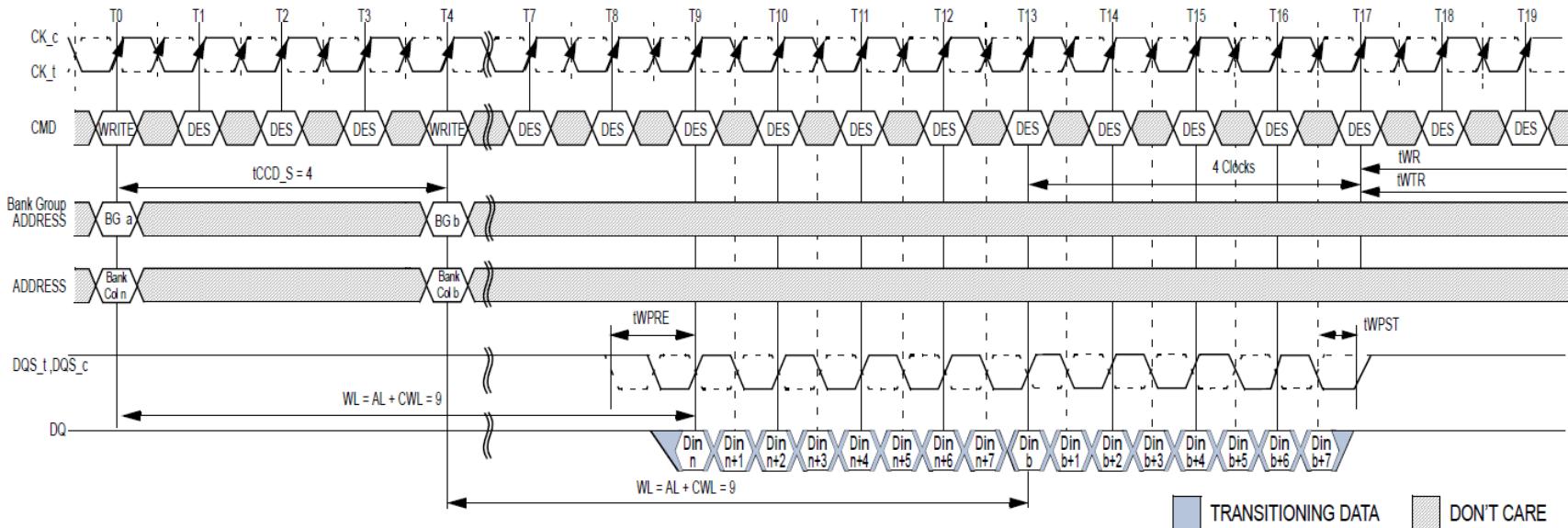
† Not explicitly specified by the JEDEC DDR3 standard [18]. Defined as a function of other timing constraints.

Table 1. Summary of DDR3-SDRAM timing constraints (derived from Micron's 2Gb DDR3-SDRAM datasheet [33])

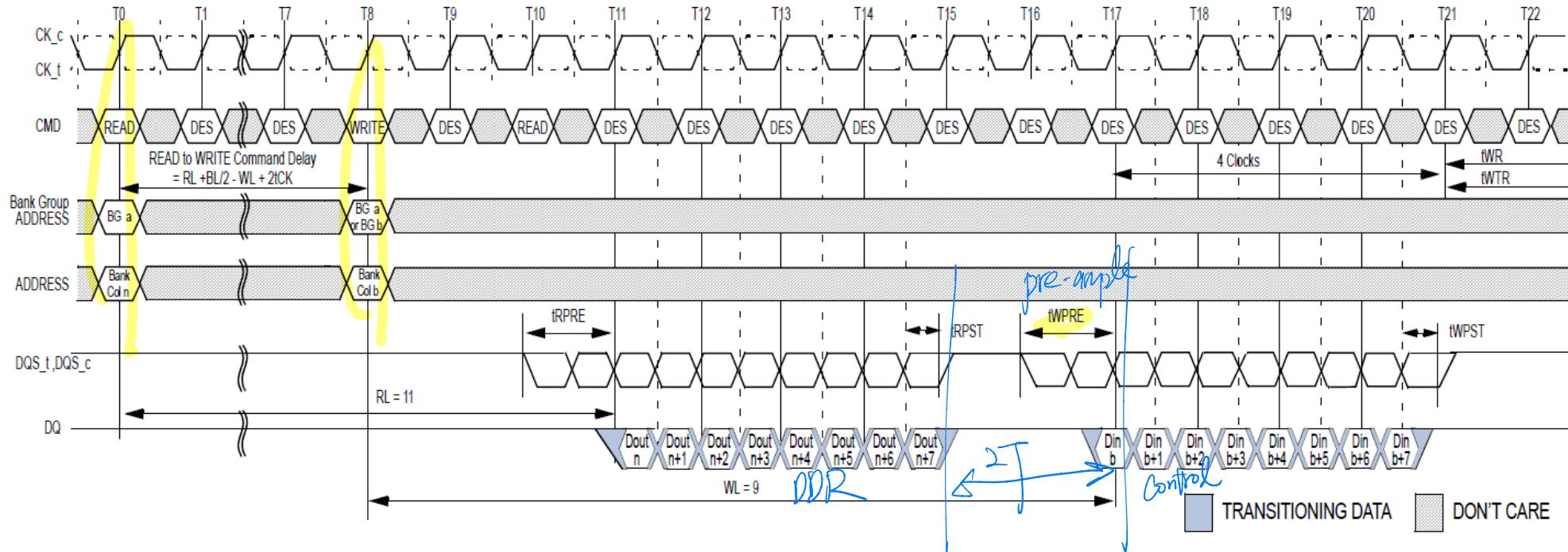
Consecutive Read BL8, BL4



Consecutive WRITE BL8, BC4



READ(BL8) to WRITE(BL8) Turn-around

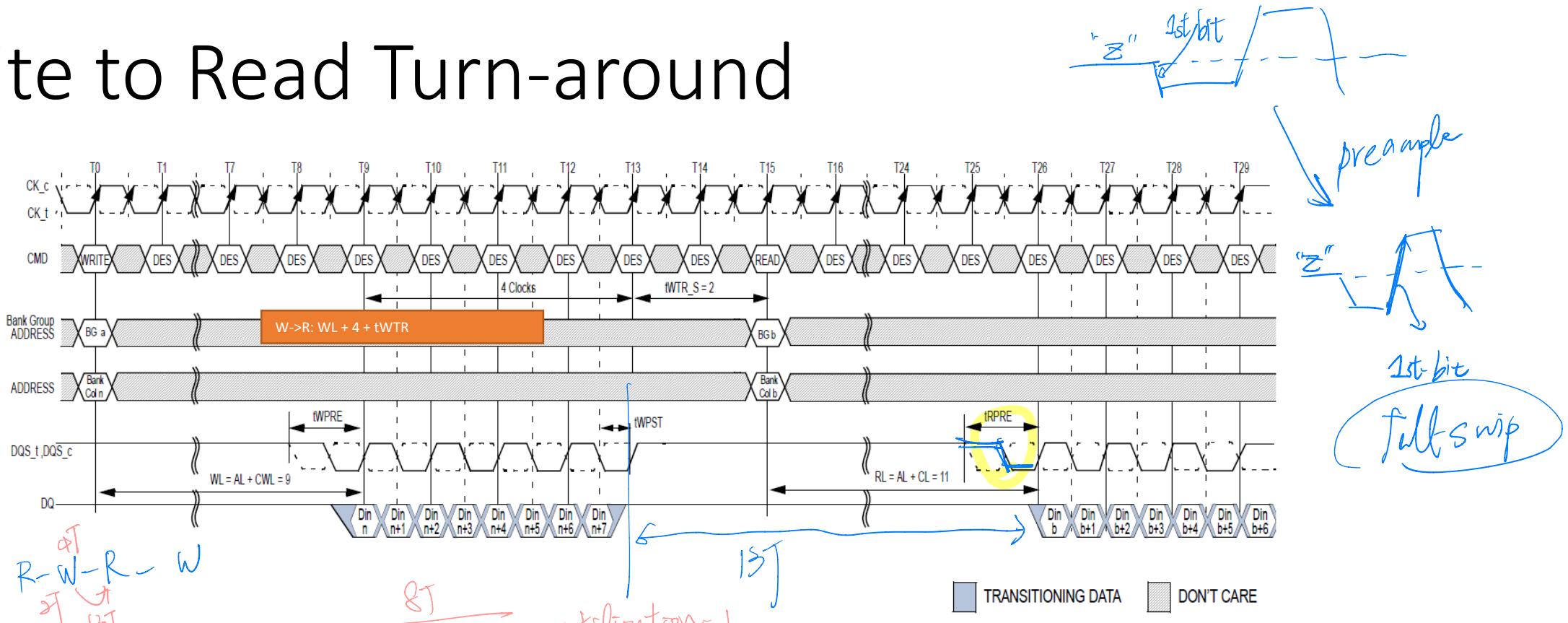


NOTE :

1. BL = 8, RL = 11 (CL = 11, AL = 0), Read Preamble = 1tCK, WL = 9 (CWL = 9, AL = 0), Write Preamble = 1tCK
2. DOUT n = data-out from column n, DIN b = data-in to column b.
3. DES commands are shown for ease of illustration; other commands may be valid at these times.
4. BL8 setting activated by either MR0[A1:A0 = 0:0] or MR0[A1:A0 = 0:1] and A12 = 1 during READ command at T0 and WRITE command at T8.
5. CA Parity = Disable, CS to CA Latency = Disable, Read DBI = Disable, Write DBI = Disable, Write CRC = Disable.

Figure 79 — READ (BL8) to WRITE (BL8) with 1tCK Preamble in Same or Different Bank Group

Write to Read Turn-around



NOTE:

1. BC = 4, AL = 0, CWL = 9, CL = 11, Preamble = 1tCK
2. DIN n = data-in to column n(or column b). DOUT b = data-out from column b.
3. DES commands are shown for ease of illustration; other commands may be valid at these times.
4. BL8 setting activated by either MR0[A1:A0 = 0:0] or MR0[A1:A0 = 0:1] and A12 = 1 during WRITE command at T0 and READ command at T15.
5. CA Parity = Disable, CS to CA Latency = Disable, Write DBI = Disable.
6. The write timing parameter (tWTR_S) are referenced from the first rising clock edge after the last write data shown at T13.

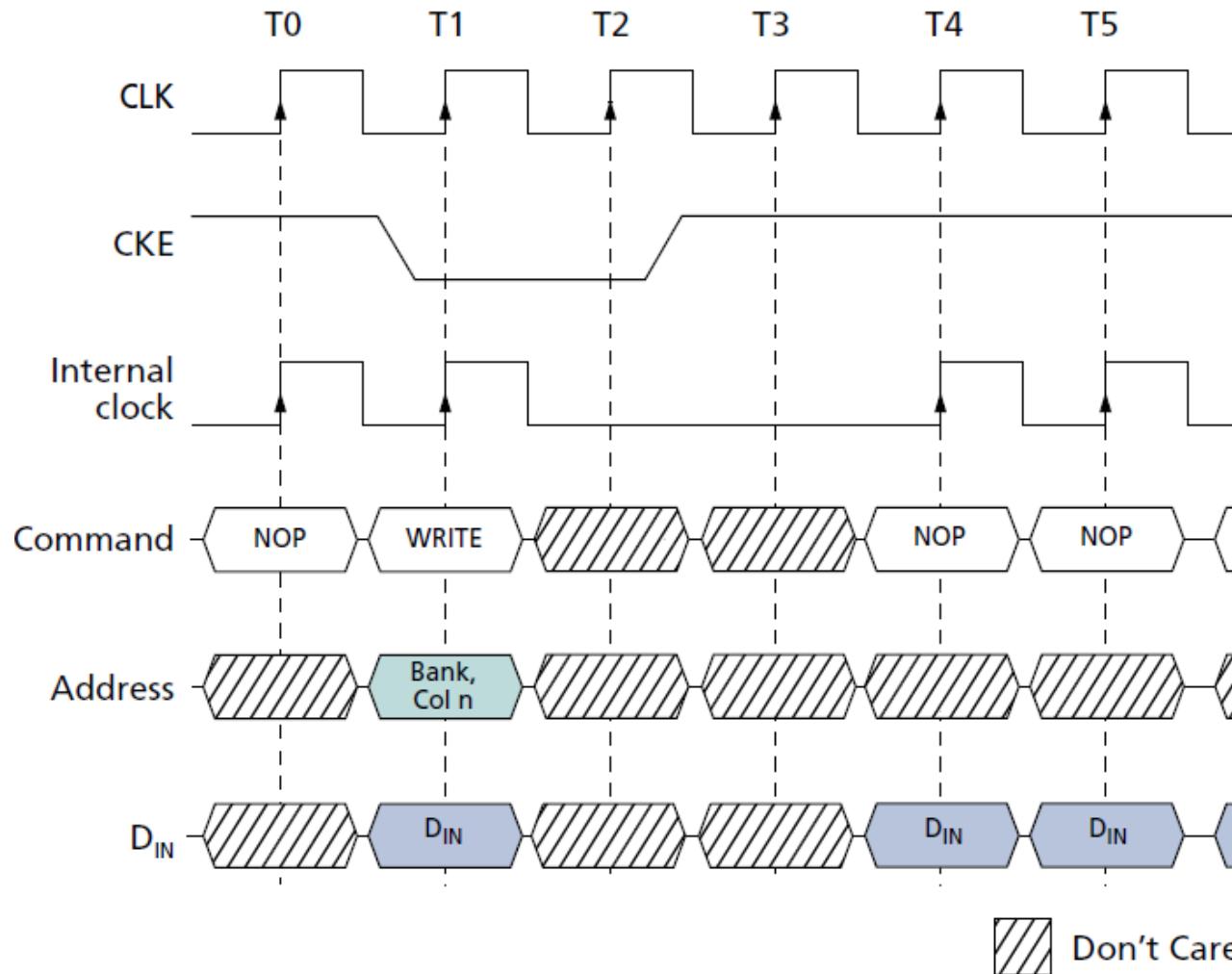
Figure 114 — WRITE (BL8) to READ (BL8) with 1tCK Preamble in Different Bank Group

Low Power

- Clock Suspend
- Self-Refresh
- Power-Down

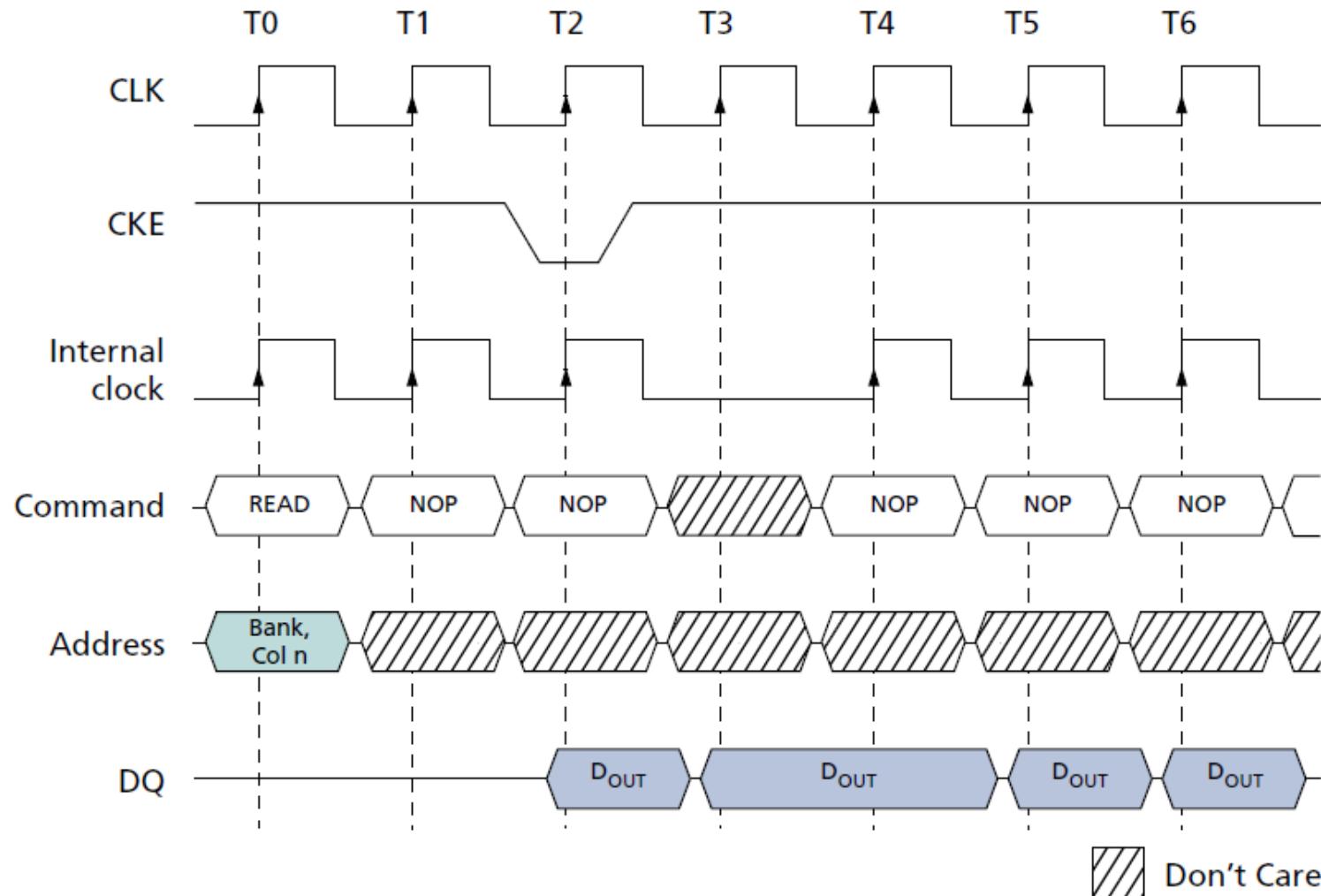
Clock Suspend During WRITE Burst

BL = 4 or greater, and DQM is LOW.



Clock Suspend During READ Burst

CL = 2, BL = 4 or greater, and DQM is LOW.



Self REFRESH Mode

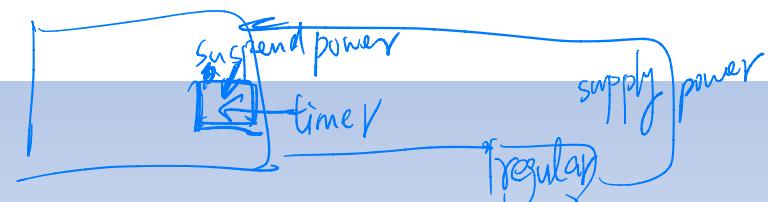
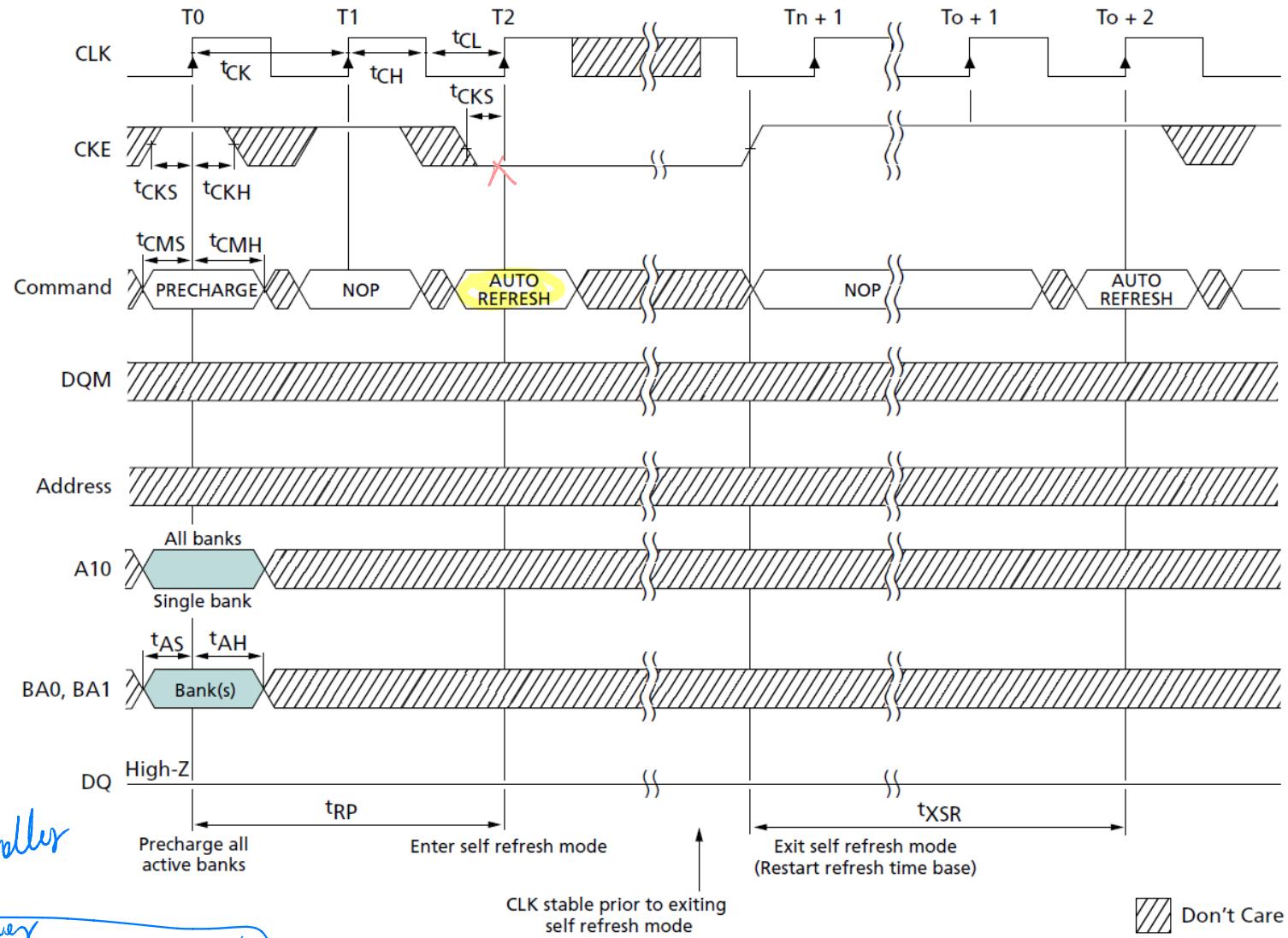
- Retain data when the rest of the system is power down
- Entry by CKE low AUTO REFRESH at CKE low
- Use internal clocking
- Can disable external clock

System suspend

1. controller \rightarrow self-prefetch mode.

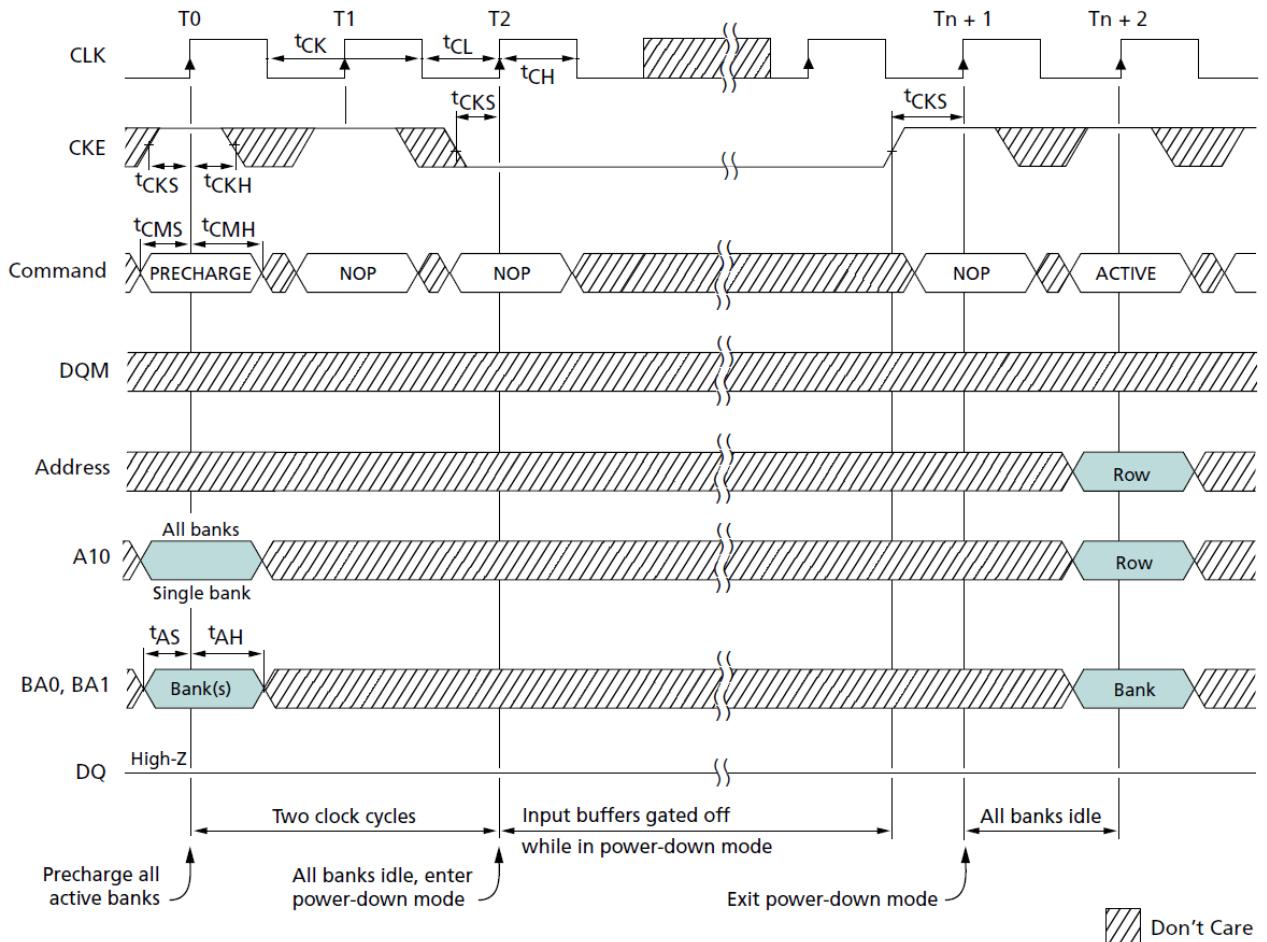
2. clock off

3. power down all system SOC include controller

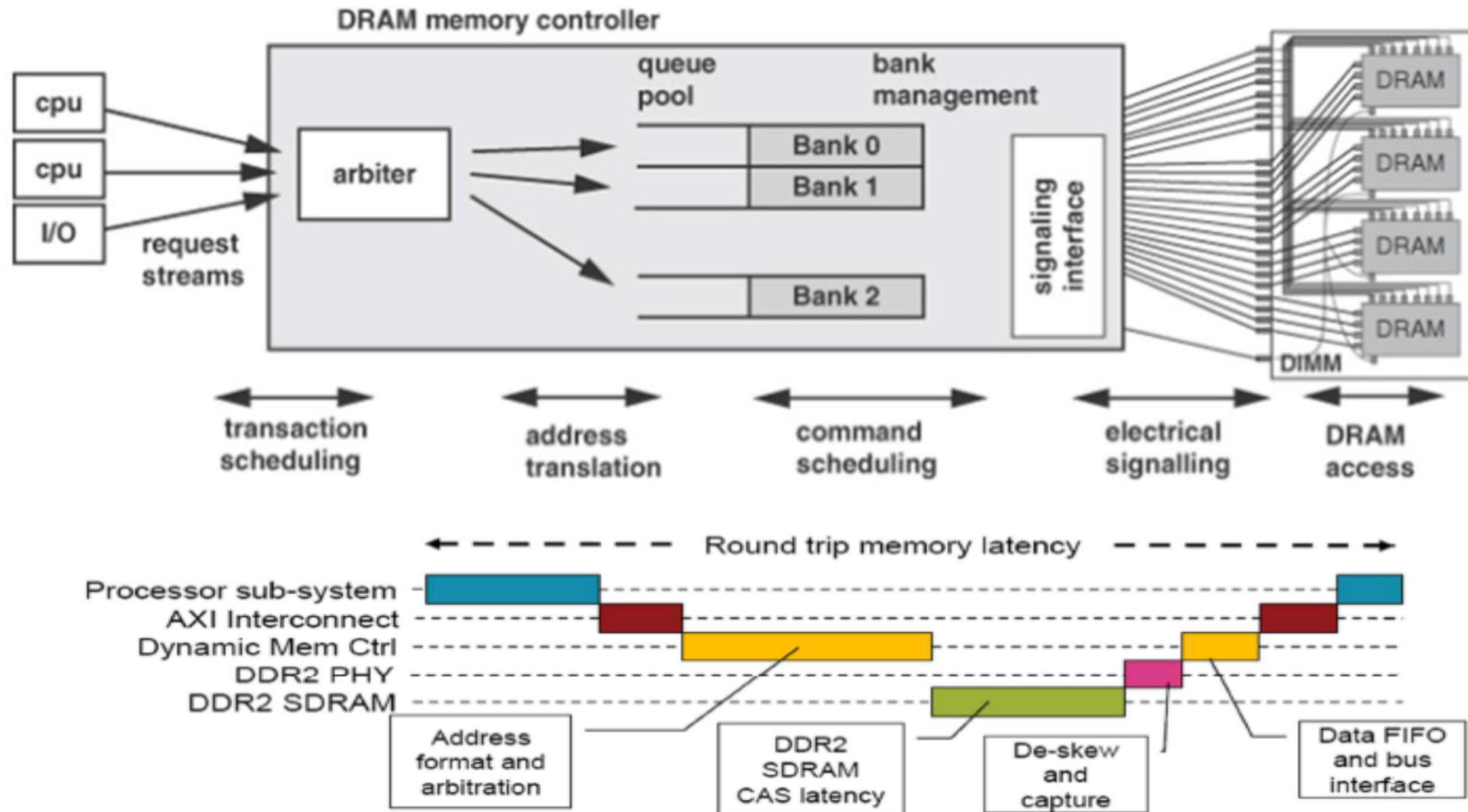


Power-Down Mode

- Enter: CKE disable (low) when NOP command
- Precharge power-down – when all banks are idle
- Active power-down – there is a row active in any bank
- Deactivates the input and output except CKE
- Can not remain in power down longer than refresh period (64ms)

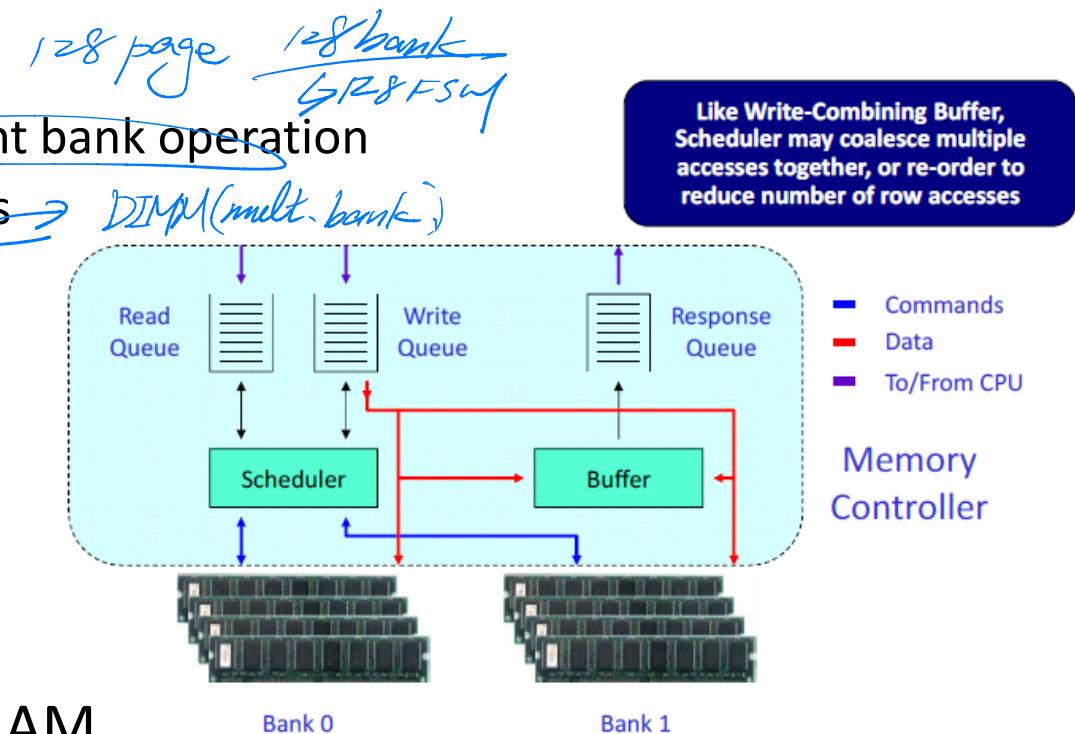


DRAM Controller – Design to Minimize Latency



DDR3/DDR4 Memory Controller Functions

- Meet DRAM protocol and Timing requirement (50+ timing constraints)
- Buffering/Scheduling for high perform + QoS
 - Command queues/state machine for each concurrent bank operation
 - Interleave DRAM command for multiple transactions
 - Reordering, page/bank/rank/channel management
 - Reordering to maximize bus utilization
- Re-order transactions (read over write)
- Open page/closed page policy (AutoPrecharge)
- Refresh scheduling - opportunistic refresh
- Manage power consumption and thermal in DRAM
 - Turn on/off DRAM chips, manage power modes



Page Management Policies

- Open row
 - Keep the row open after an access
 - + Next access might need the same row → row hit
 - Next access might need a different row → row conflict, wasted energy
- Closed row
 - Close the row after an access (if no other requests already in the request buffer need the same row)
 - + Next access might need a different row → avoid a row conflict
 - Next access might need the same row → extra activate latency
- Adaptive policies
 - Predict whether or not the next access to the bank will be to the same row and act accordingly, e.g. idle timer

Application Optimization

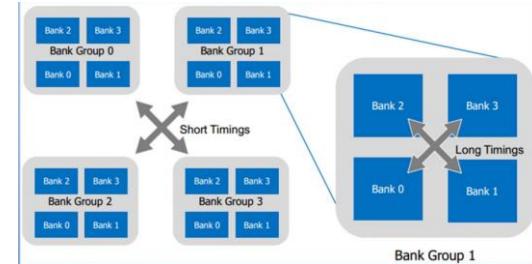
- Considering DRAM access latency – deeper transaction pipeline
- Minimize read-write turn-around (minimize read/write dependency)
- Carefully design the data structure/addressing scheme
 - Take advantage on-page & bank-interleave (bank-group)
 - Know DRAM controller address map (page size)
- Local buffering to resolve conflict
 - Minimize random, short access
 - Streamline DRAM access – longer read/write burst
 - Resolve read-write turn-around

Adaptive Linear Address mapping:

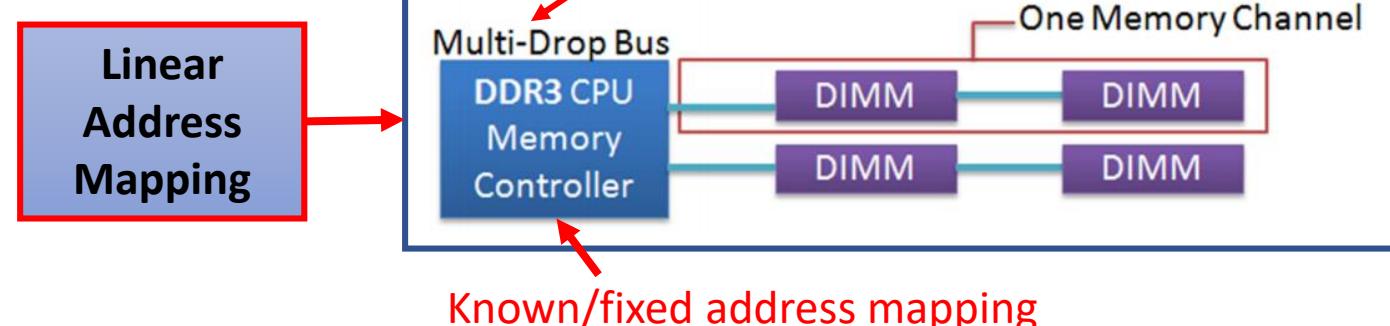
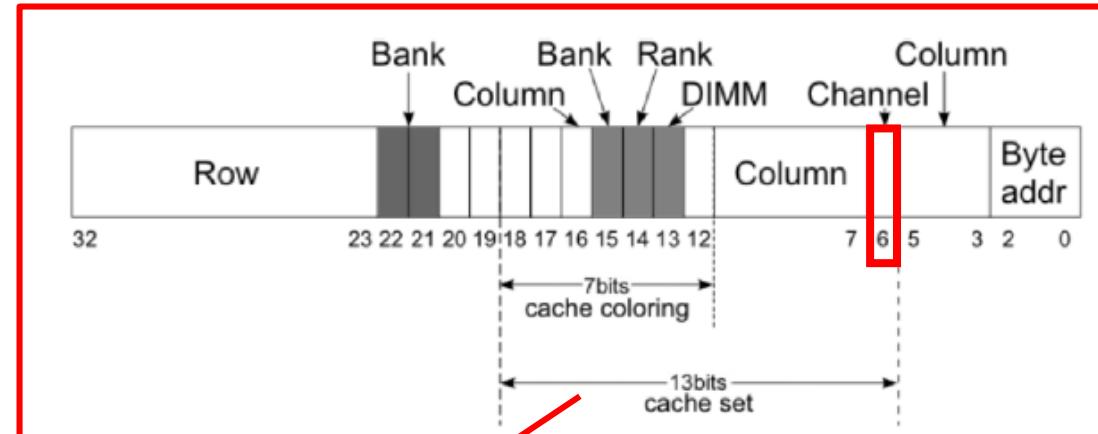
<http://epub.vgu.edu.vn/bitstream/dlibvgu/90/1/Adaptive%20linear%20address%20map%20for%20bank%20interleaving%20in%20DRAMs.pdf>



Adaptive Linear Address Mapping

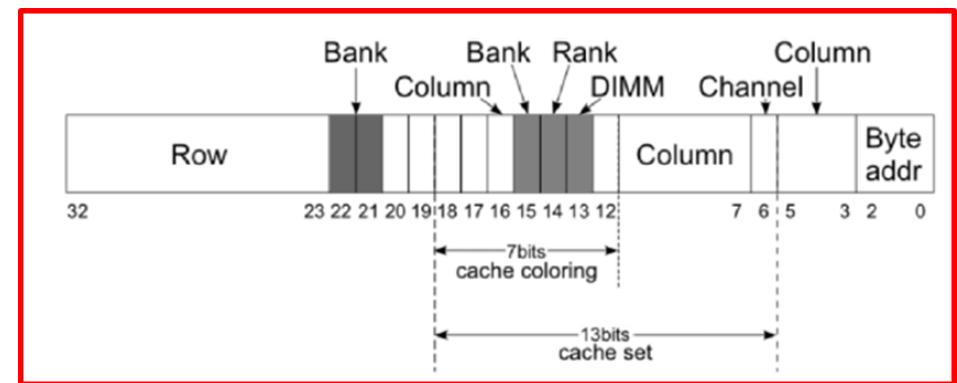
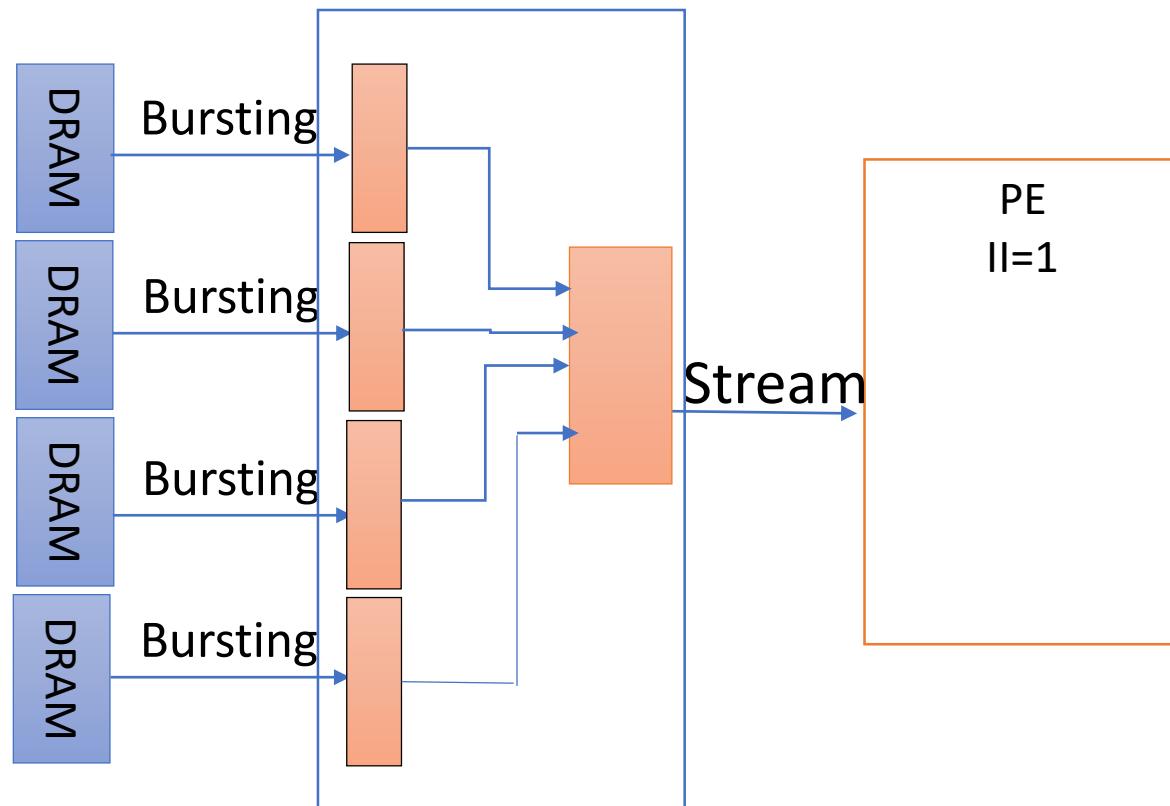


- Eliminate bank interfere by linear address mapping
 - Multiple Kernels
 - Non-linear access pattern
- Example
 - CPU cacheline size matches DRAM burst $8 = 64B$
 - Channel A[6] for interleaving
 - What if your dataset size 128B
 - Channel interleaves at 128B



Memory Stripping

Lab-SDRAM



- Use multiple memory bank to increase bandwidth
- Stripping/Interleave across multi-channel memory
- Join or split data stream from multiple memory interfaces

AXI Interface to access HBM

- 8GB HBM memory
- 32 HBM Banks/Pseudo Channels (PC) – each 256MB
- 32 AXI channels (PC) communicate with FPGA using segmented crossbar switch
- 14.375 GB/s max theoretical bandwidth per PC
- 460 GB/S ($32 * 14.375$ GB/s per PC)
- Note: 14.375GB is less than 19.25GB/s for a DDR channel
 - Use multiple AXI masters efficiently into the HBM subsystem.
- 32 AXI channels / AXI switch network
- 256bit data width
- Flexible Addressing
- **RAMA (Random Access Master Attachment) to improve throughput**
 - Random access across multiple HBM banks
 - Techniques: Resizing burst, Multiple outstanding requests, response re-ordering

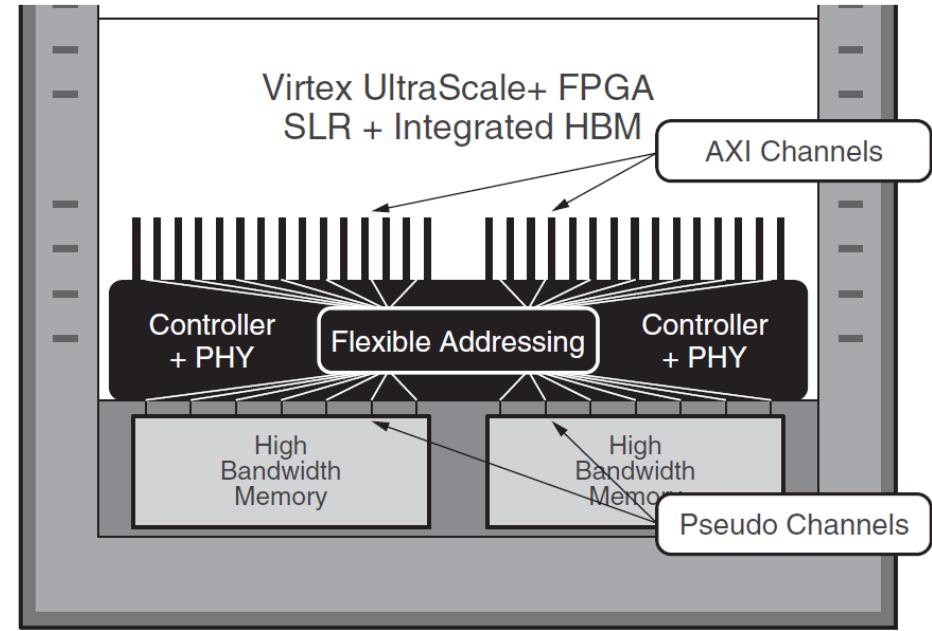
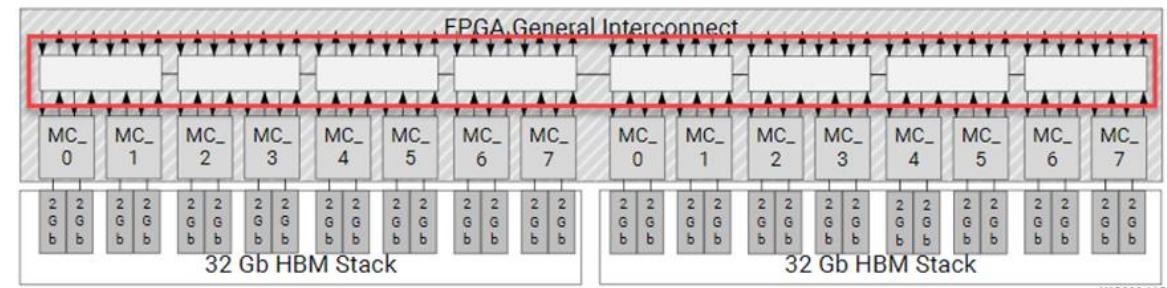


Figure 5: AXI Interfaces (to User Logic) and HBM Pseudo Channels (to HBM Stacks)



HBM Controller Features

- Memory performance
 - Configurable access reordering to improve bandwidth utilization
 - Reordering transactions with different IDs
 - Honors ordering rules within IDs
 - Read after Write and Write after Write coherency checking for transactions generated by same master with same ID
 - Refresh cycles are handled by the controller
 - Temperature controlled refresh rates
 - Optional hidden single row refresh option to minimize overhead
 - Increase efficiency based on user access patterns
 - Flexible memory address mapping from AXI to the physical HBM stacks
 - Configurable reordering and memory controller behaviors to optimize latency or bandwidth
 - Grouping of Read/Write operations
 - Minimizing page opening activation overhead
 - Performance monitoring and logging activity registers
 - Bandwidth measured at each HBM memory controller
 - Configurable sample duration
 - Maximum, minimum, and average Read and Write bandwidth is logged
- Reliability (RAS) support
 - Optional SECDED ECC
 - Partial word writes supported with Read Modify Write (RMW) operation
 - Background scan of memory for error scrubbing
 - Correctable ECC errors found are fixed and updated in memory
 - Optional parity with memory access retry due to data parity errors in Write operations
 - Parity data protection available in datapath between user logic and HBM
 - The external parity uses the 32-bit WDATA_PARITY and RDATA_PARITY buses with one parity bit per byte of data.
 - Uses Odd parity where a 1 is asserted for every data byte when the sum of the bits in the byte is an odd value.
 - Error logging registers
- Power management
 - Per memory channel clock gating
 - Per memory channel divided clock rate for reduced power
 - Power down mode supported
 - Optional self-refresh mode to retain contents of memory
 - Selectable idle timeout to self-refresh entry