Memory Systems

Memory Hierarchy

- Smaller and closer are faster
  - Flipflops
  - Register files
- Larger and further away are cheaper
  - DRAM
  - Hard disks

	STORAGE	DELAYS: Absolute	DELAY: in CPU clocks
Fastest	Isolated latches	a few psecs to a few nsecs	<< 1 CPU clock
	Registers/ Small RAM	a few 10s of psecs to a few nsecs.	< 1 CPU clock
	Large on–chip SRAM	a few nsecs.	< 1 to a few CPU clocks
	Off chip SRAM	a few nsecs. to 20 nsecs.	
	Off–chip DRAM	40 nsecs. and higher **NEW non–volatile**: P–RAM, STT–RAM*	
	Flash storage	tens of microsecs, slower writes	
	Magnetic disks	6 to 14 msecs.	
Slowest	Optical disks	10 to 20 msecs.	


:on–chip storage resources

* price point unclear

Examples:

- Register file bandwidth for a 4–issue per cycle superscalar CPU running at 3 GHz.:

  – 8 operands, 32–bits wide per cycle => BW = 96 Gbits/sec.
  – Double this for 64–bit datapaths

- CPU I/O pin bandwidth: 128 I/O pins, 1 GHz. external memory bus: 32 Gbits/sec. – this is the best data rate one can realize from off–chip memory devices. Can double this by using two transfers per clock (one per clock edge, as in DDR memory interfaces)

- Typical disk transfer rate: several 10s to 125 MBytes/sec. (single disk unit, sustained data transfer rate)

- DRAM bandwidth: regular DRAM chips –
  - Single byte access (“byte–wide” DRAM chip): 1 byte/60 nsecs.
  - Internal BW based on a 1024–bit row size: 1K to 4K bits/40 nsecs.
  - Significant loss in data rate occurs as only a byte gets selected internally within the chip and gets sent out!
  - Burst or page mode data rate from DRAM is faster (roughly about 2 times individual byte access rates)

Characteristics of Memory Traffic

- Memory instructions vs. all instructions:
  - RISC: 15% to 20%
  - CISC (x86): 35%
- Loads vs. Stores: 2 to 1
- Caches make external memory access bursty (smoother would be better)
- Strided accesses in many applications

Memory Performance

- Processor-memory performance gap
- Latency = Request & Response time
- Bandwidth = Steady-state transfer speed
- Effective transfer time:

  latency + (data_size / bandwidth)

- Bandwidth easier to improve than latency

Techniques for Improving Memory Performance

- Memory interleaving
- Cache memory
- Newer memory interfaces
- Prefetching
- Stream buffers
- Store bypassing by loads and predicting store bypassing (“dynamic memory disambiguation”)
- Cache miss prediction
- Simultaneous multithreading

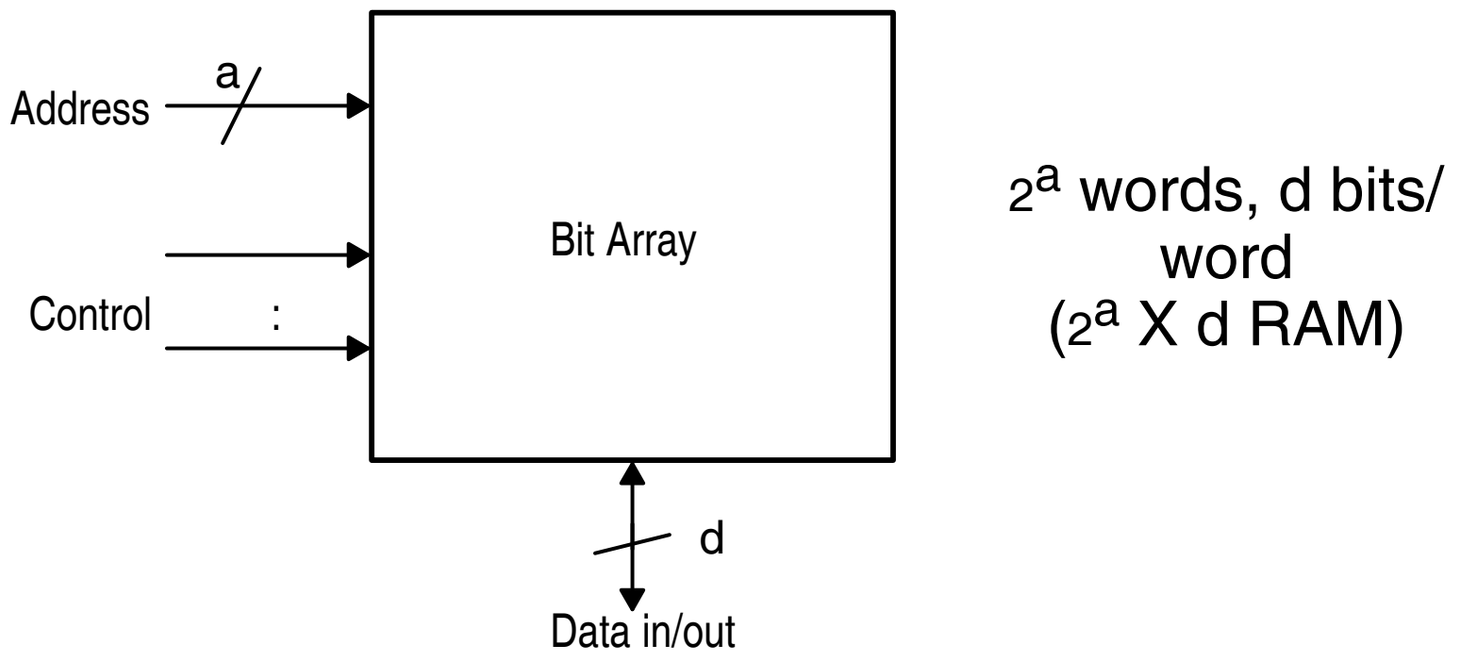
RAM Devices

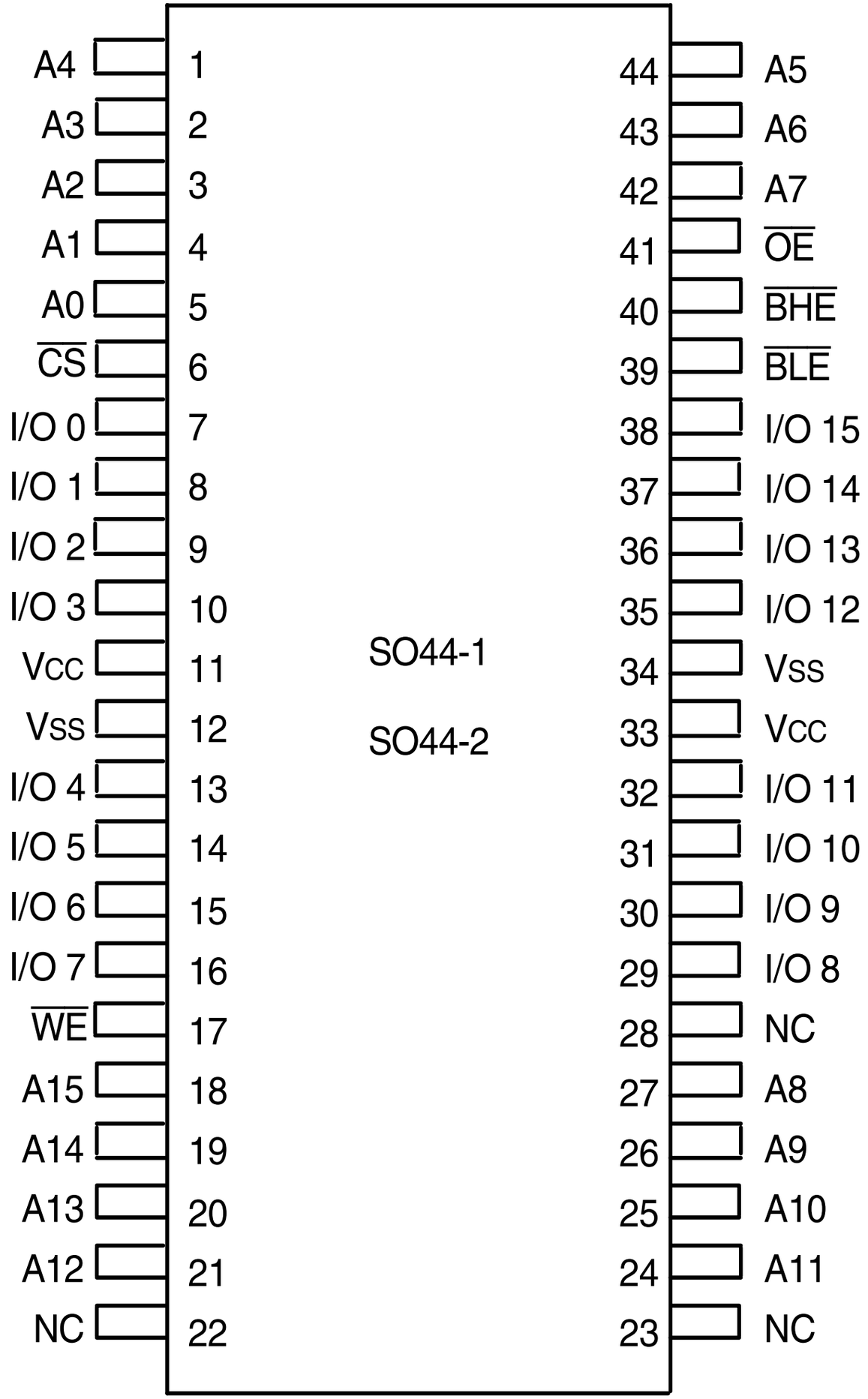
- Basic storage component: *bitcell* – one bit of storage:

- Static RAM (SRAM) bitcell = flip–flop (back–to–back connected inverters)

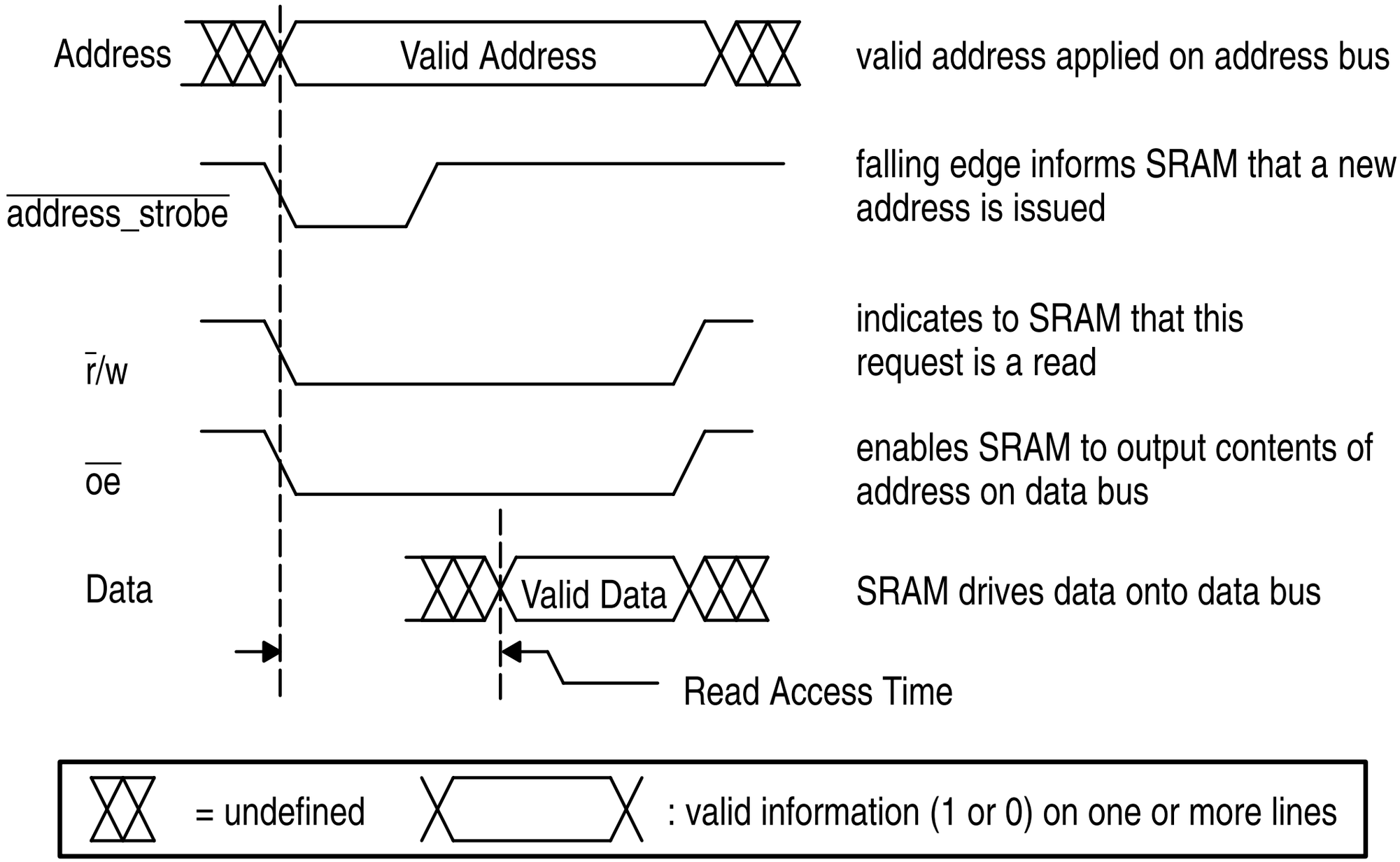
- Dynamic (DRAM) bitcell = MOS capacitor: charged => 1, discharged => 0

- The RAM as a black box:

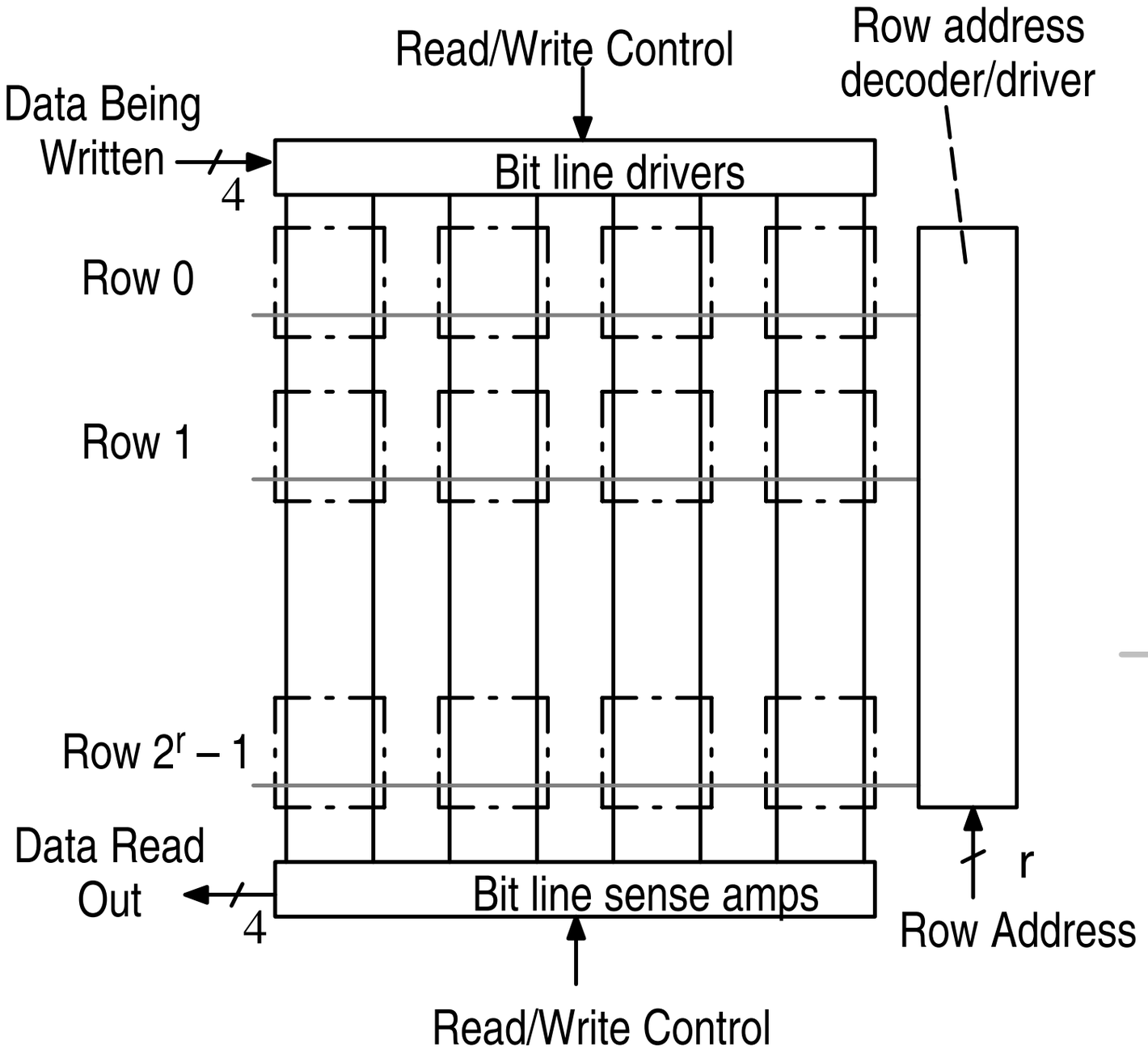


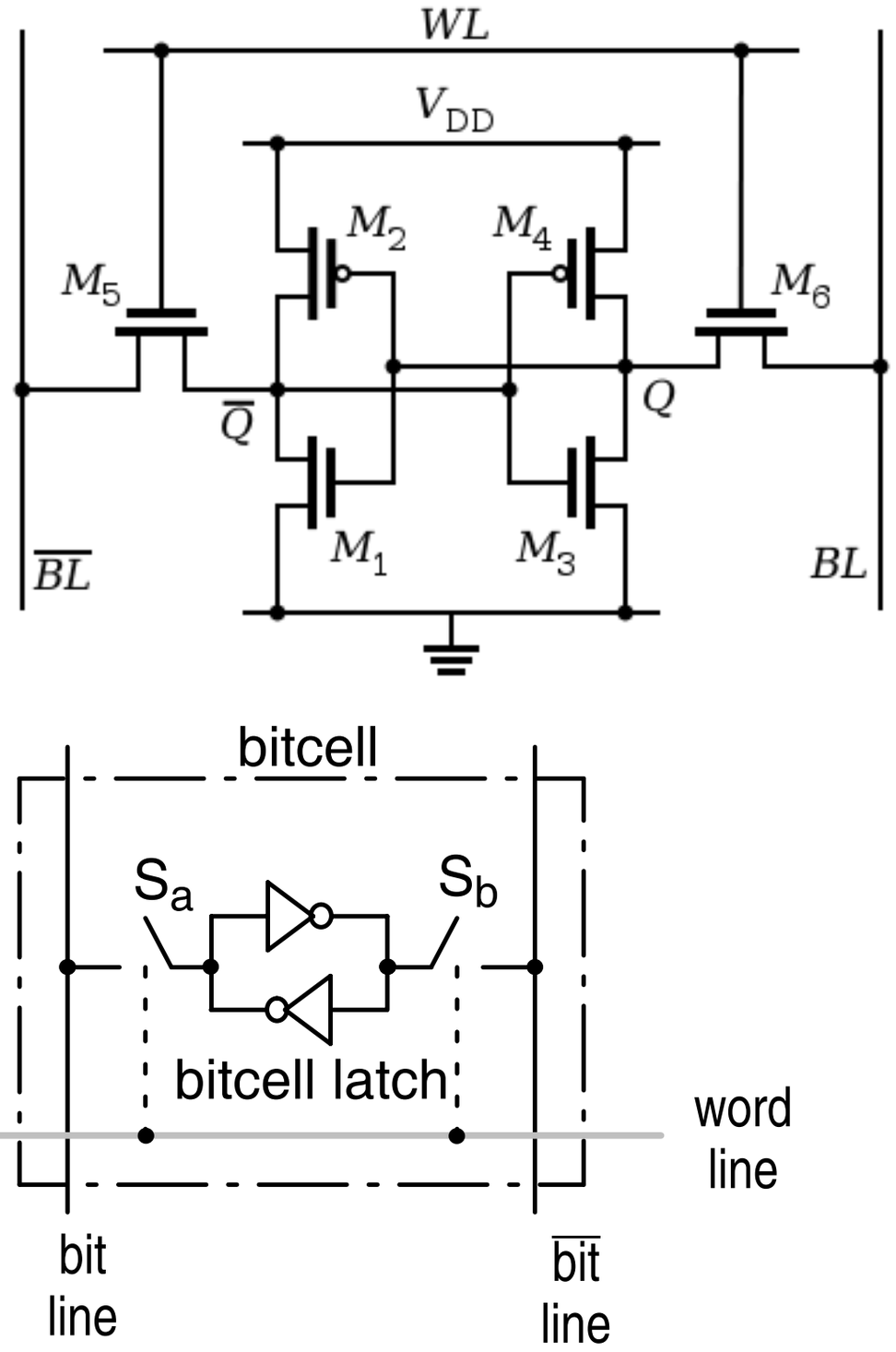

A4 1
A3 2
A2 3
A1 4
A0 5
CS 6
I/O 0 7
I/O 1 8
I/O 2 9
I/O 3 10
Vcc 11
Vss 12
I/O 4 13
I/O 5 14
I/O 6 15
I/O 7 16
WE 17
A15 18
A14 19
A13 20
A12 21
NC 22
44 A5
43 A6
42 A7
41 OE
40 BHE
39 BLE
38 I/O 15
37 I/O 14
36 I/O 13
35 I/O 12
34 Vss
33 Vcc
32 I/O 11
31 I/O 10
30 I/O 9
29 I/O 8
28 NC
27 A8
26 A9
25 A10
24 A11
23 NC
SO44-1
SO44-2

Asynchronous static RAM chip

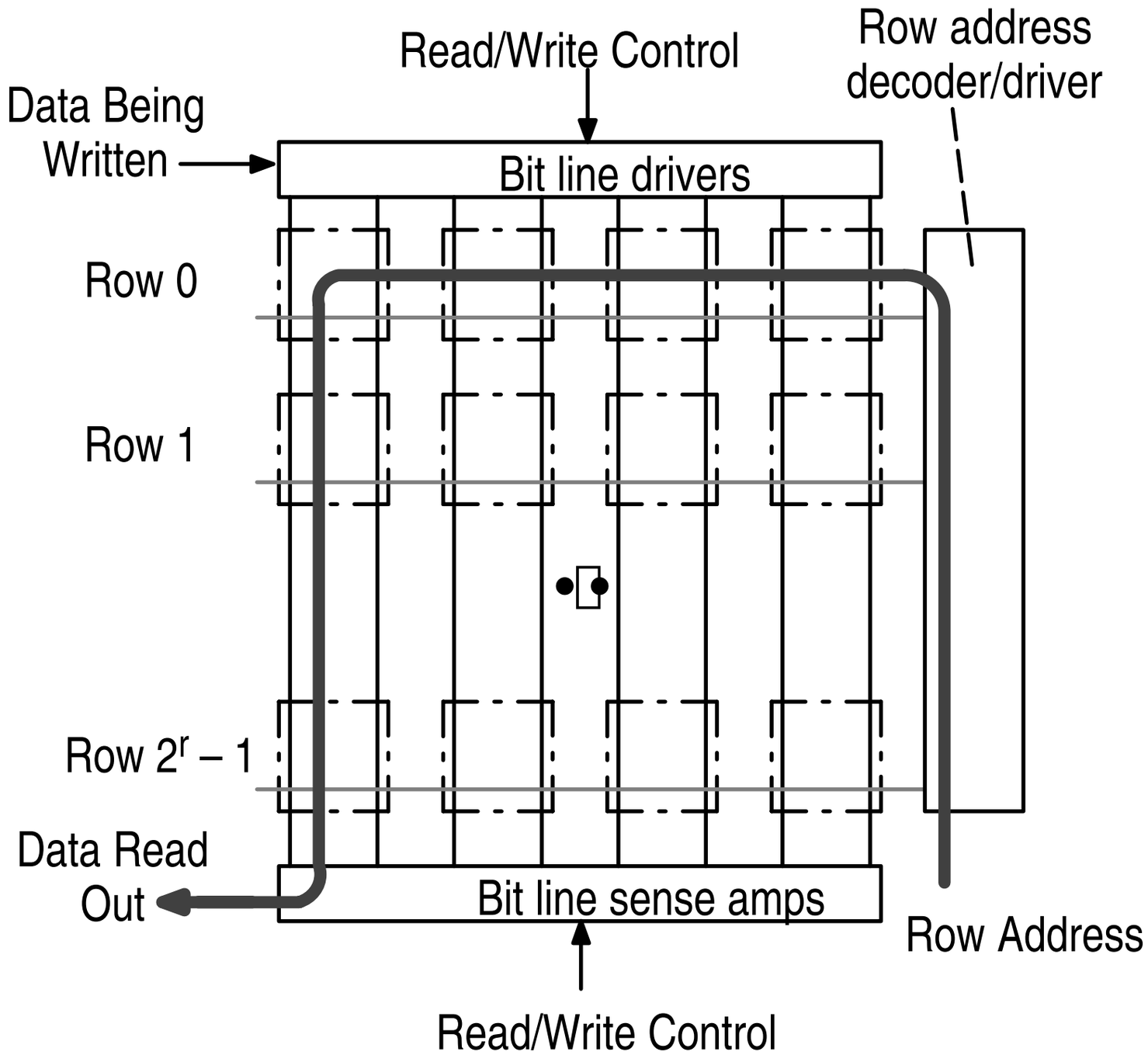


SRAM Internal Organization





- What are the components of the SRAM access time?



- The above figure shows the longest delay path in a SRAM (in gray)

- Between the application of an address and control signals and the time the data is read out, the delays in the path shown are:

  $t_{decoder}$ = the delay of the decoder and the driver of the word lines: this is proportional to the # of address bits (a)

  t_{row_line} = the delay of the row wire – from the decoder to the farthest end: this is proportional to the # of bits/word (d)

  t_{bit_line} = the delay of the bit lines – from the row farthest from the sense amp to the sense amp: this is proportional to the number of words (2^a)

  t_{sense} = the delay of the sense amp

- The read access time is thus ($t_{decoder} + t_{row\_line} + t_{bit\_line} + t_{sense}$)

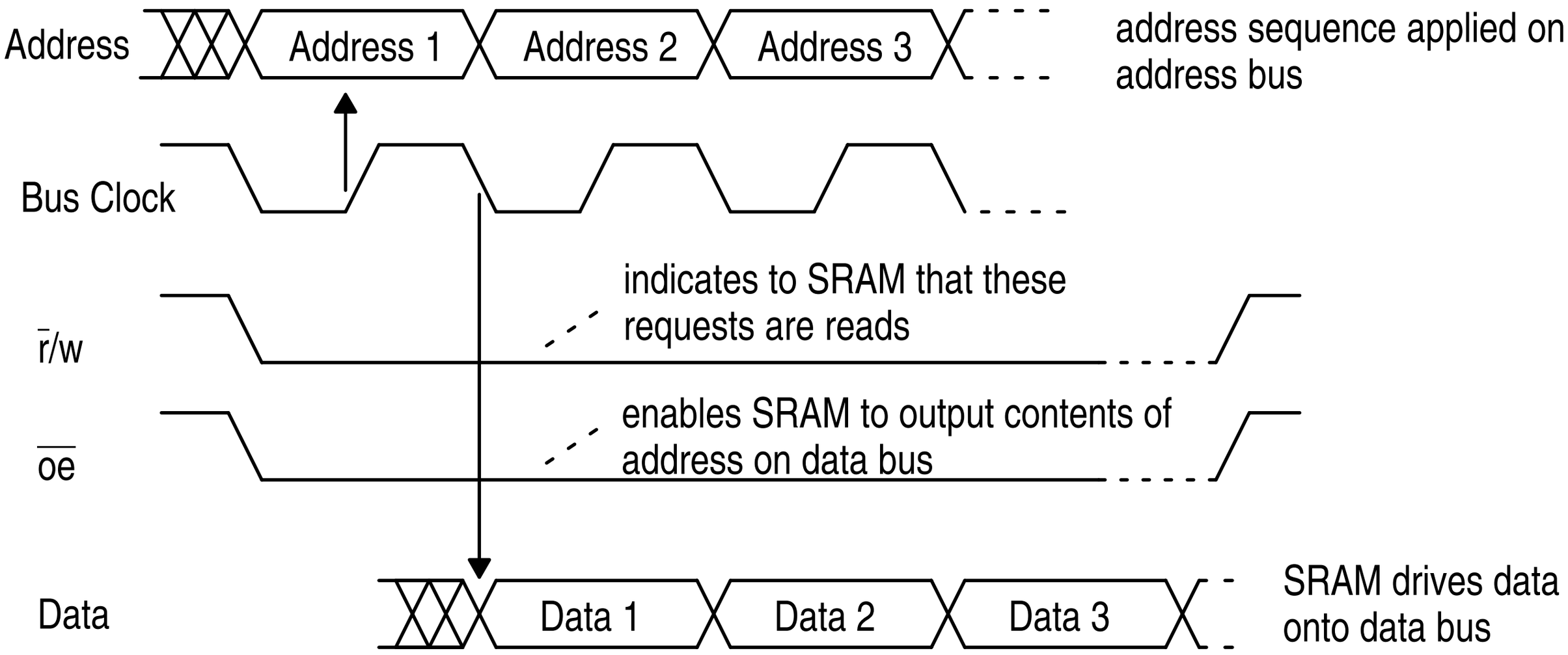
  - Note how the access time depends on the dimensions of the array and why "smaller is faster" holds in this case.

  - Possible ways to speed up the SRAM access time:

  (a) Reduce the array dimensions

  (b) Use smaller transistors (i.e., a more aggressive fabrication process): this reduces the various proportionality constants.

Pipelining SRAM access



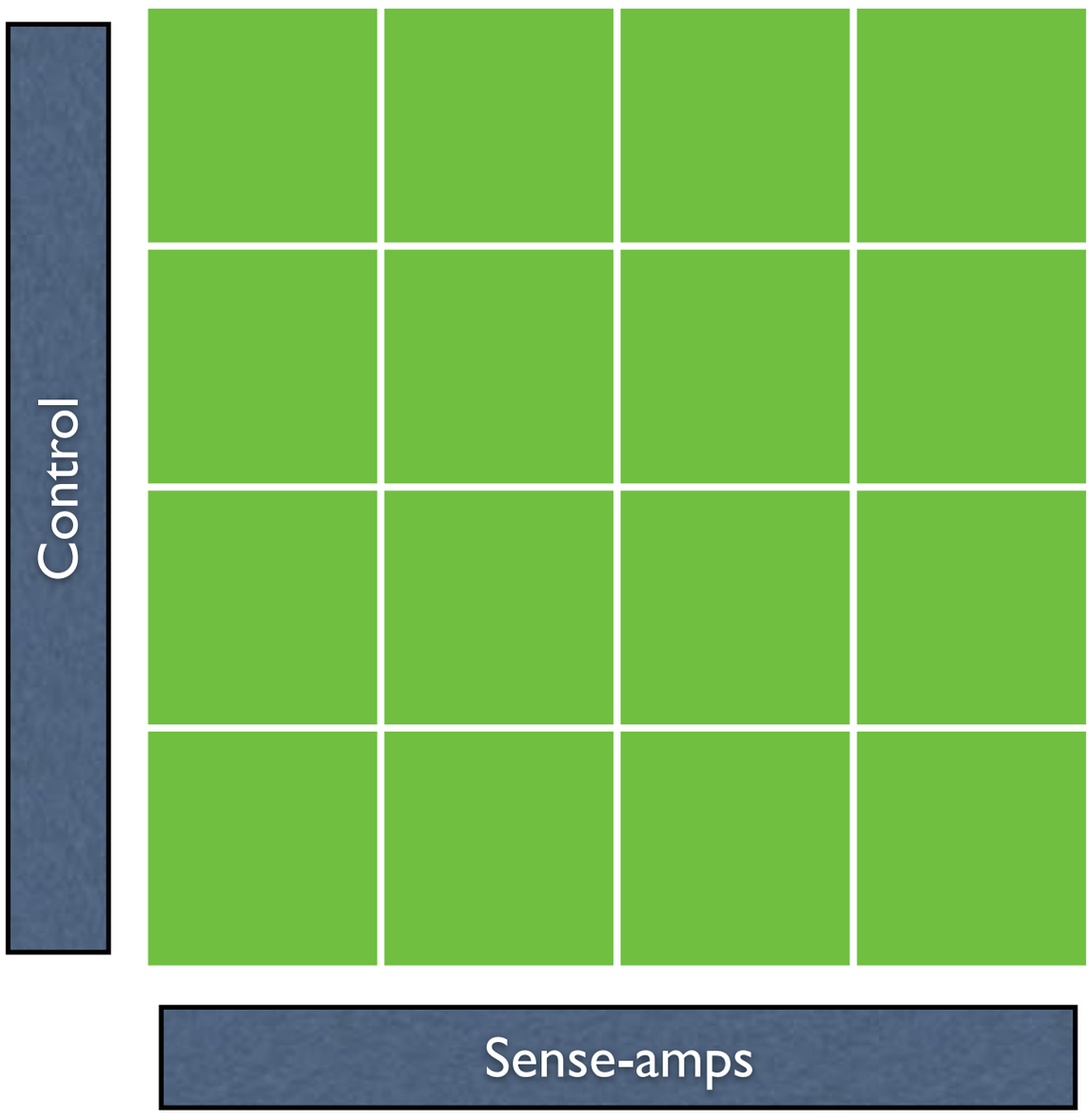
What do we make from SRAMs?

- Physical register files
- Register alias tables
- Reorder buffers
- Scoreboards
- Instruction Queues

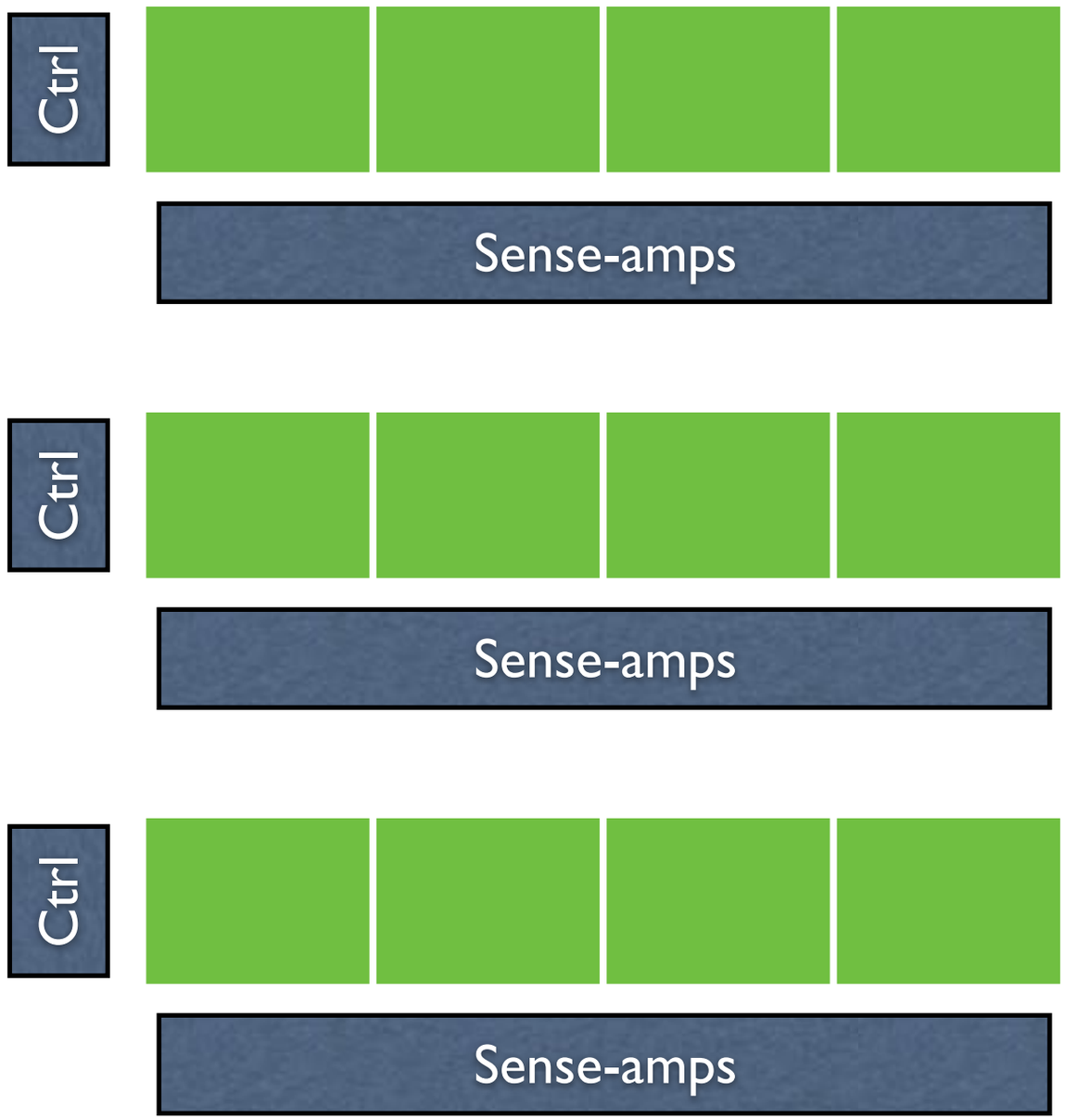
Register files and ILP

- Superscalar processors perform many simultaneous accesses to a register file
- More ports requires more area
- SRAM delay is a function of area

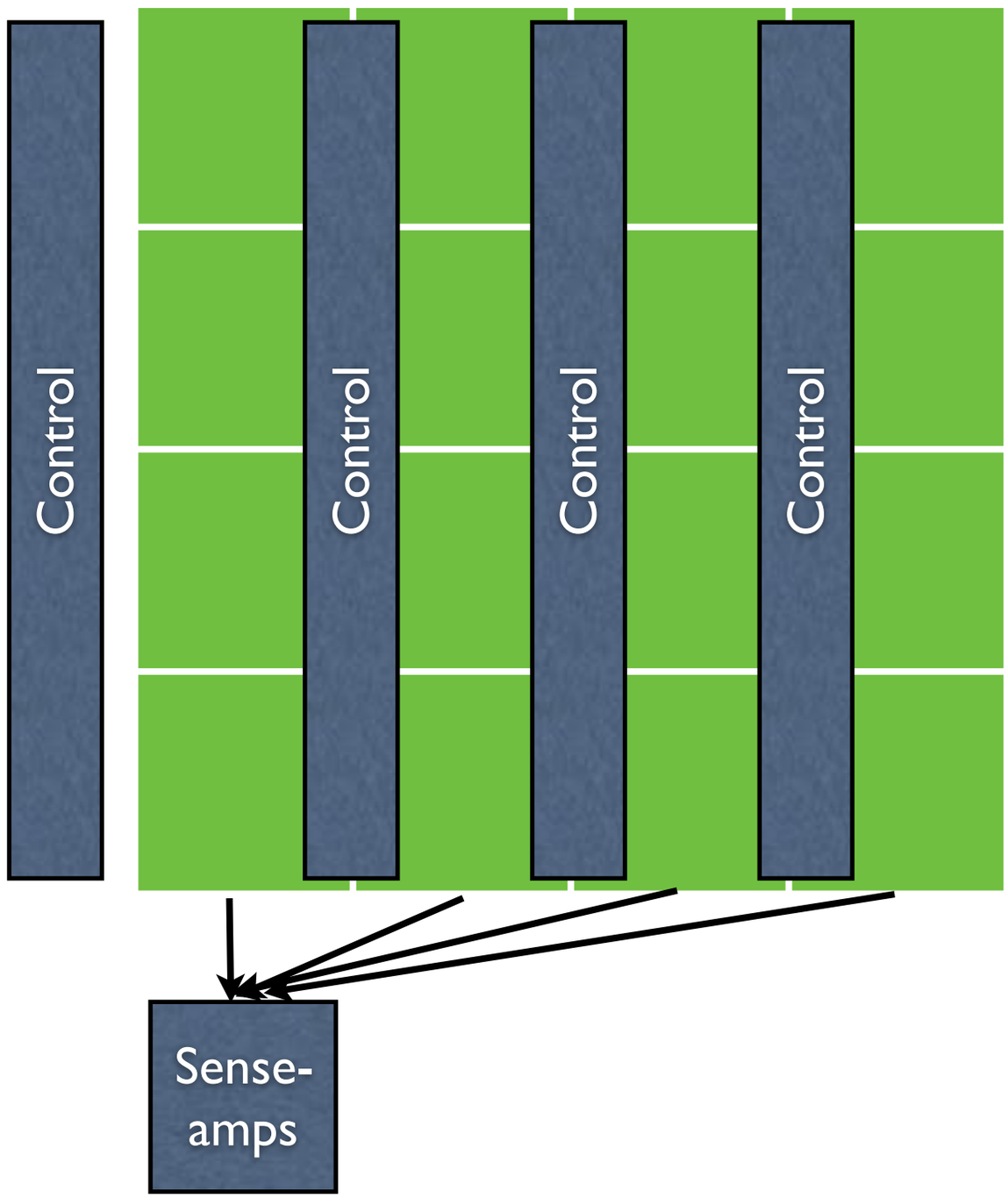
Sub-banking (vertical stripes)



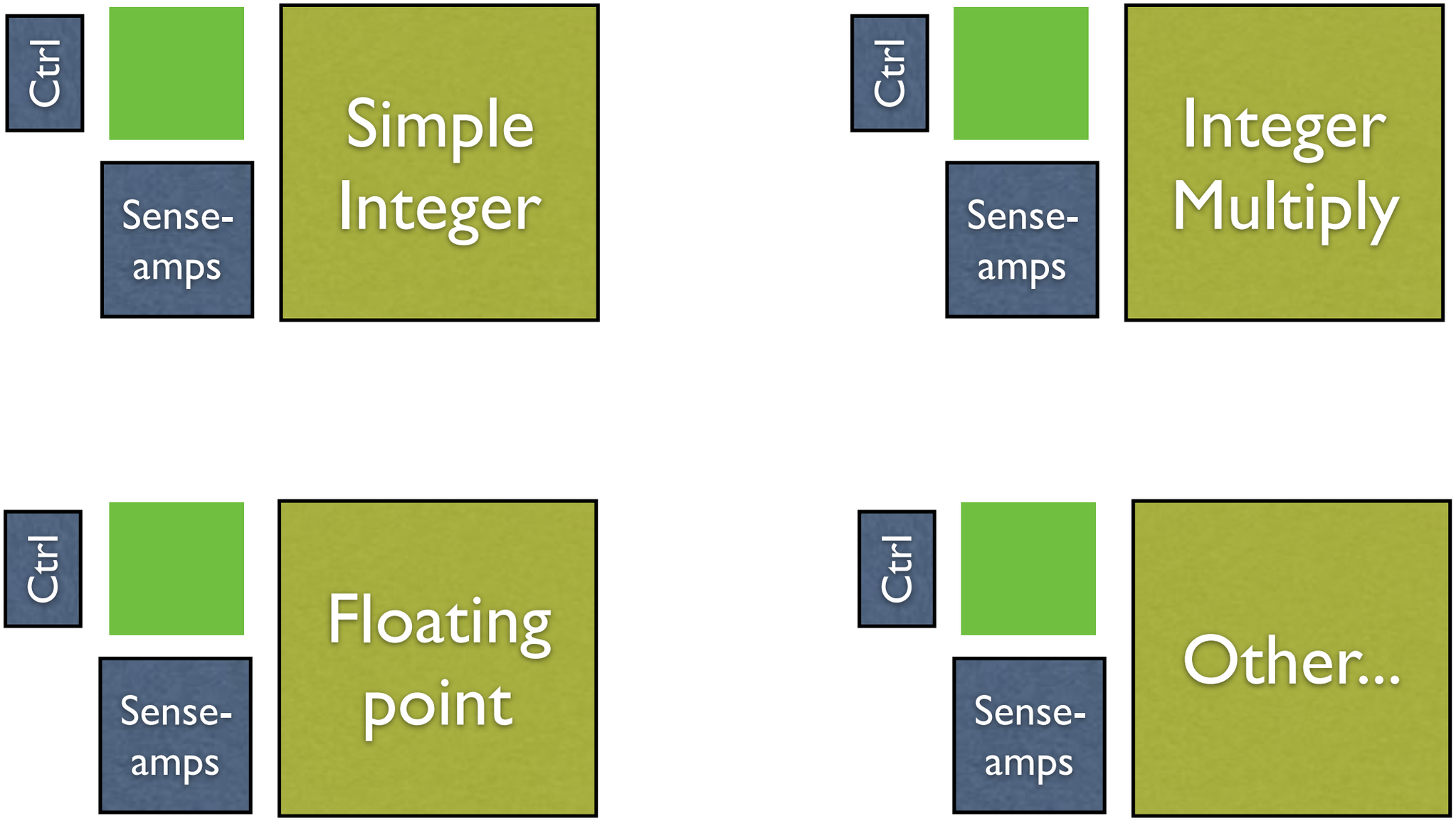
Multi-banking (horizontal stripes)



Sub-banking (vertical stripes)



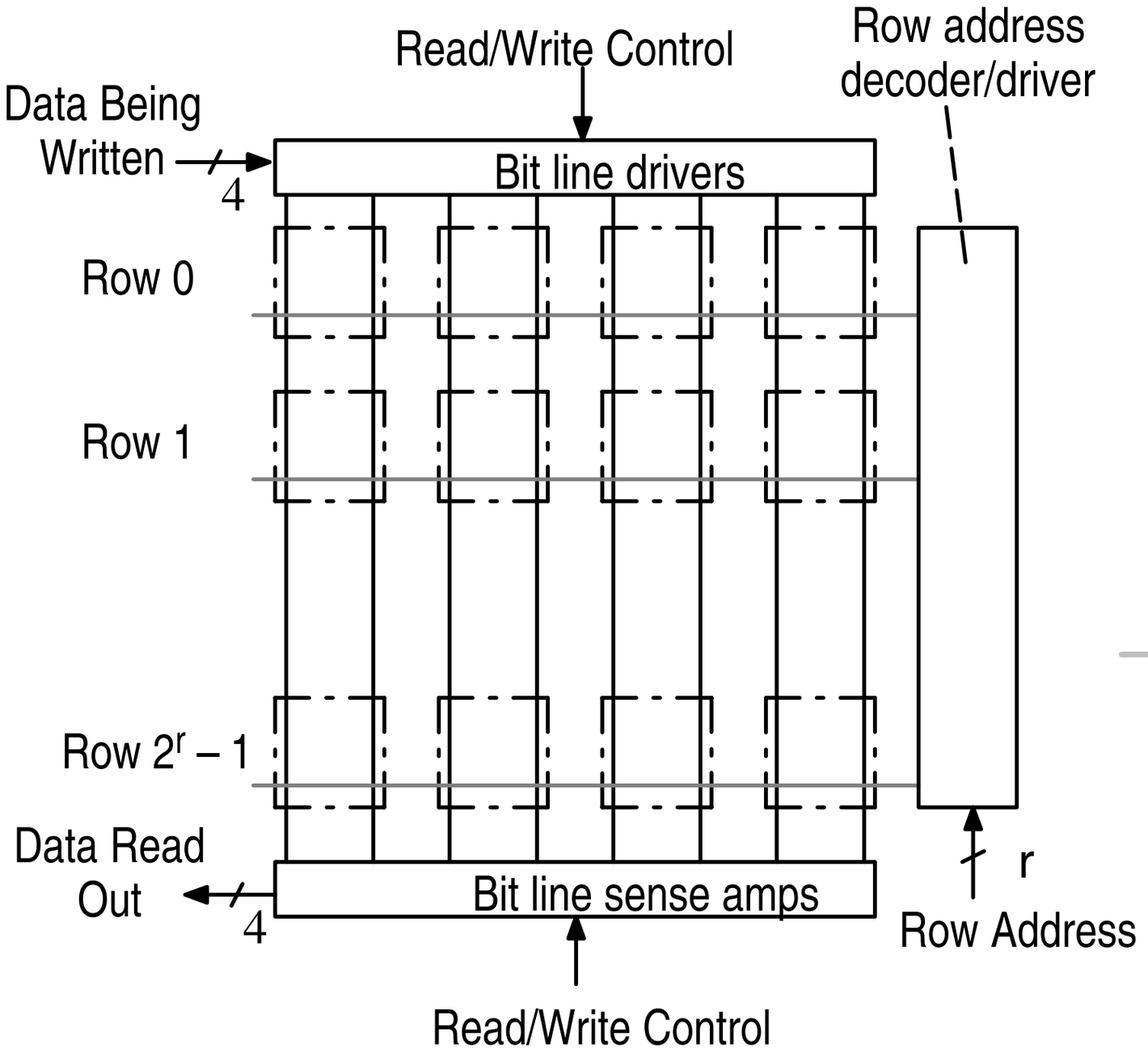
Clustered

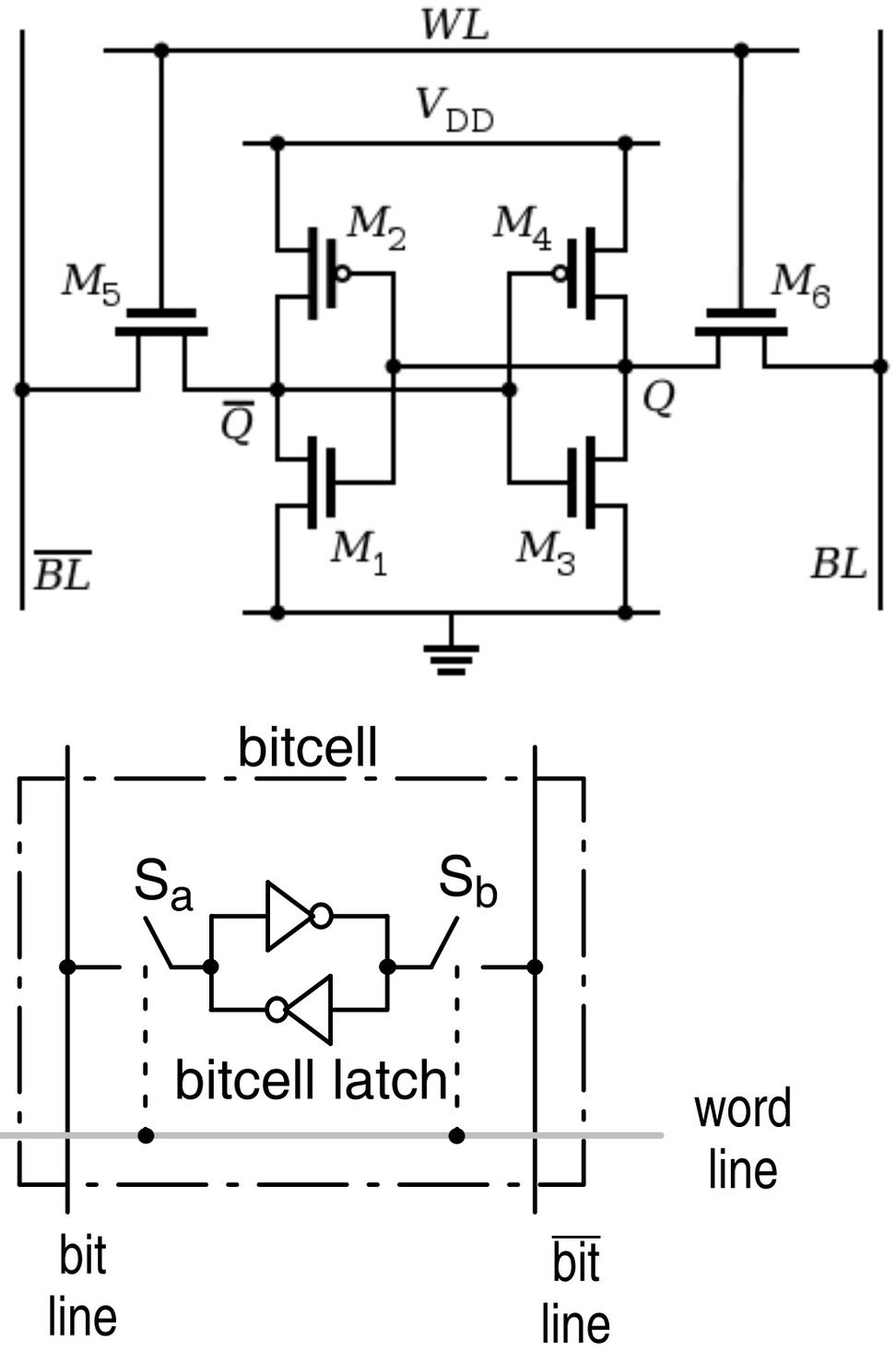

Ctrl
Sense-amps
Simple Integer
Ctrl
Sense-amps
Integer Multiply
Ctrl
Sense-amps
Floating point
Ctrl
Sense-amps
Other...

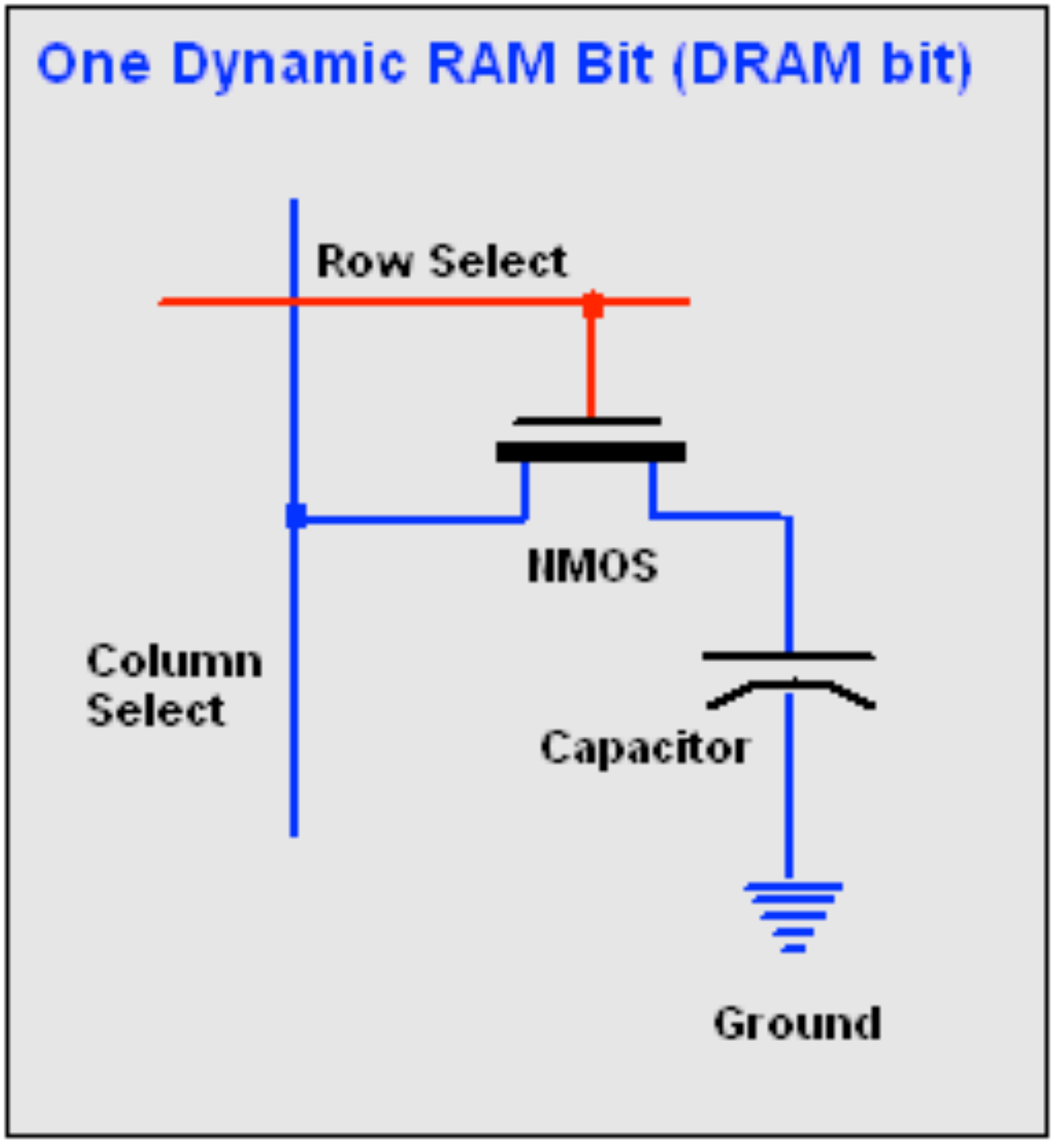
DRAM Organization

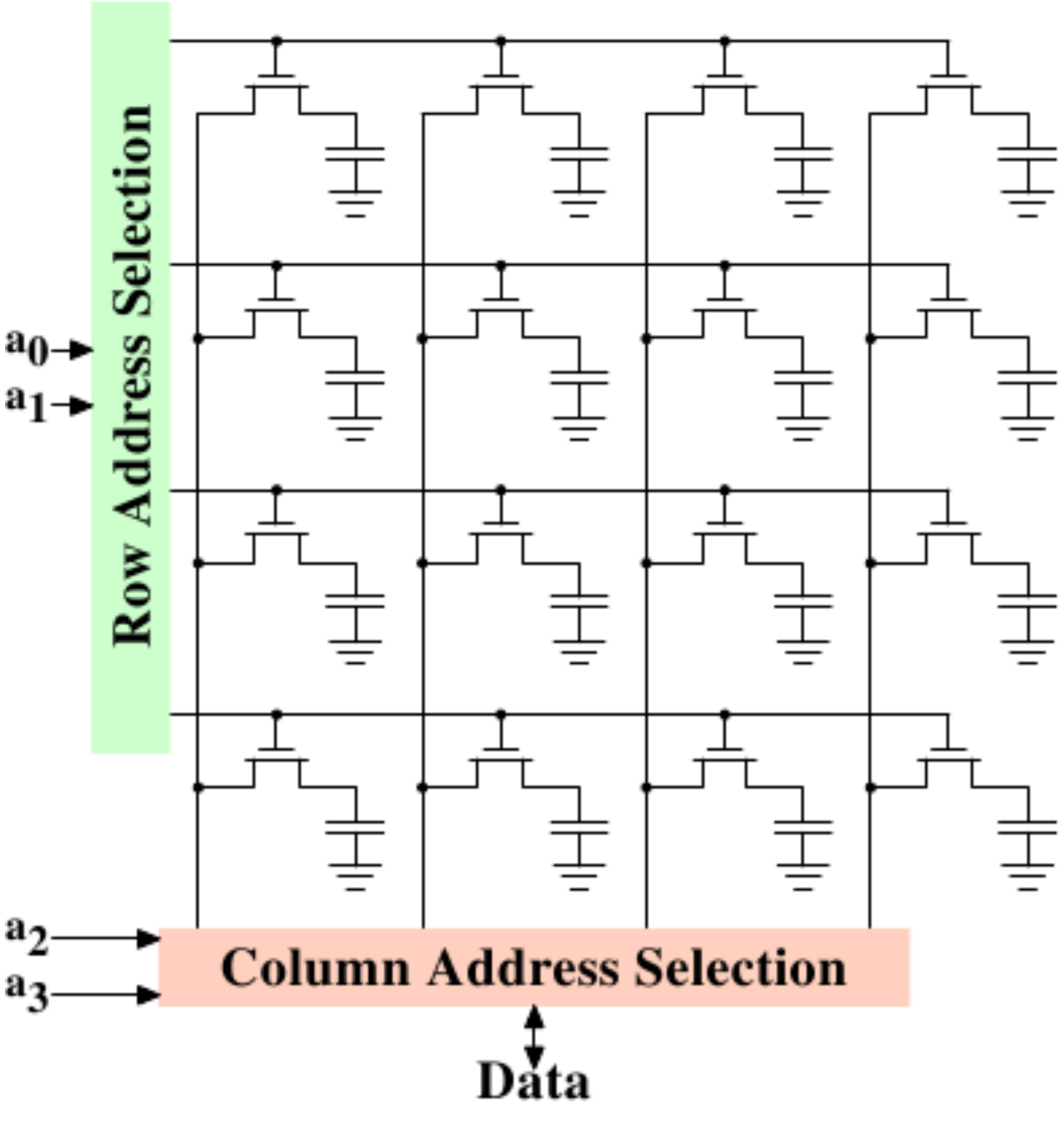
- Stored as 2D array with very long rows (e.g. 8K bits)
- “Activating” copies row into row buffer
- Reads and write columns in row buffer
- “Precharge” copies row back into array

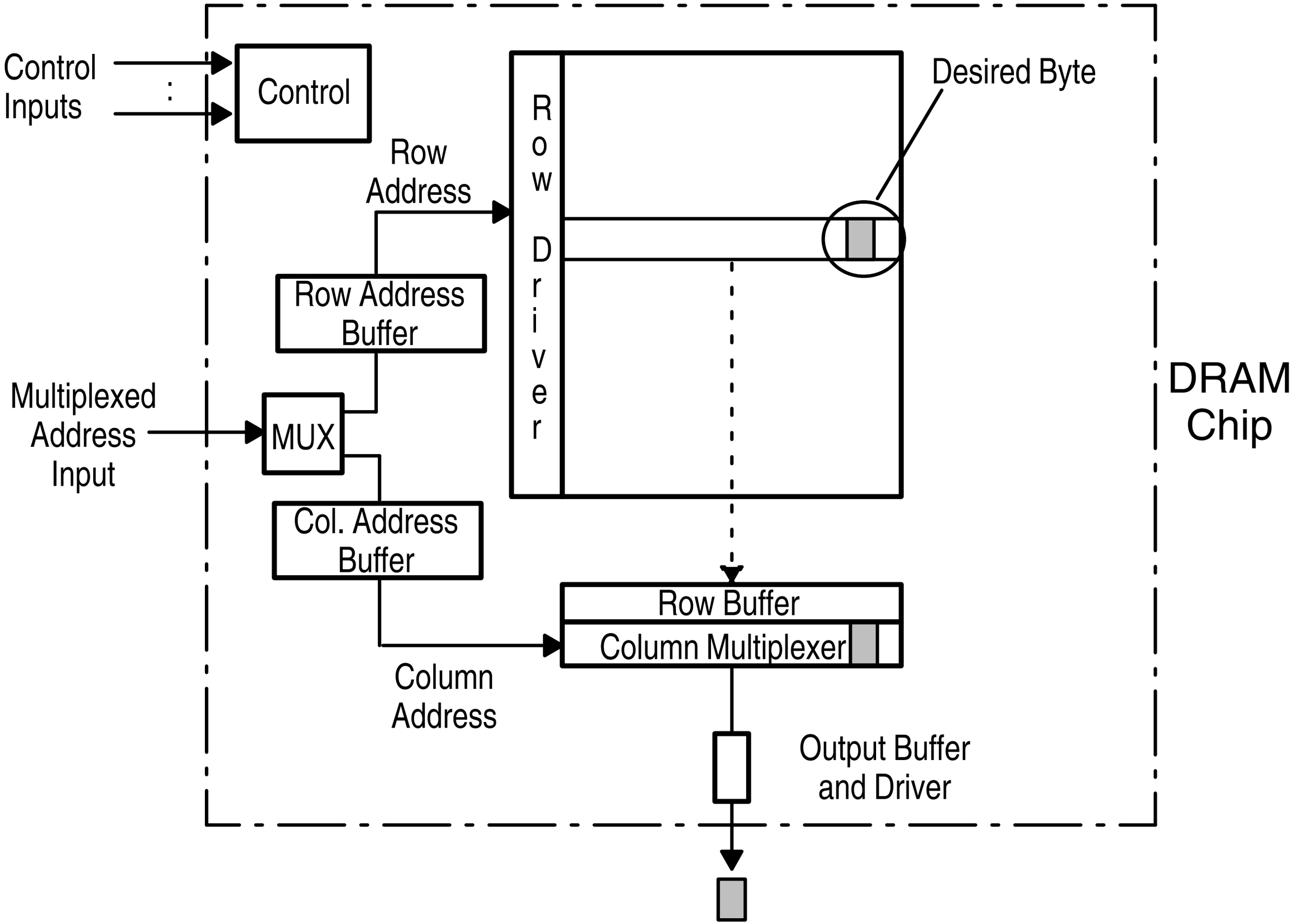
SRAM Internal Organization

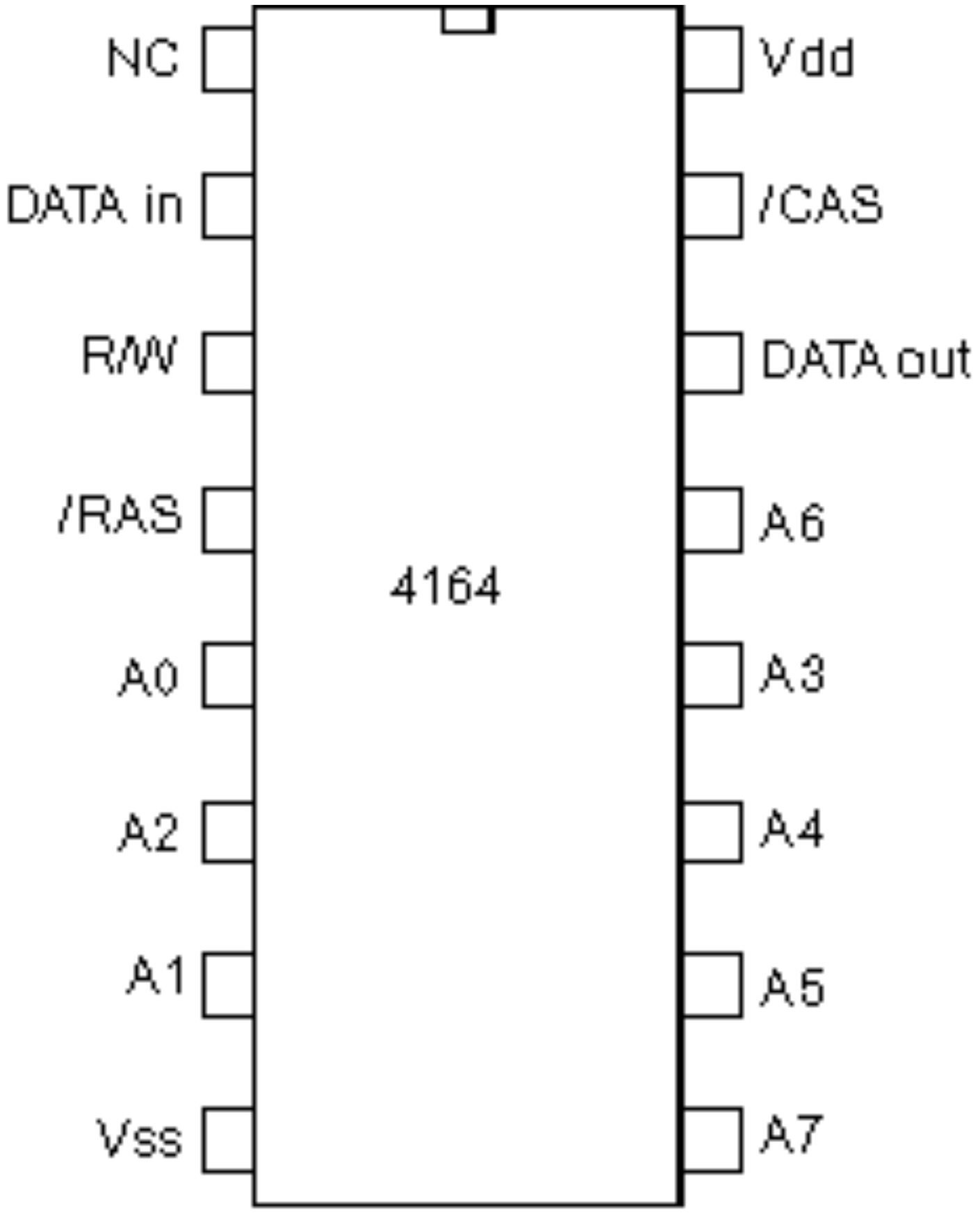


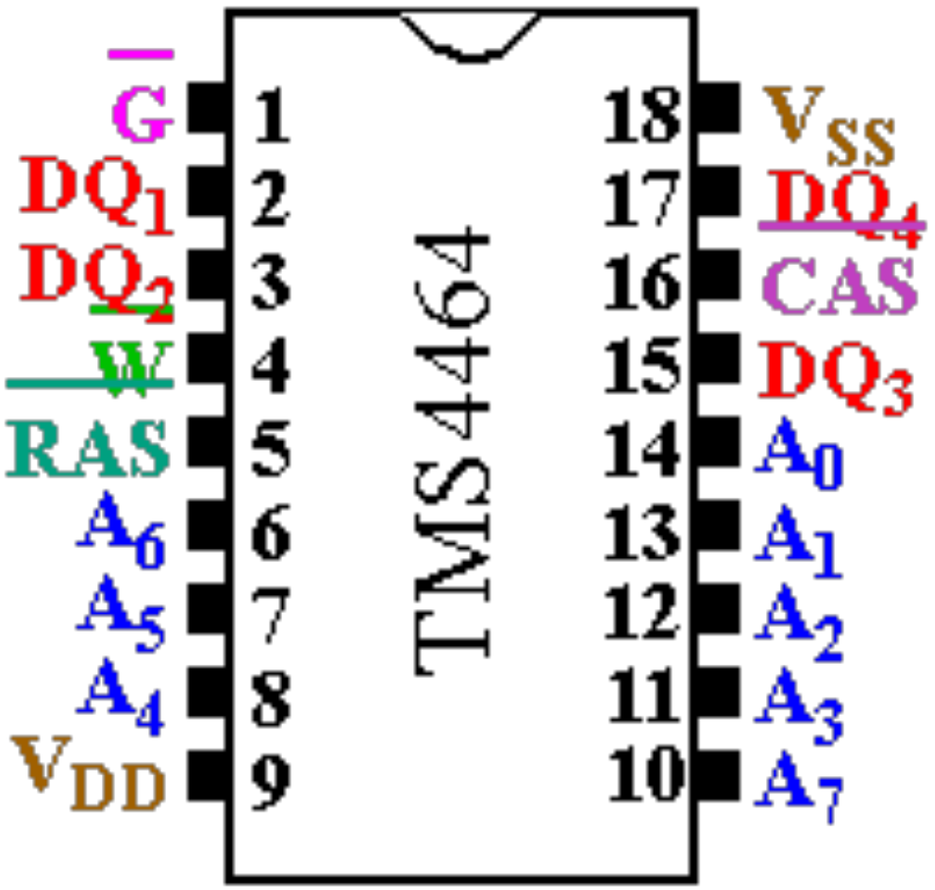





Row Address Selection
a_0
a_1
a_2
a_3
Column Address Selection
Data

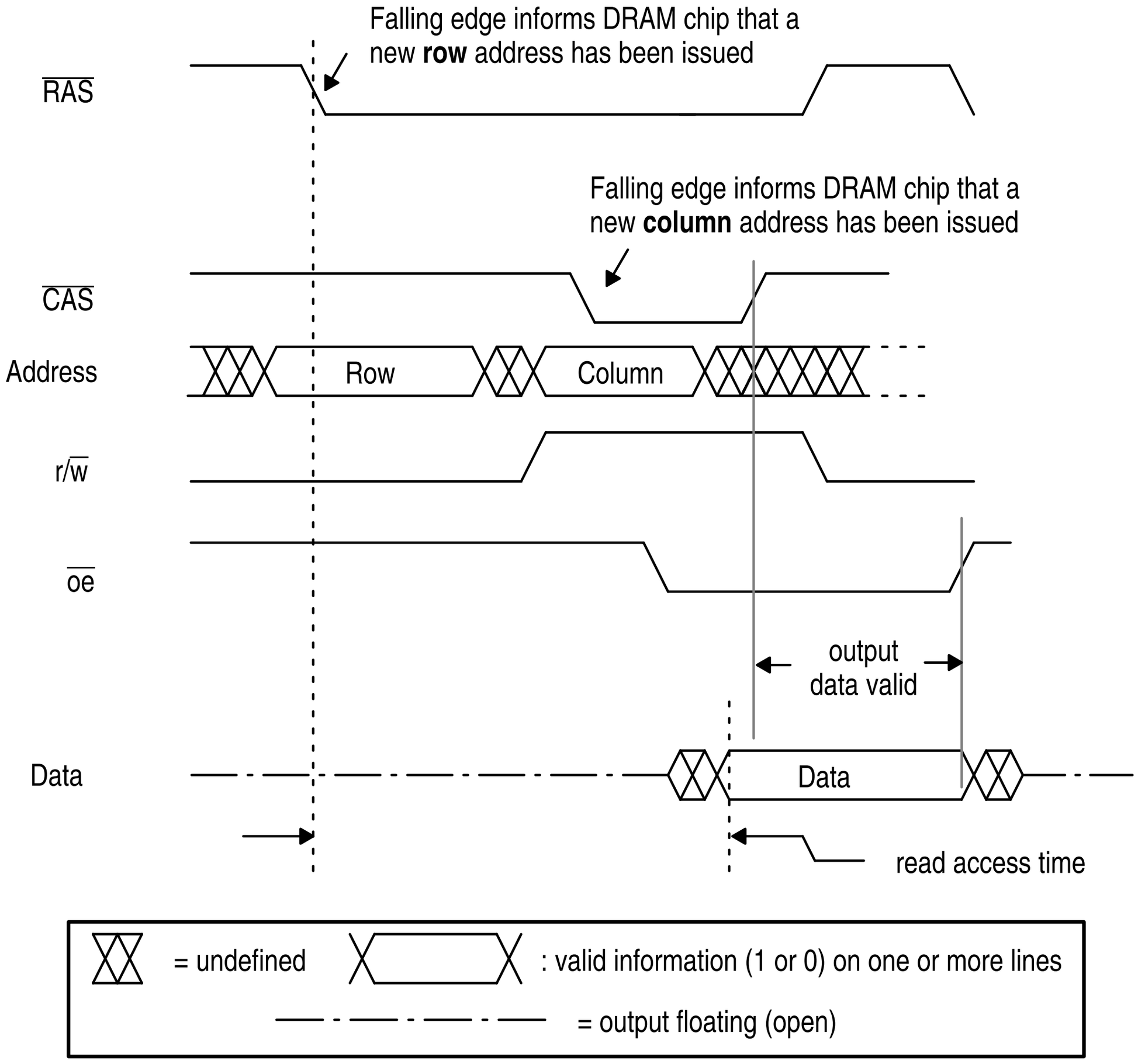

Control Inputs
Control
Row Address
Row Address Buffer
Multiplexed Address Input
MUX
Col. Address Buffer
Column Address
Row Driver
Desired Byte
Row Buffer
Column Multiplexer
Output Buffer and Driver
DRAM Chip


NC
DATA in
R/W
/RAS
A0
A2
A1
Vss
Vdd
/CAS
DATA out
A6
A3
A4
A5
A7
4164



64K x 4 DRAM

Pin(s)	Function
A_0-A_7	Address
DQ_0-DQ_4	Data In/Data Out
$\overline{RAS}$	Row Address Strobe
$\overline{CAS}$	Column Address Strobe
$\overline{G}$	Output Enable
$\overline{W}$	Write Enable


Falling edge informs DRAM chip that a new **row** address has been issued
RAS
Falling edge informs DRAM chip that a new **column** address has been issued
CAS
Address
Row
Column
r/w
oe
output data valid
Data
Data
read access time
= undefined
: valid information (1 or 0) on one or more lines
= output floating (open)

DRAM Facts

- DRAM Bit cells **leak**
  - Require refresh every 10 to 15 microseconds
- A DRAM controller:
  - Must perform complex scheduling
  - Generates read, write, activate, precharge, and refresh commands
  - Schedules refresh
  - Performs error checking

Refresh Mechanism

- Read contents of row into row buffer before contents degrade
- Write row back into DRAM array
  - Replenishes charge in cells holding ‘1’

Refresh Cycles

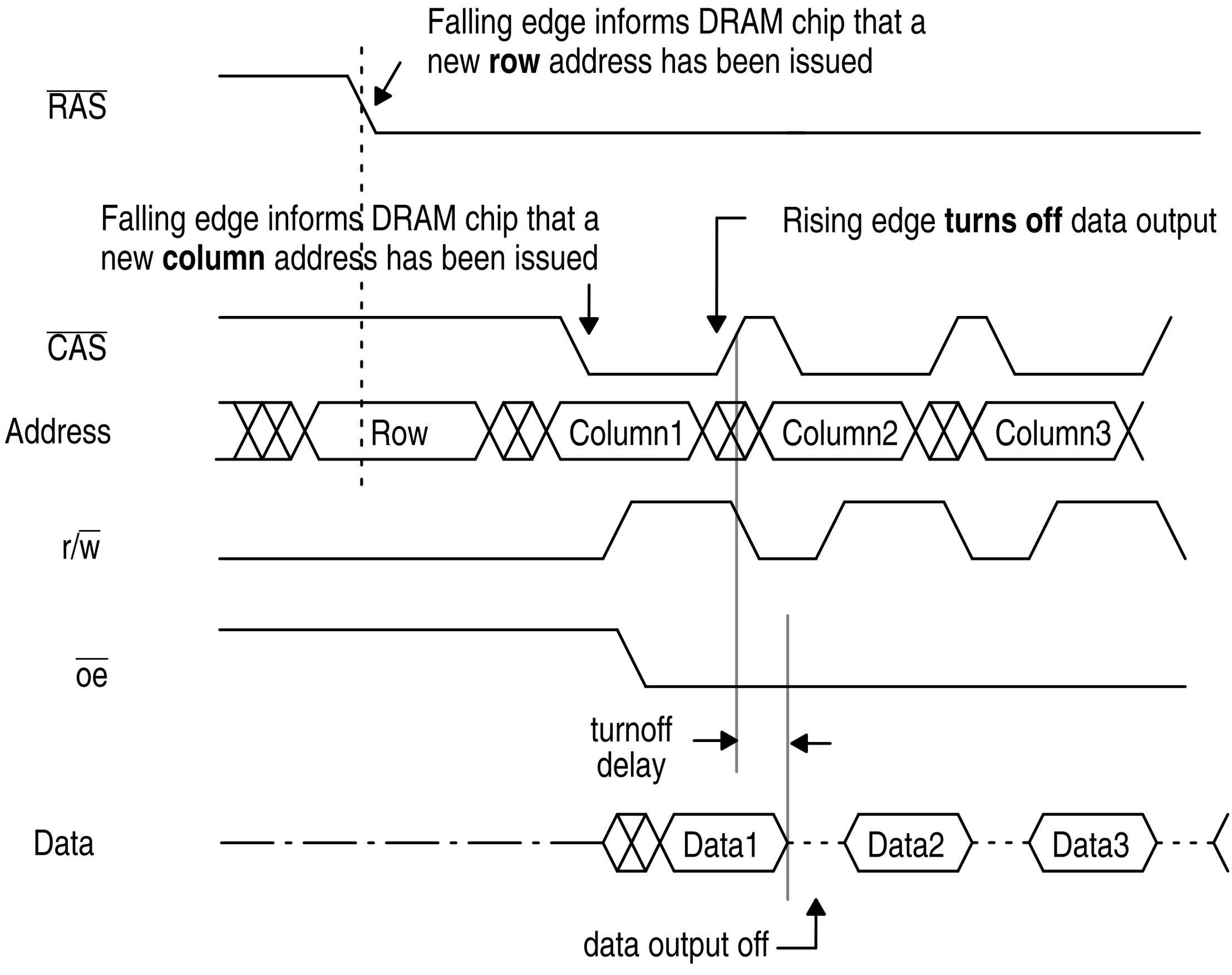
- Distributed mode -- Evenly spreads out refreshed amid other activity
- Burst mode -- refresh many rows consecutively during idle periods

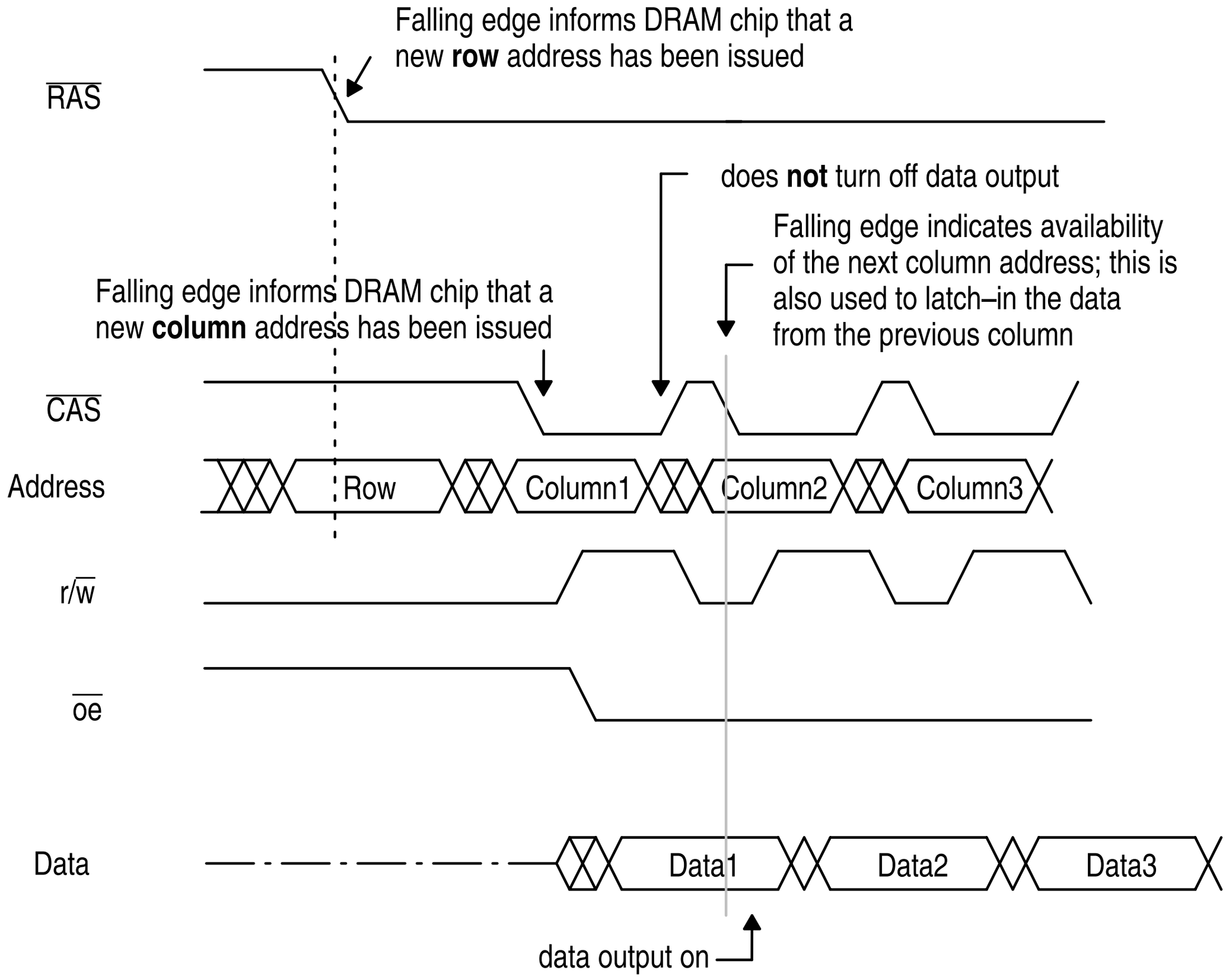
Refresh Options

- RAS-only
- CAS-before-RAS
- Hidden

DRAM Interface Evolution

- Asynchronous interfaces
- Fast page mode (FPM) -- allows multiple column access to same row
- Extended Data Out (EDO) -- primitive pipelining
- Synchronous DRAM -- fully pipelined


Falling edge informs DRAM chip that a new **row** address has been issued
RAS
Falling edge informs DRAM chip that a new **column** address has been issued
Rising edge **turns off** data output
CAS
Address
Row
Column1
Column2
Column3
r/w
oe
turnoff delay
Data
Data1
Data2
Data3
data output off


Falling edge informs DRAM chip that a new **row** address has been issued
RAS
does **not** turn off data output
Falling edge indicates availability of the next column address; this is also used to latch–in the data from the previous column
Falling edge informs DRAM chip that a new **column** address has been issued
CAS
Address
Row
Column1
Column2
Column3
r/w
oe
Data
Data1
Data2
Data3
data output on

Synchronous DRAM

- Interface is now synchronous, timed by an explicit clock signal
- Interface is pipelined, allowing for high clock rates and high throughput for column access
- Introduces "CAS latency"
- With DDR, data bus faster than control bus
- More recent advancements focus on signal integrity

Multiple Banks

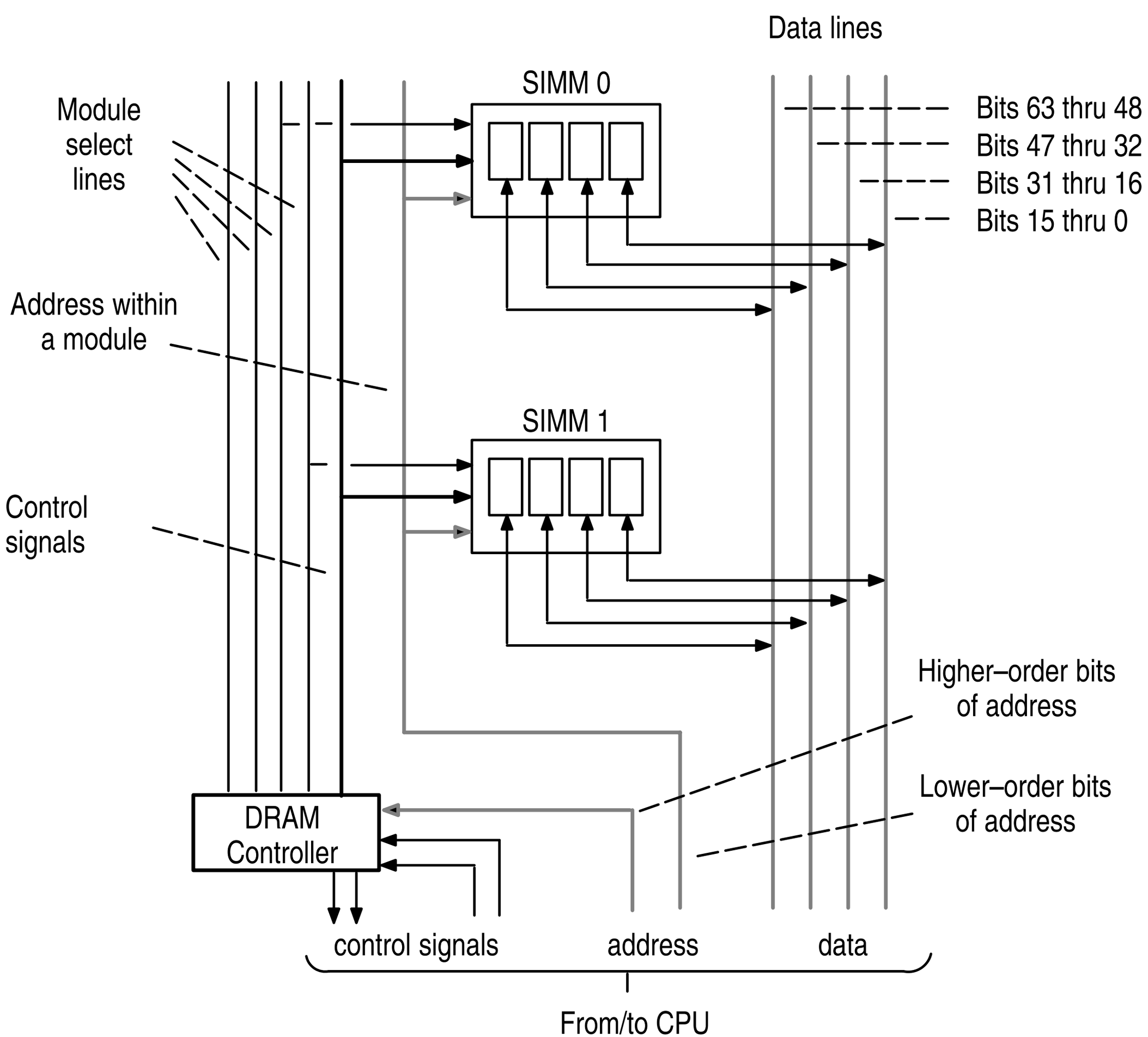
- DRAM internals divided into multiple independent banks
- Each has its own row buffer
- Share common interface
- Used to hide activate/precharge latencies

DRAM Commands

- RAS-CAS-WE
- 111 - No operation
- 110 - Burst terminate
- 101 - Read (CAS low, WE high)
- 100 - Write (CAS low, WE low)
- 011 - Activate (RAS low)
- 010 - Precharge
- 001 - Auto-refresh
- 000 - Load mode register

DRAM modules

- DRAM chips packages into modules
  - SIMMs and DIMMs
- 8 chips on one module constitute a “rank”
  - Common CS, all access in parallel
  - Controller has one CS for each rank

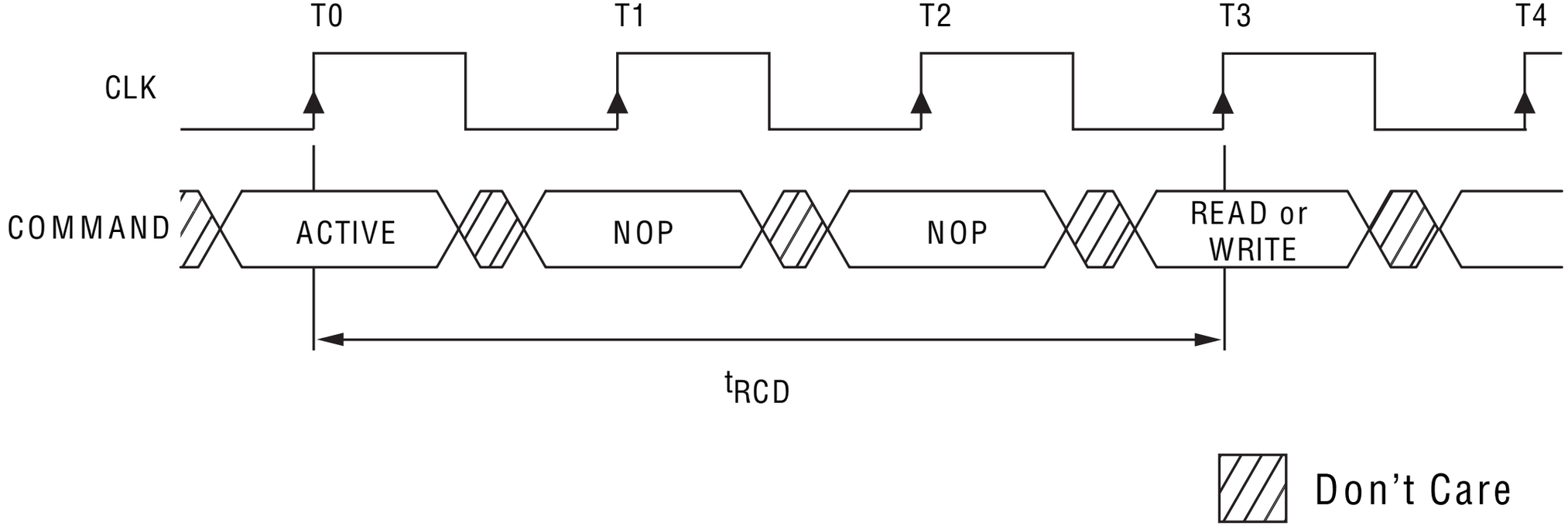

Data lines
SIMM 0
Module
select
lines
Bits 63 thru 48
Bits 47 thru 32
Bits 31 thru 16
Bits 15 thru 0
Address within
a module
SIMM 1
Control
signals
Higher–order bits
of address
Lower–order bits
of address
DRAM
Controller
control signals
address
data
From/to CPU

DRAM Timing Numbers

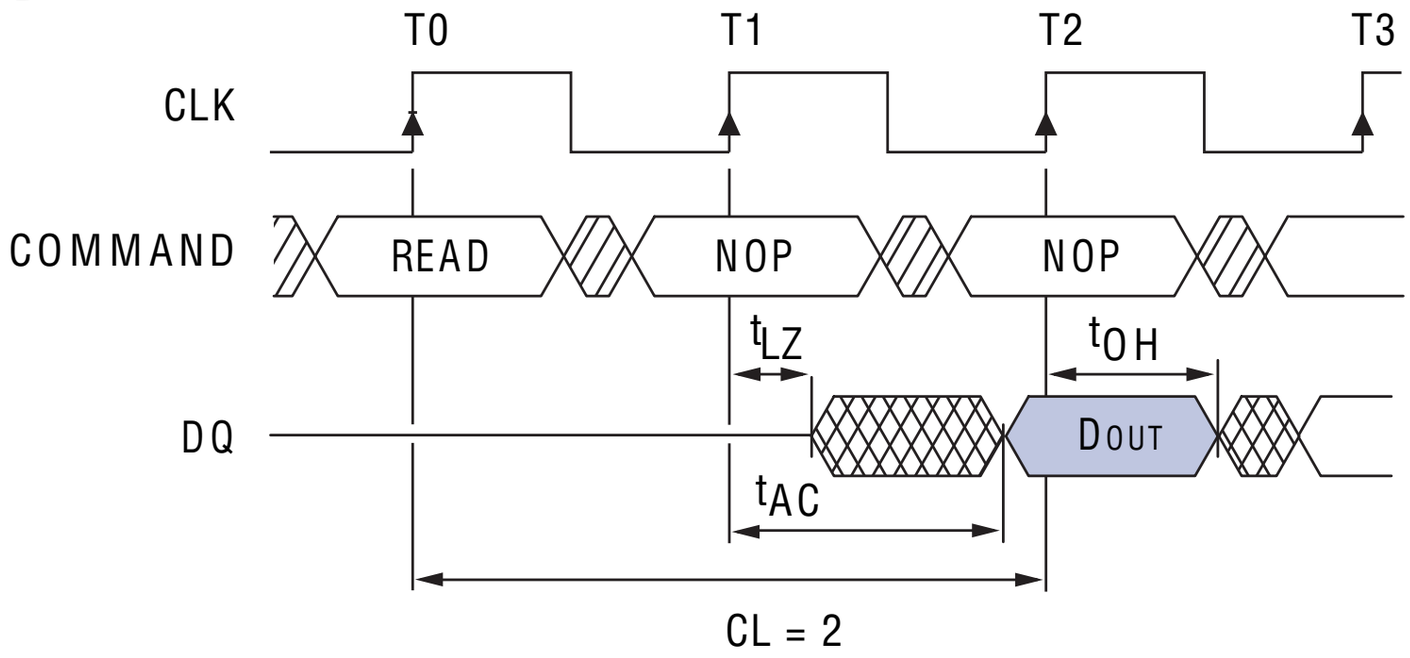
From… \ To…	Activate	Read	Write	Precharge
Activate	tRRD	tRCD	tRCD	tRAS
Read	--	1	tRTW	tRTP
Write	--	tWTR	1	tWR
Precharge	tRP	--	--	tPTP

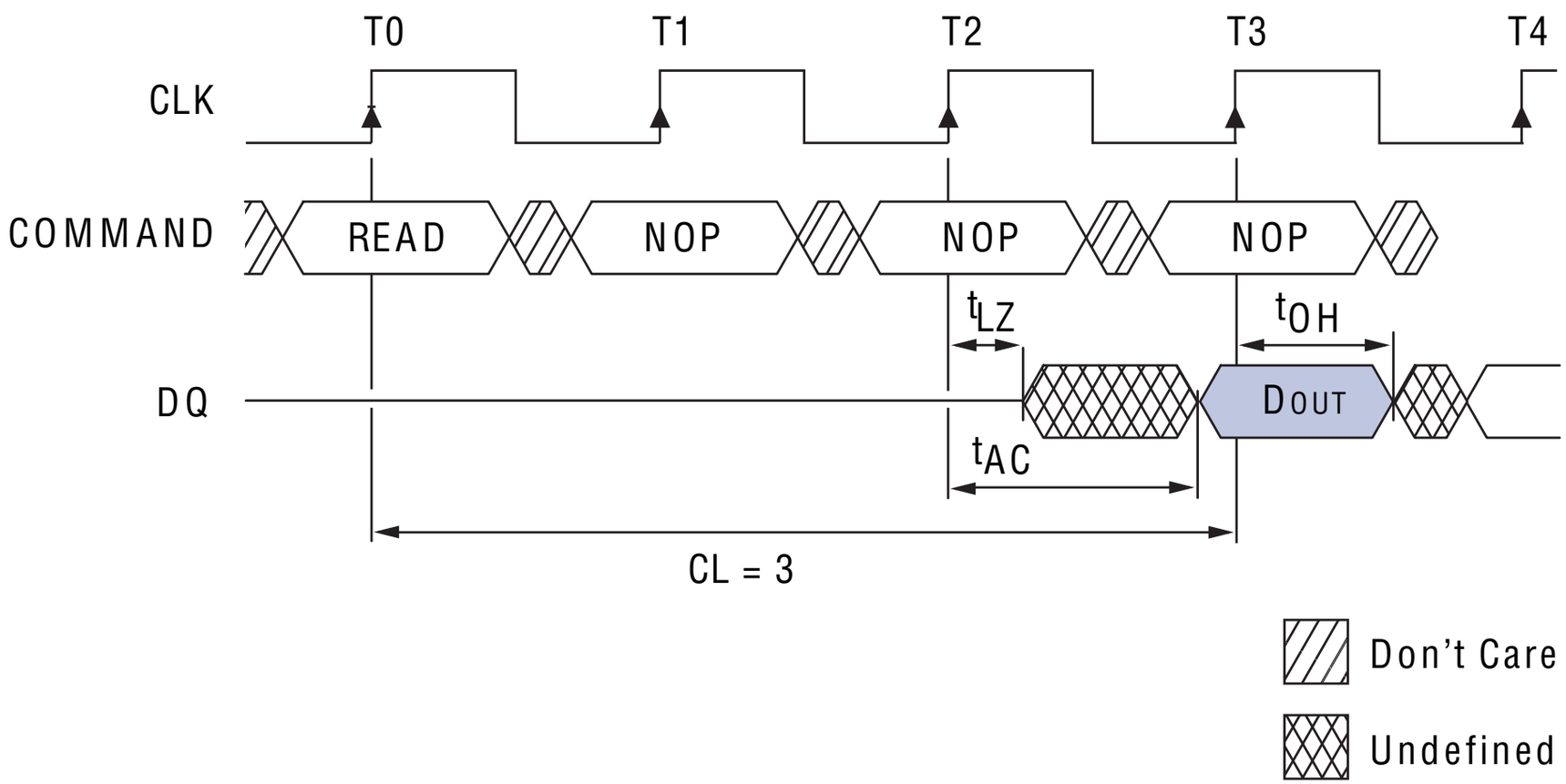
tCL = pipeline latency for reads

Example Meeting ^{t}RCD (MIN) when 2 < ^{t}RCD (MIN)/^{t}CK ≤ 3

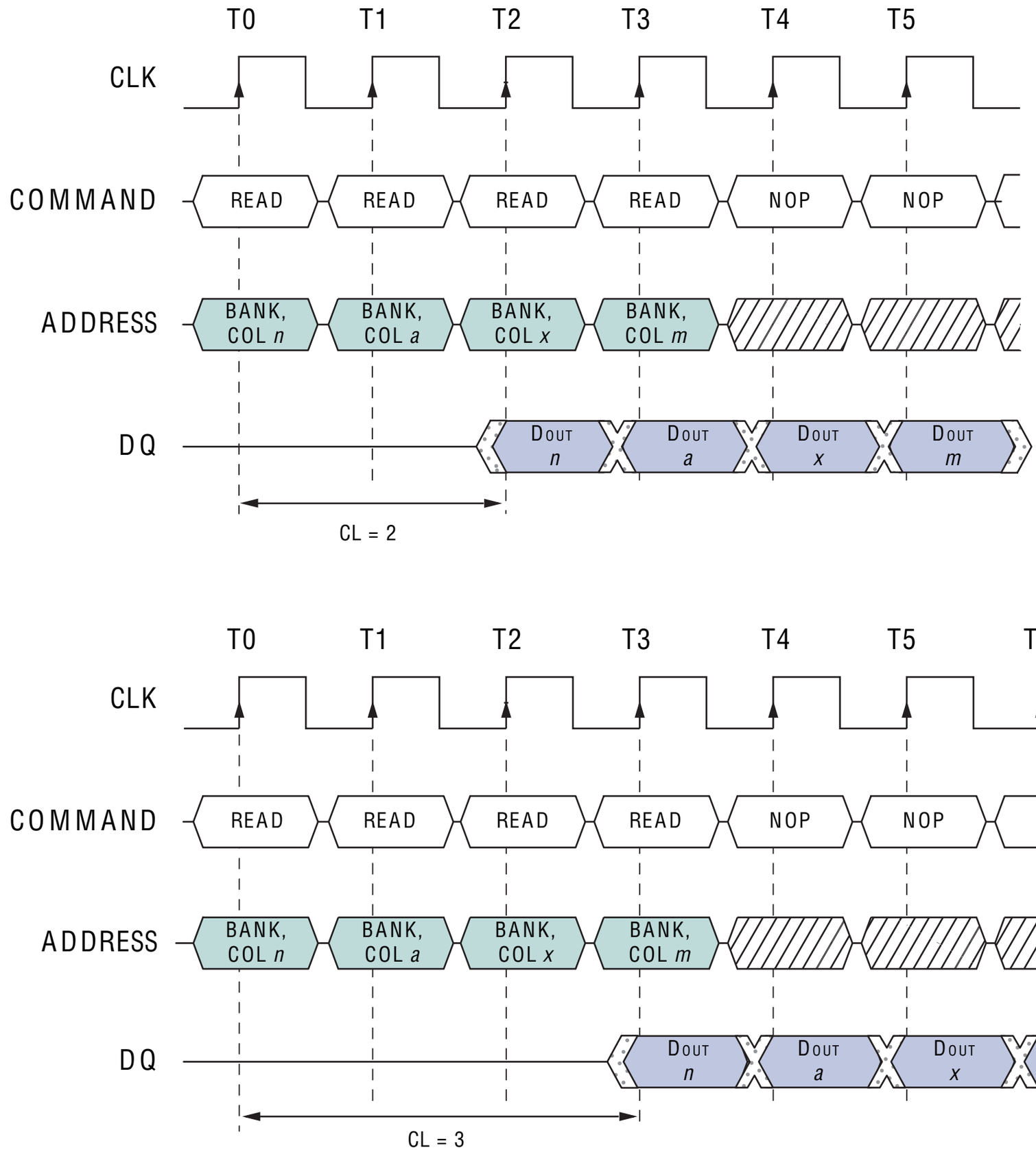


CAS Latency

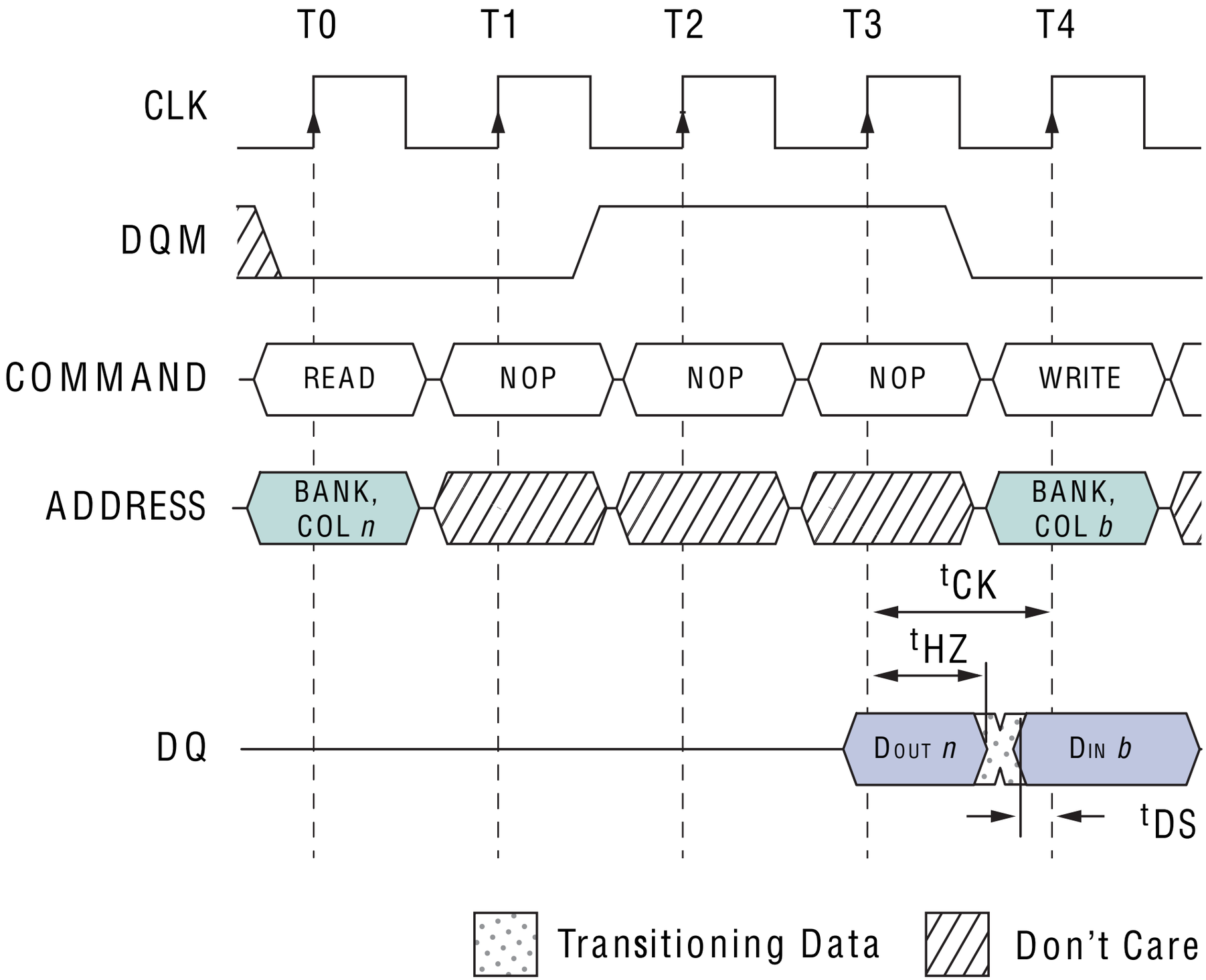

T0
T1
T2
T3
CLK
COMMAND
READ
NOP
NOP
tLZ
tOH
DQ
DOUT
tAC
CL = 2


T0
T1
T2
T3
T4
CLK
COMMAND
READ
NOP
NOP
NOP
tLZ
tOH
DQ
DOUT
tAC
CL = 3
Don't Care
Undefined

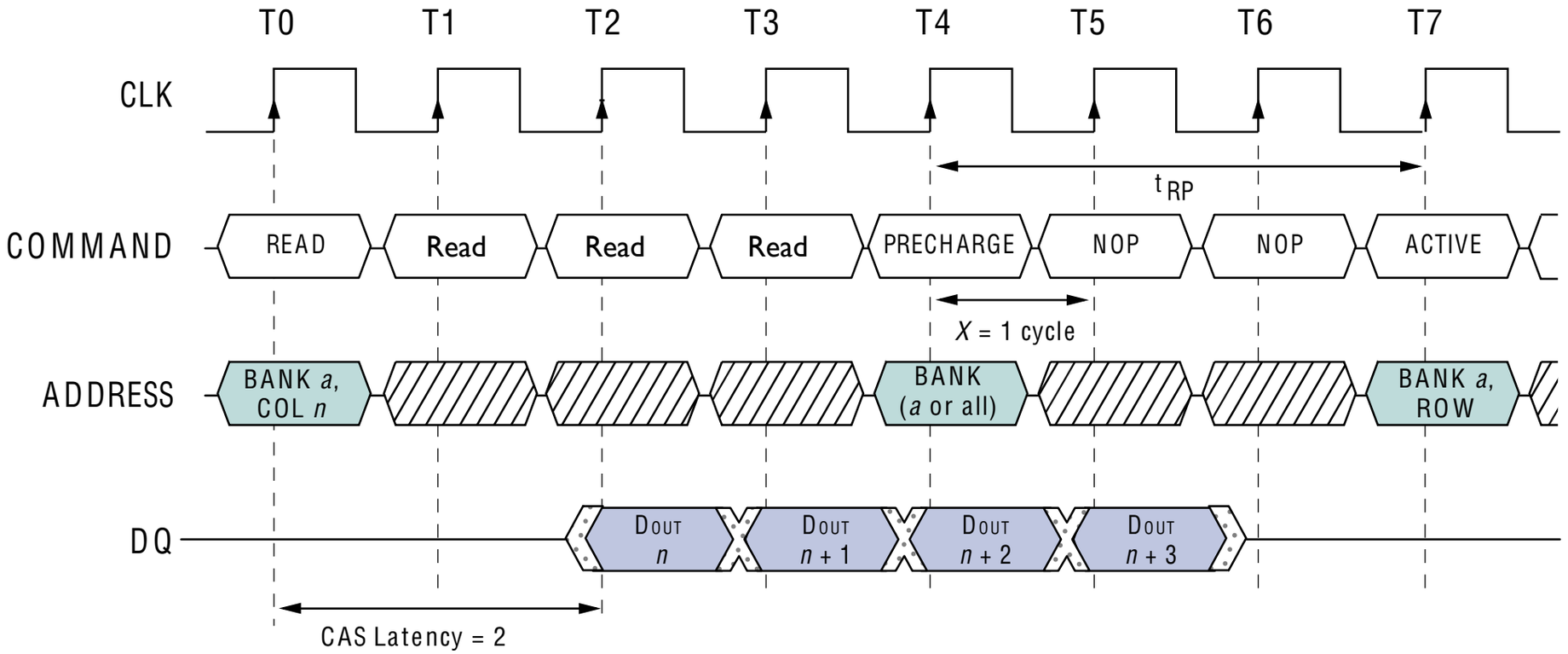
Random READ Accesses

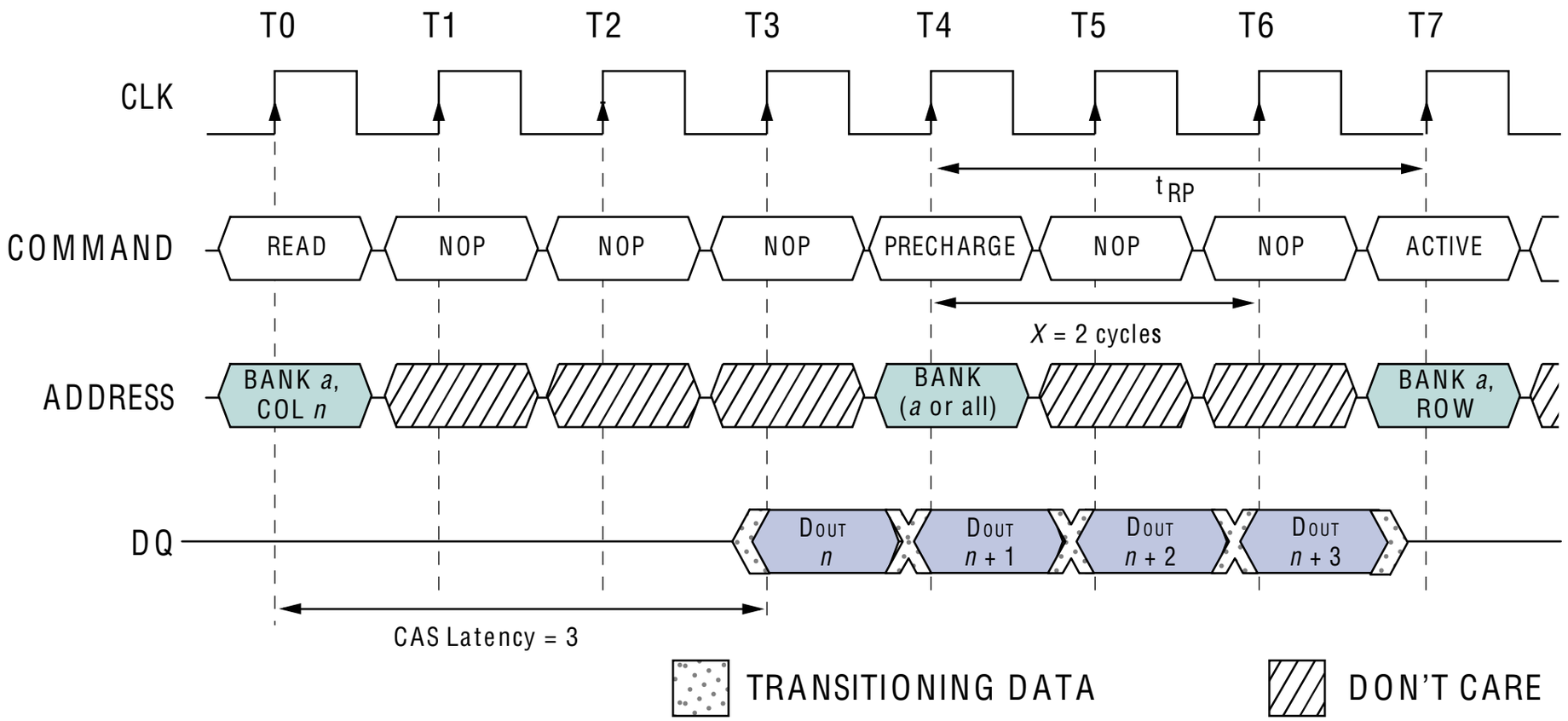


READ-to-WRITE

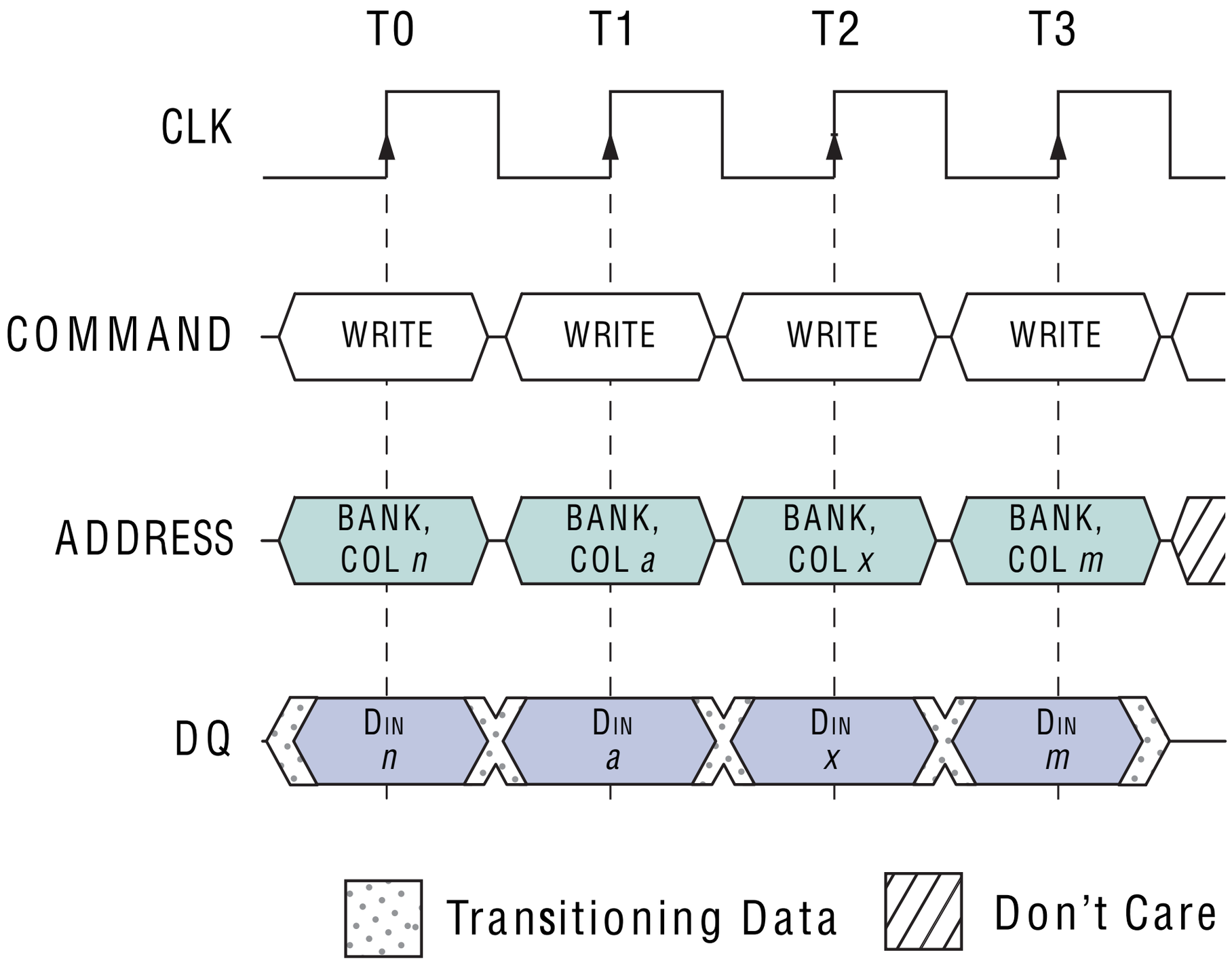


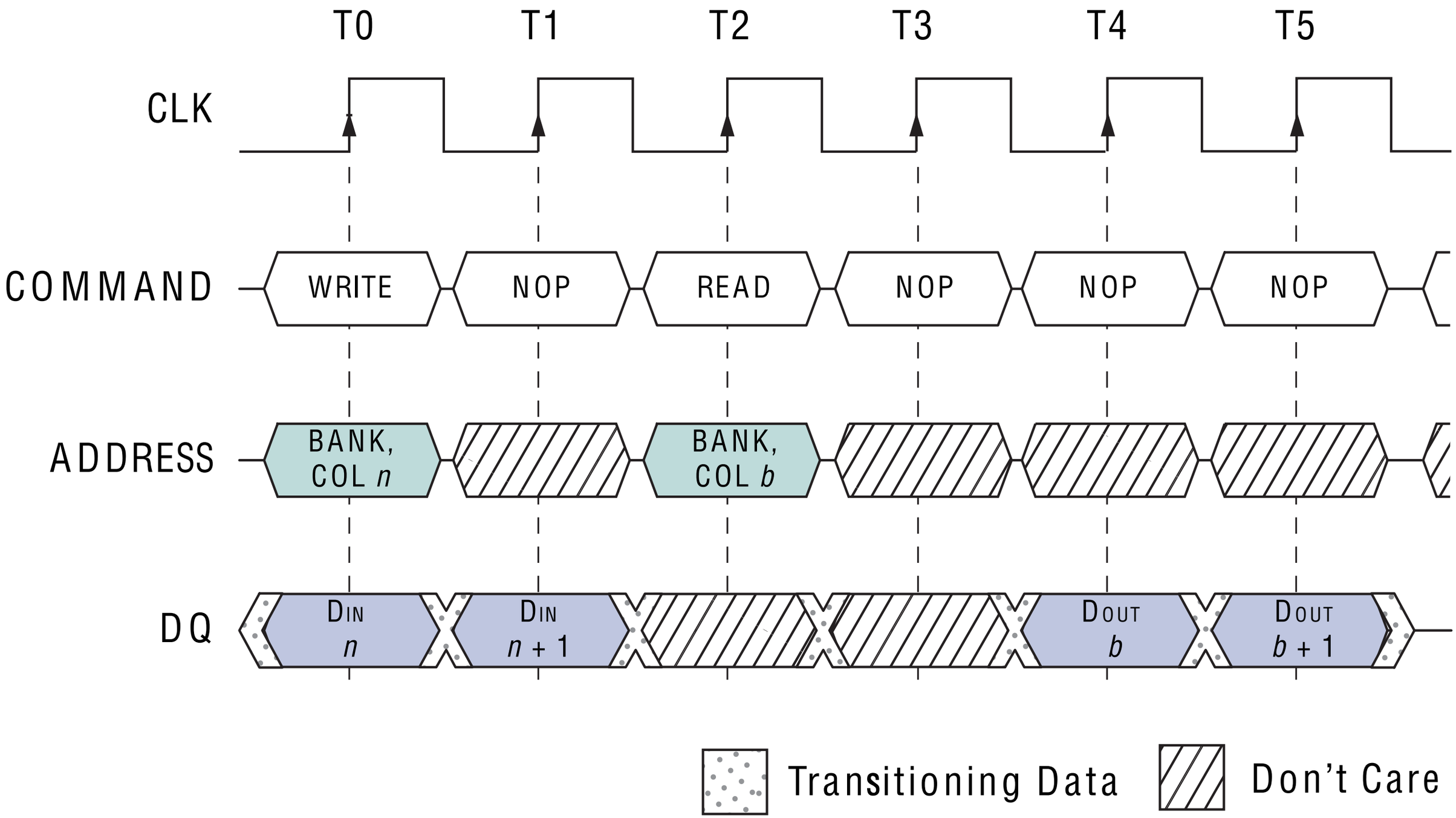
READ-to-PRECHARGE

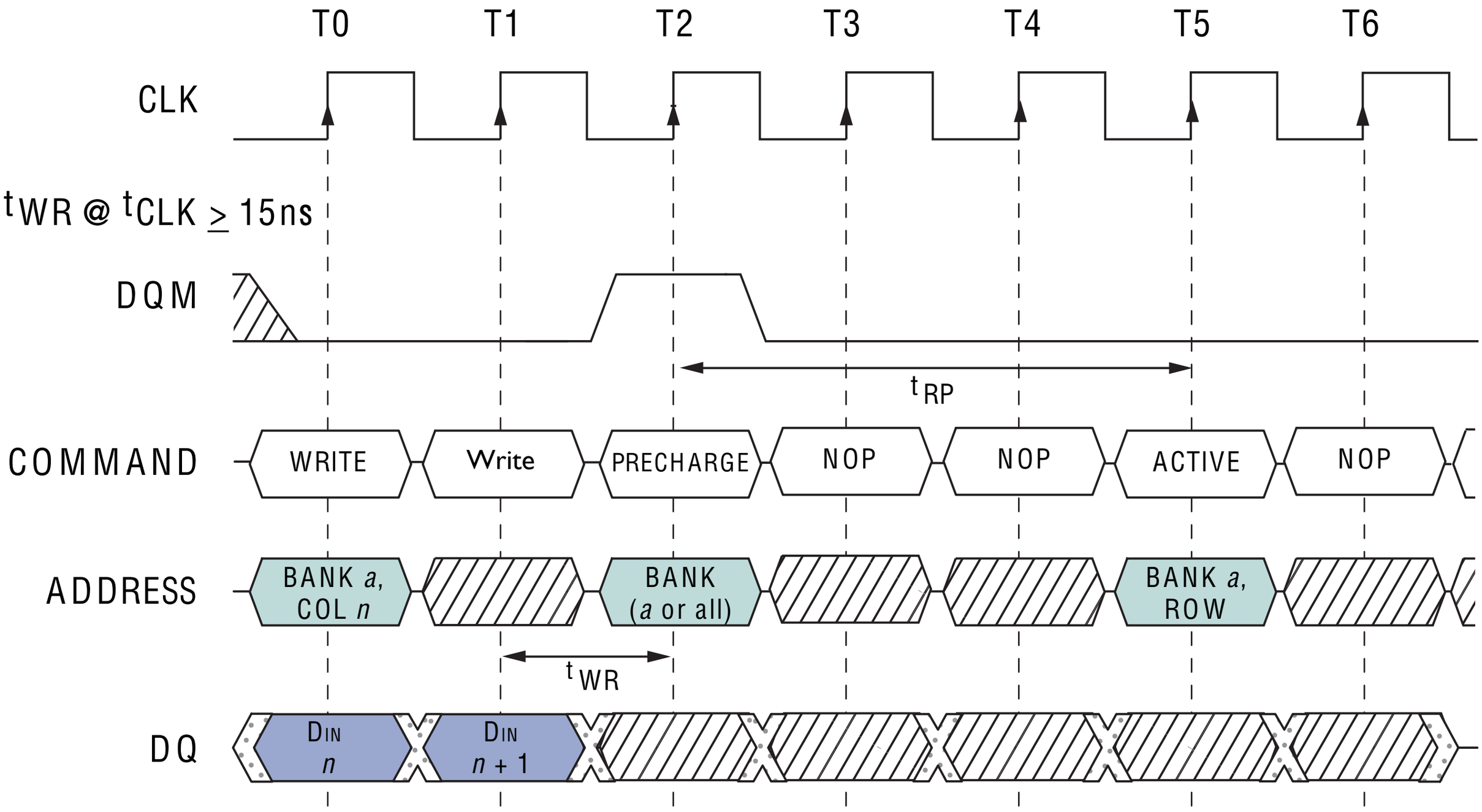




Random WRITE Cycles




T0
T1
T2
T3
T4
T5
CLK
COMMAND
WRITE
NOP
READ
NOP
NOP
NOP
ADDRESS
BANK, COL n
BANK, COL b
DQ
DIN n
DIN n + 1
DOUT b
DOUT b + 1
Transitioning Data
Don't Care


T0
T1
T2
T3
T4
T5
T6
CLK
tWR @ tCLK ≥ 15ns
DQM
tRP
COMMAND
WRITE
Write
PRECHARGE
NOP
NOP
ACTIVE
NOP
ADDRESS
BANK a, COL n
BANK (a or all)
BANK a, ROW
tWR
DQ
DIN n
DIN n + 1