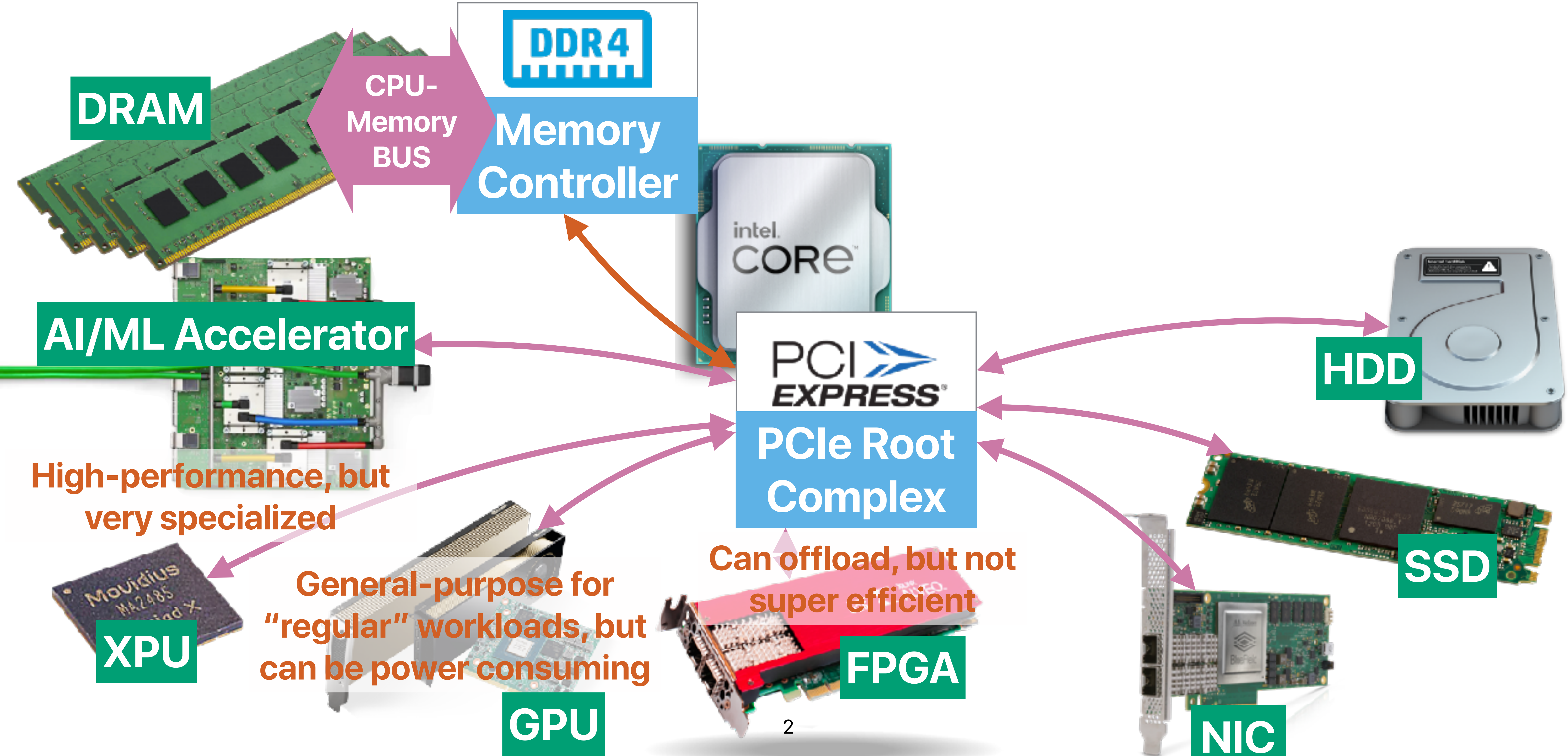
Memory subsystem

Hung-Wei Tseng

Recap: The landscape of modern computers



Recap: Approx. on NPU v.s. GPTPU

	Approx. on NPU	GPTPU
Performance	Better if the algorithm is more complex, but worse otherwise	Depending on the complexity of the algorithm
Result quality	Approximate, not accurate	Exact — if the hardware supports the desired precision
Programming efforts	Easier — the programmer does not need to take care of programming on AI/ML accelerators	Difficult — the programmer has to map the computation into operators that the hardware supports
Applicability	Any algorithm on an application tolerating inexactness	Only algorithms that map well to the hardware architecture

Recap: TCUDB

```
SELECT A.customer, B.brand, SUM(A.Quantity * B.Price) AS value FROM A INNER JOIN B WHERE ON
A.ProductID = B.ProductID GROUP BY A.customer, B.brand;
```

A

Customer	ProductID	Brand	Quant.
Abe	i9-12900	Intel	1
Abe	i7-12700	Intel	1
Abe	RTX3080	NVIDIA	1
Abe	660p	Intel	2
Bob	RyZen 5800G	AMD	1
Bob	980Pro	Samsung	1
Cindy	RTX3080	NVIDIA	1
Cindy	i7-12700	Intel	2
Cindy	660p	Intel	2
Diana	RyZen 5800G	AMD	1
Diana	RX5000	AMD	1
Diana	980Pro	Samsung	1

B $O(n)$

Brand	ProductID	Price
Intel	i9-12900	600
Intel	i7-12700	400
Intel	660p	100
AMD	RX5000	500
AMD	RyZen 5800G	200
NVIDIA	RTX3080	1200
Samsung	980Pro	100

Customer	Intel	AMD	NVIDIA	Samsung
Abe	1200	0	1200	0
Bob	0	400	0	100
Cindy	1000	0	1200	0
Diana	0	900	0	100

=

Matrix multiplications

$O(m \times n \times k)$

— This could be done by 1 Tensor Core OP!

Left

Customer/ ProductID	i9-12900	i7-12700	660p	RX5000	RyZen 5800G	RTX3080	980Pro
Abe	1	1	2	0	0	1	0
Bob	0	0	0	0	1	0	1
Cindy	0	2	2	0	0	1	0
Diana	0	0	0	1	1	0	1

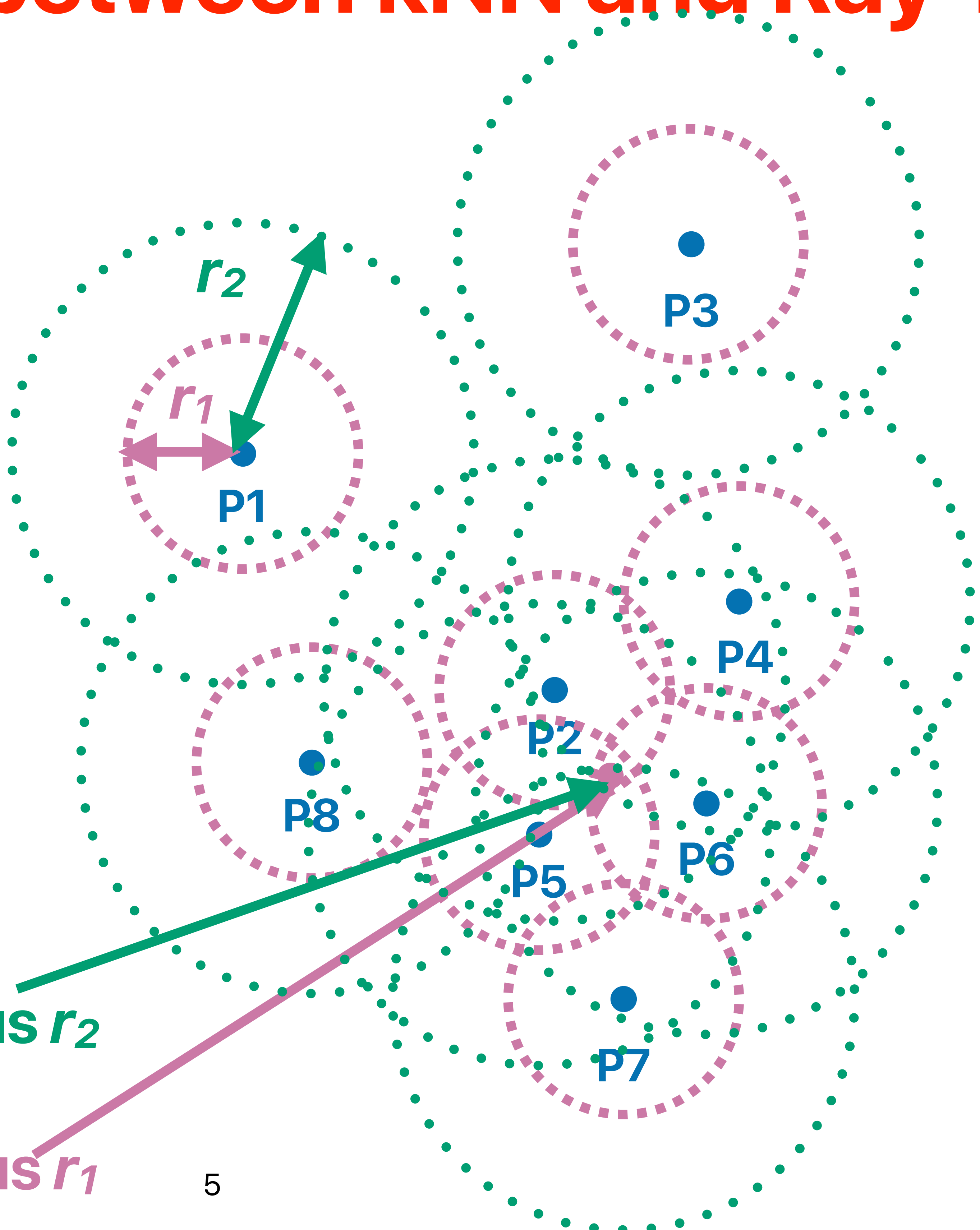
×

ProductID/ Brand	Intel	AMD	NVIDIA	Samsung
i9-12900	600	0	0	0
i7-12700	400	0	0	0
660p	100	0	0	0
RX5000	0	500	0	0
RyZen 5800G	0	400	0	0
RTX3080	0	0	1200	0
980Pro	0	0	0	100

Memory copy

$O(m)$

Recap: Similarities between kNN and Ray Tracing



Assume we want to find “5” NN

Answer: P2, P4, P5, P6, P7

Q_1 with radius r_2

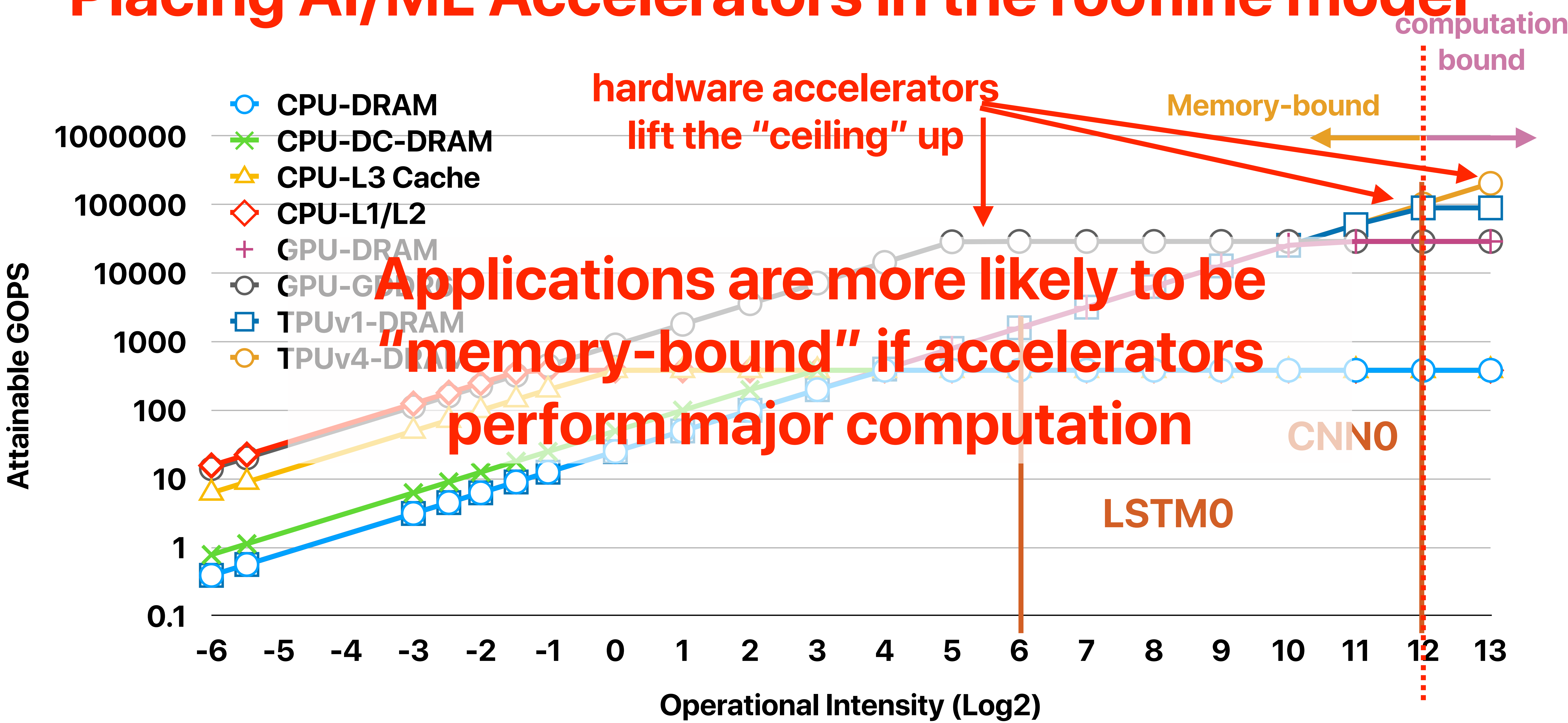
Answer: P2, P5, P6

Q_1 with radius r_1

Recap: Summary of General-purpose computing on "something"

- Map your problem as a problem that the target hardware can solve
  - Machine learning models (NPU)
  - Machine learning mathematical operations (GP Edge TPU)
  - Matrix multiplications (TCUDB)
  - Ray Tracing (RTNN)
- Due to the domain specific nature of accelerators, we have to program in a more domain specific way, "currently".
- Not necessarily the most performant, but typically more energy-efficient

Placing AI/ML Accelerators in the roofline model



Applications are more likely to be "memory-bound" if accelerators perform major computation

Outline

- Memory

3:00

What are potential approaches to lift the roofline on memory-bound applications?

Ideas of improving memory-bound programs?

Shifting the roofline

- Faster memory technologies
- Higher memory bandwidth
- Lower data volume

Memory technologies

3:00

What kinds of memory technologies are presented in modern computer systems? Strength? Weakness?

Memory technologies we have today

Volatile v.s. Non-volatile

- Volatile memory
  - The stored bits will vanish if the cell is not supplied with electricity
  - Register, SRAM, DRAM
- Non-volatile memory
  - The stored bits will not vanish “immediately” when it’s out of electricity — usually can last years
  - Flash memory, PCM, MRAM, STTRAM

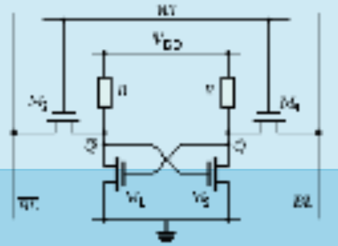
Memory technologies we have today

- SRAM
- DRAM
- Registers
- PCM
- 3DXPoint
- RRAM
- Flash memory

Memory technologies we have today

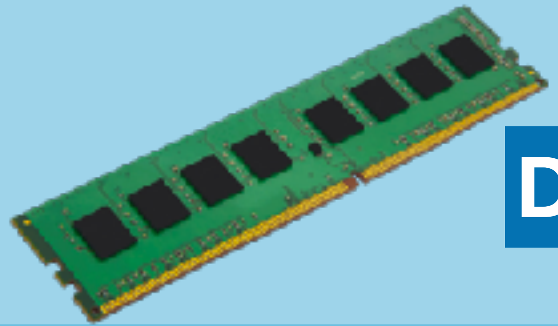
Volatile Memory | **Non-Volatile Memory**

100ps

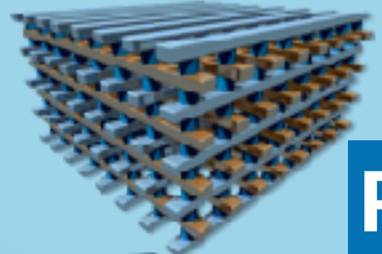


SRAM

ns



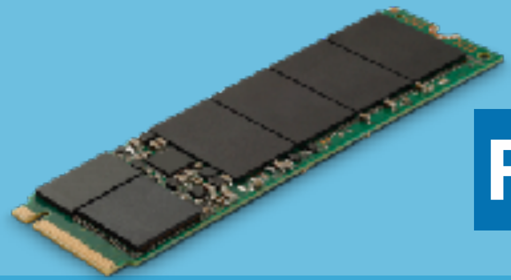
DRAM



RRAM

PCM 3DXPoint

us

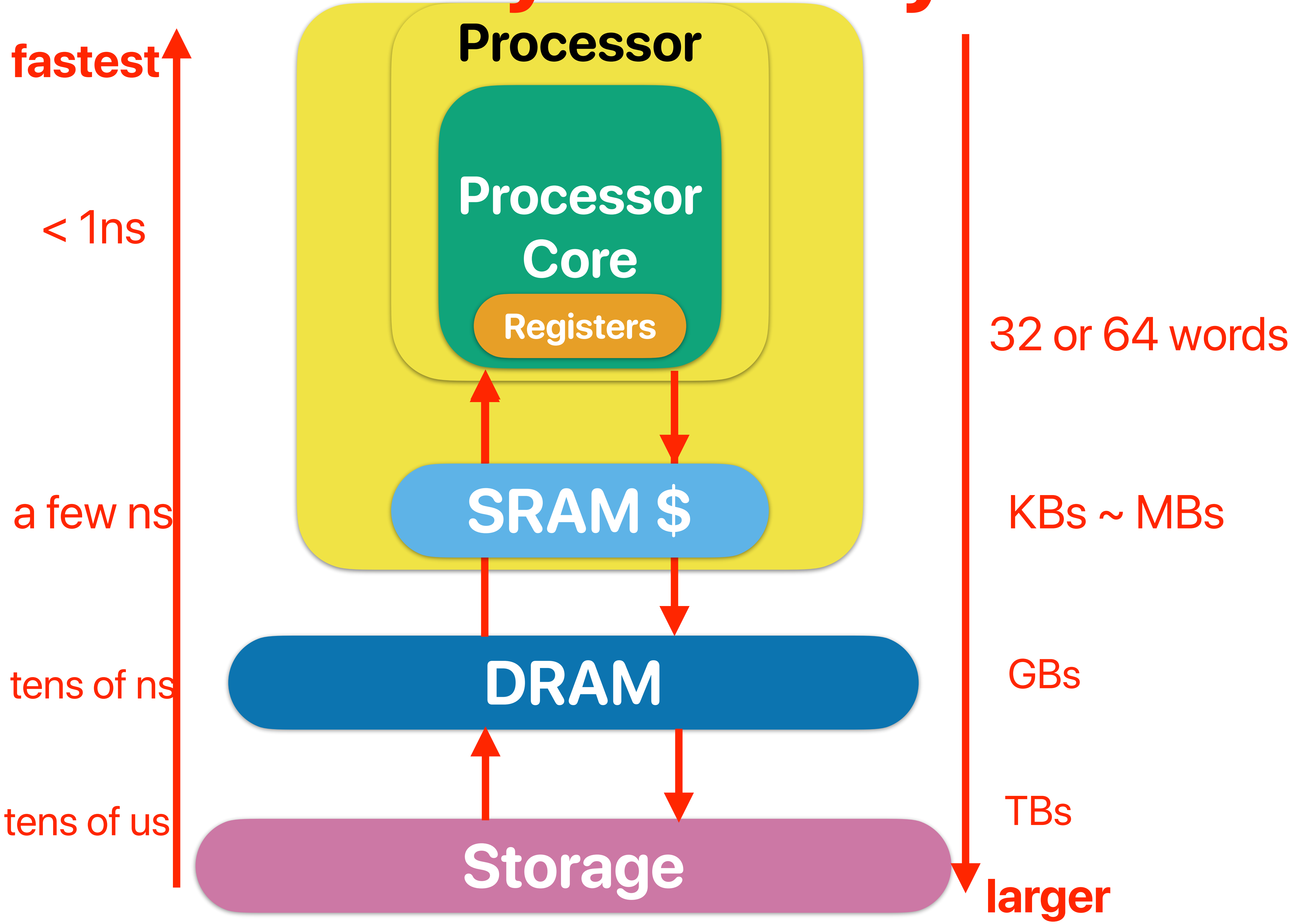


Flash memory

ms

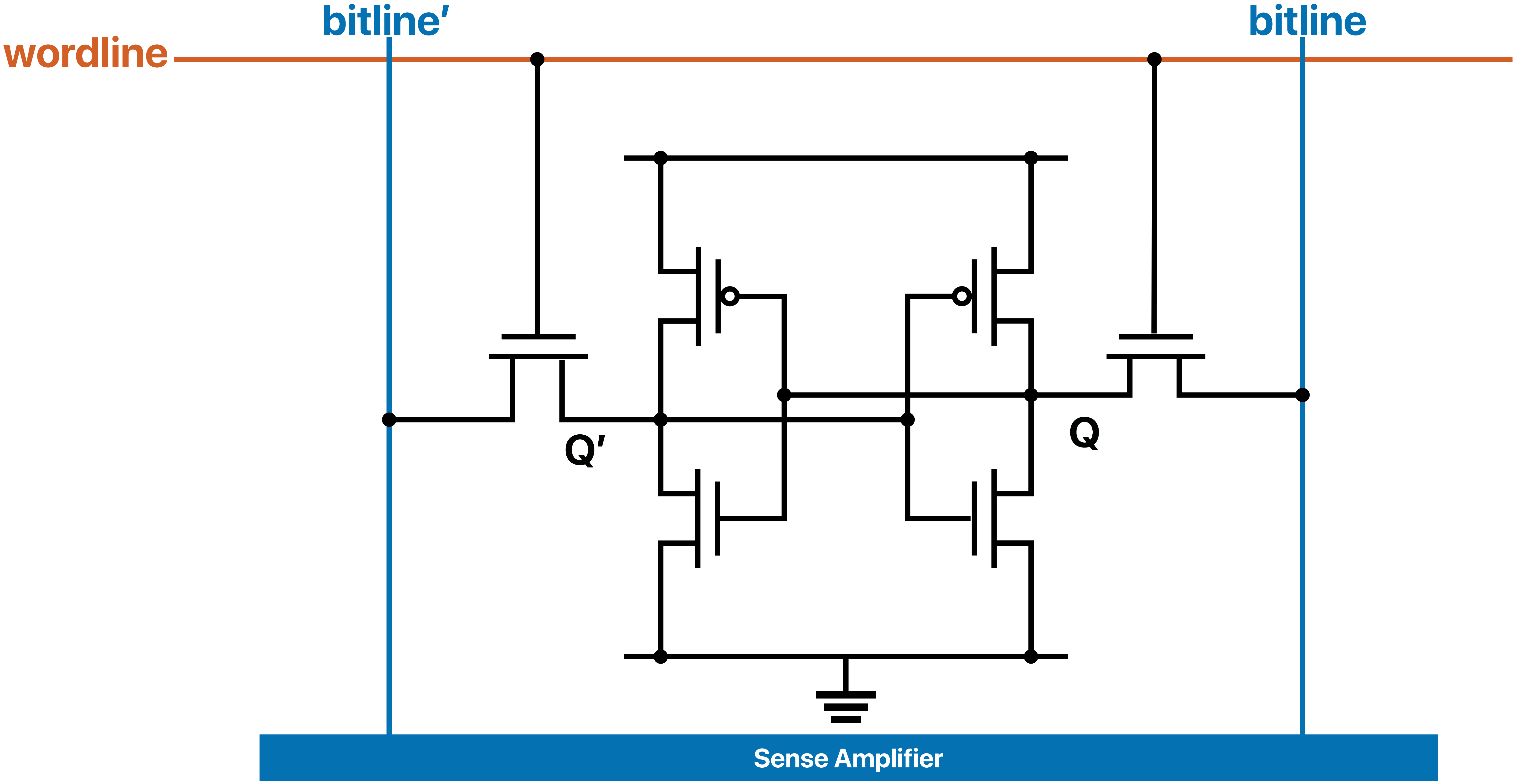
Hard Disk Drives

Memory Hierarchy

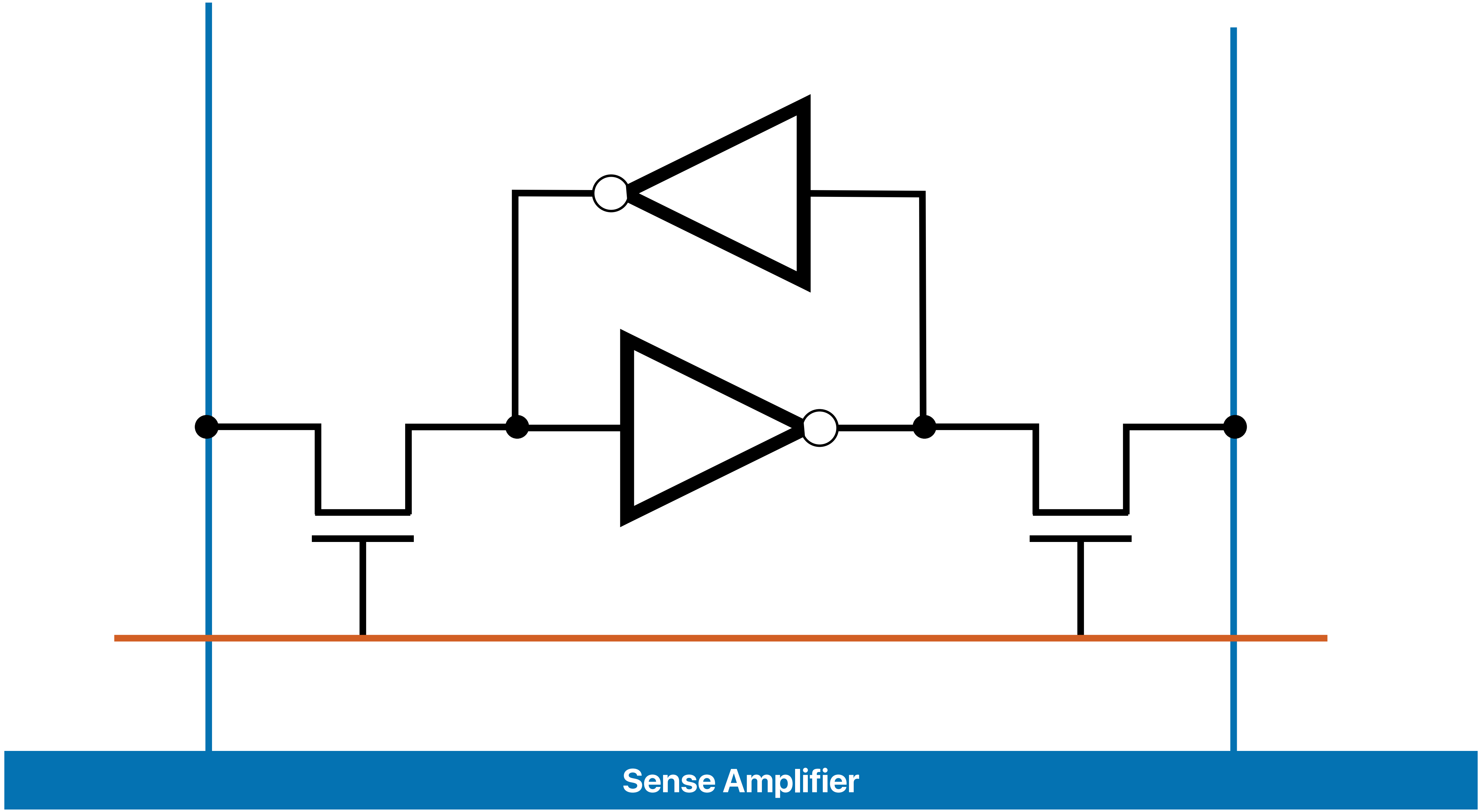


Static Random Access Memory (SRAM)

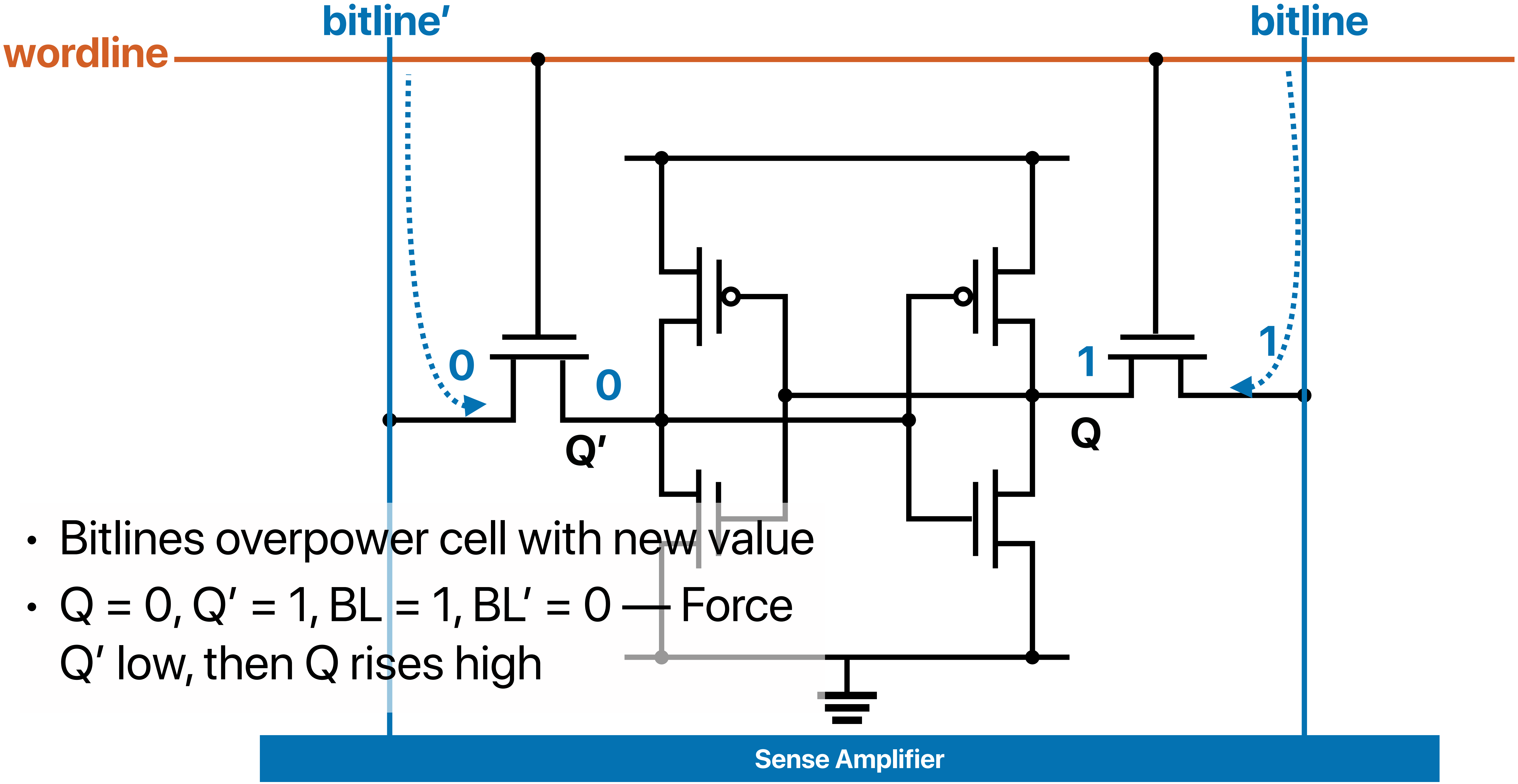
A Classical 6-T SRAM Cell



A Classical 6-T SRAM Cell

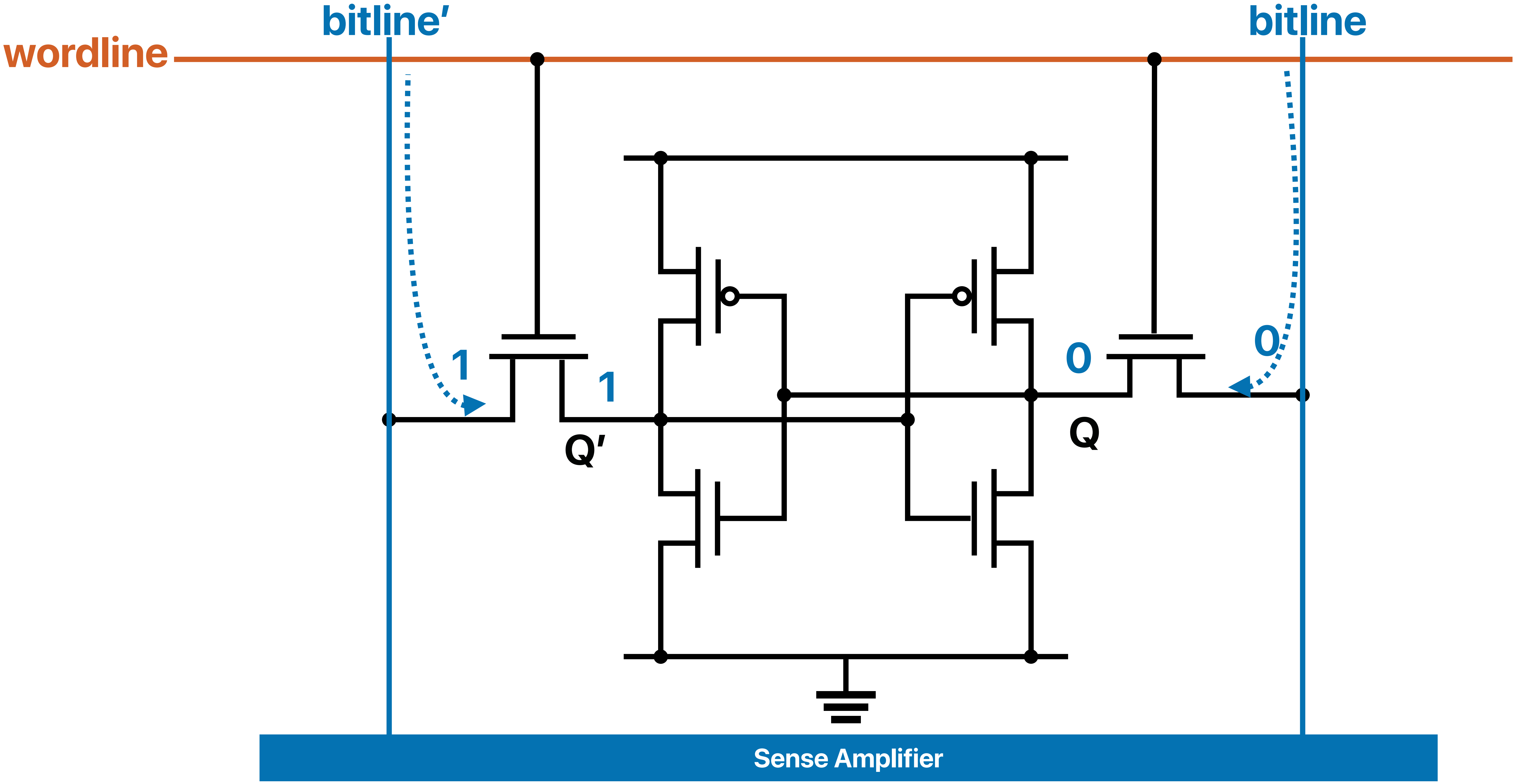


Write "1" to an SRAM Cell

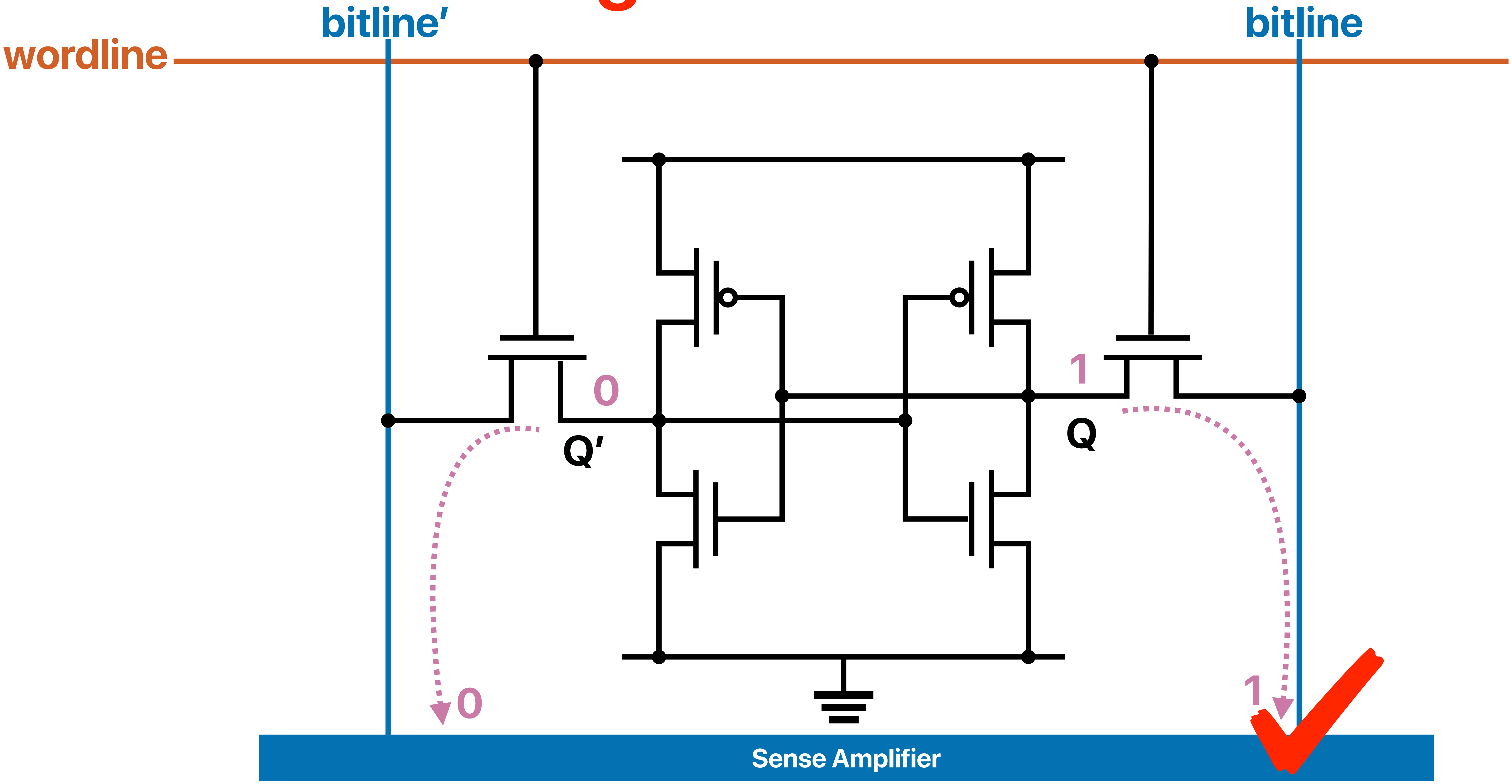


- Bitlines overpower cell with new value
- Q = 0, Q′ = 1, BL = 1, BL′ = 0 — Force Q′ low, then Q rises high

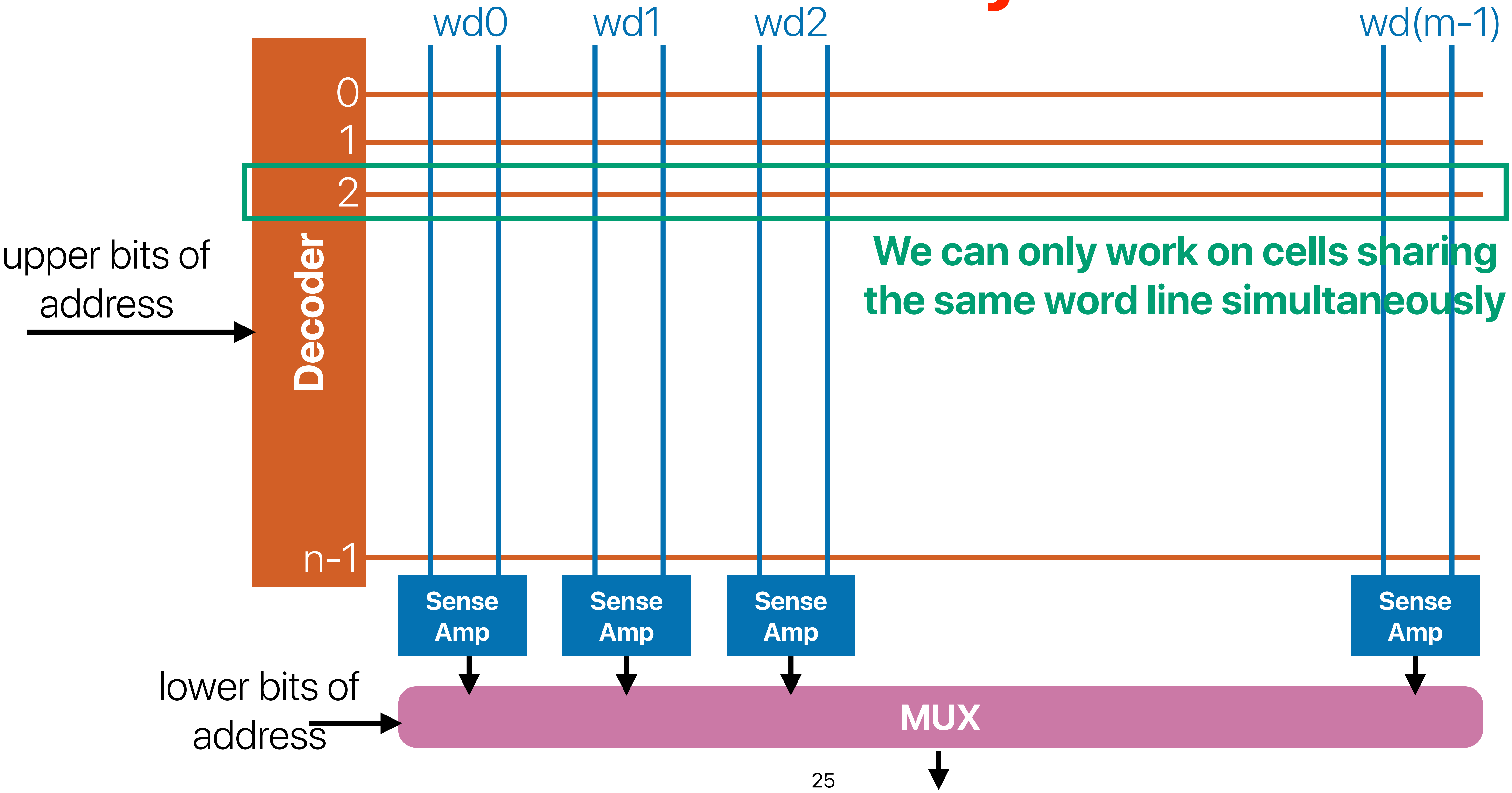
Write “0” to an SRAM Cell



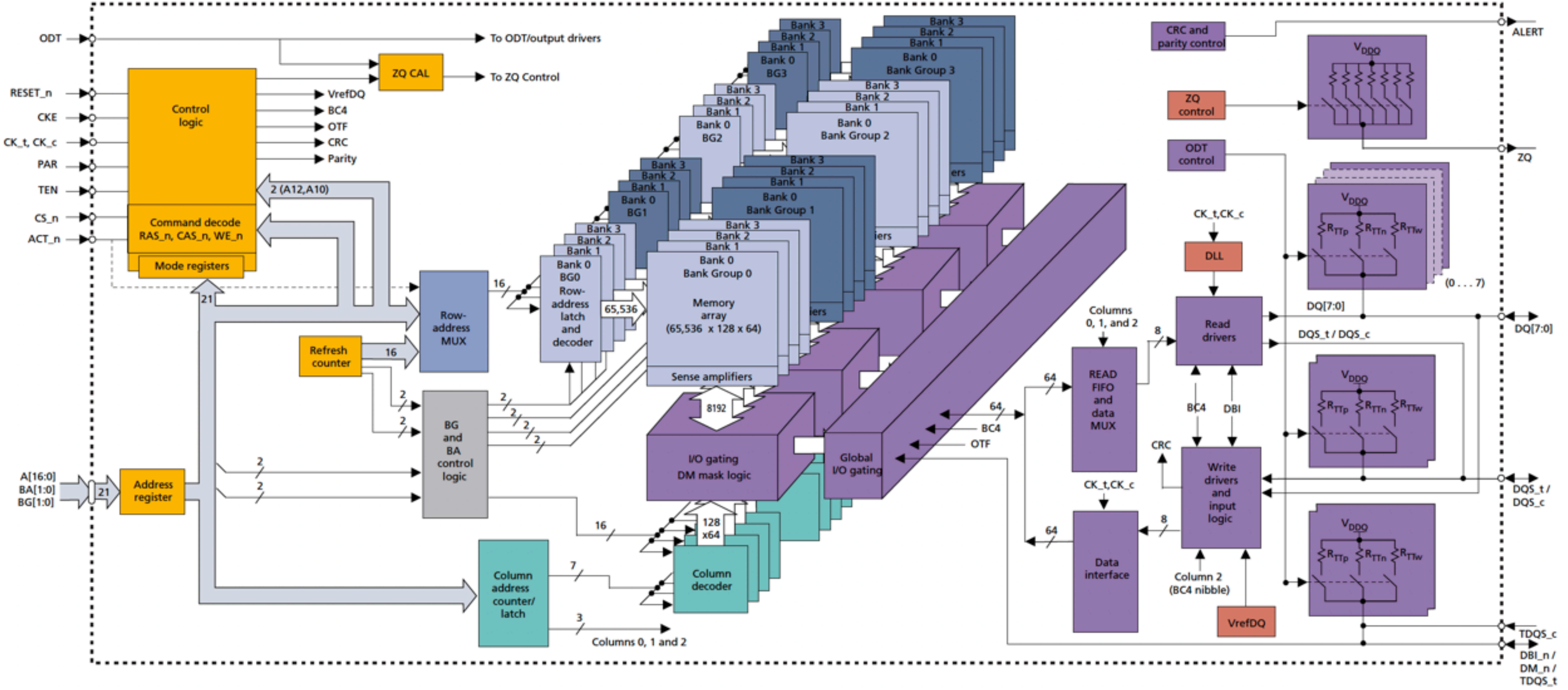
Reading from an SRAM Cell



SRAM array

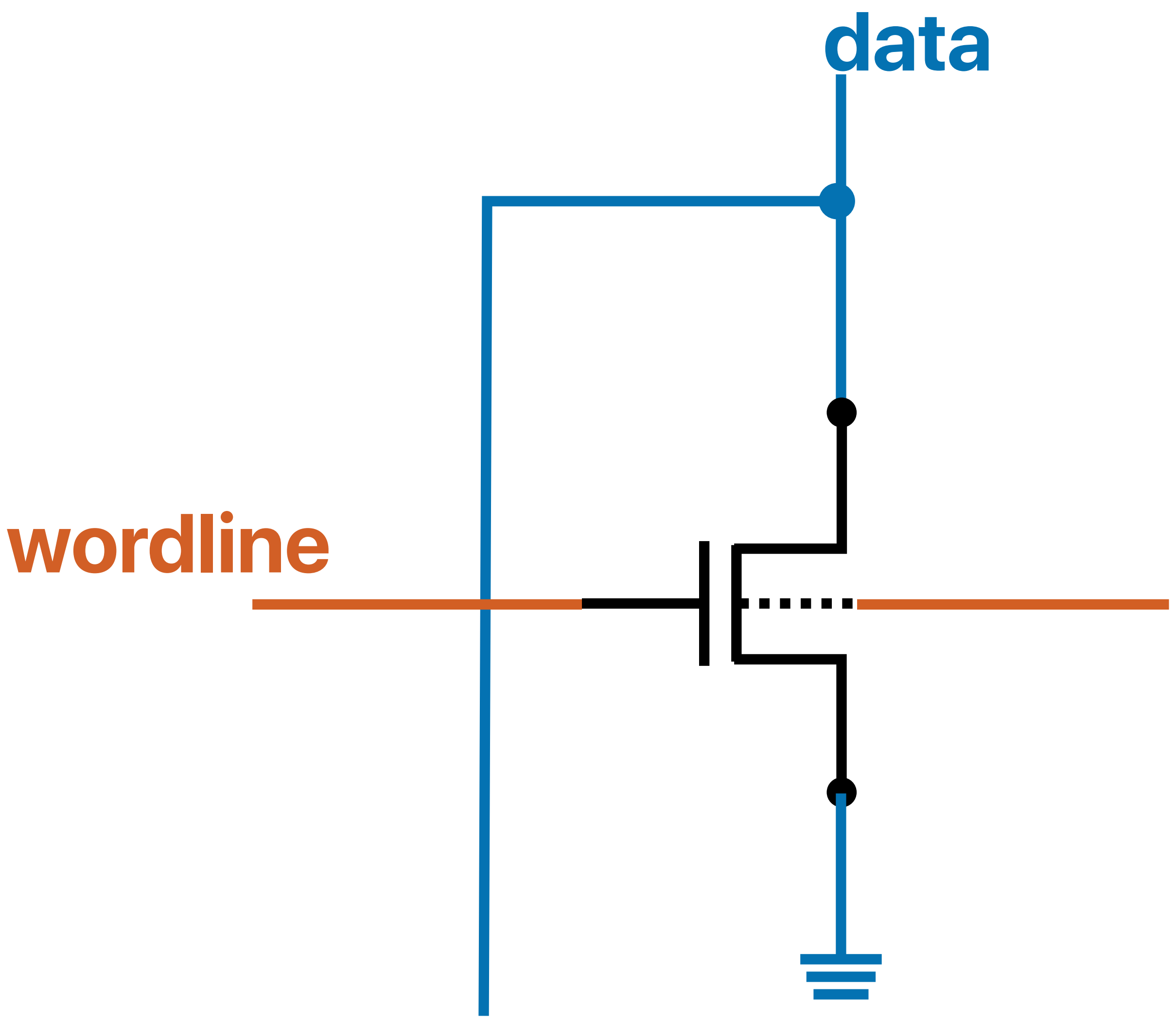


wd0
wd1
wd2
wd(m-1)
0
1
2
n-1
Decoder
upper bits of address
We can only work on cells sharing the same word line simultaneously
Sense Amp
Sense Amp
Sense Amp
Sense Amp
lower bits of address
MUX

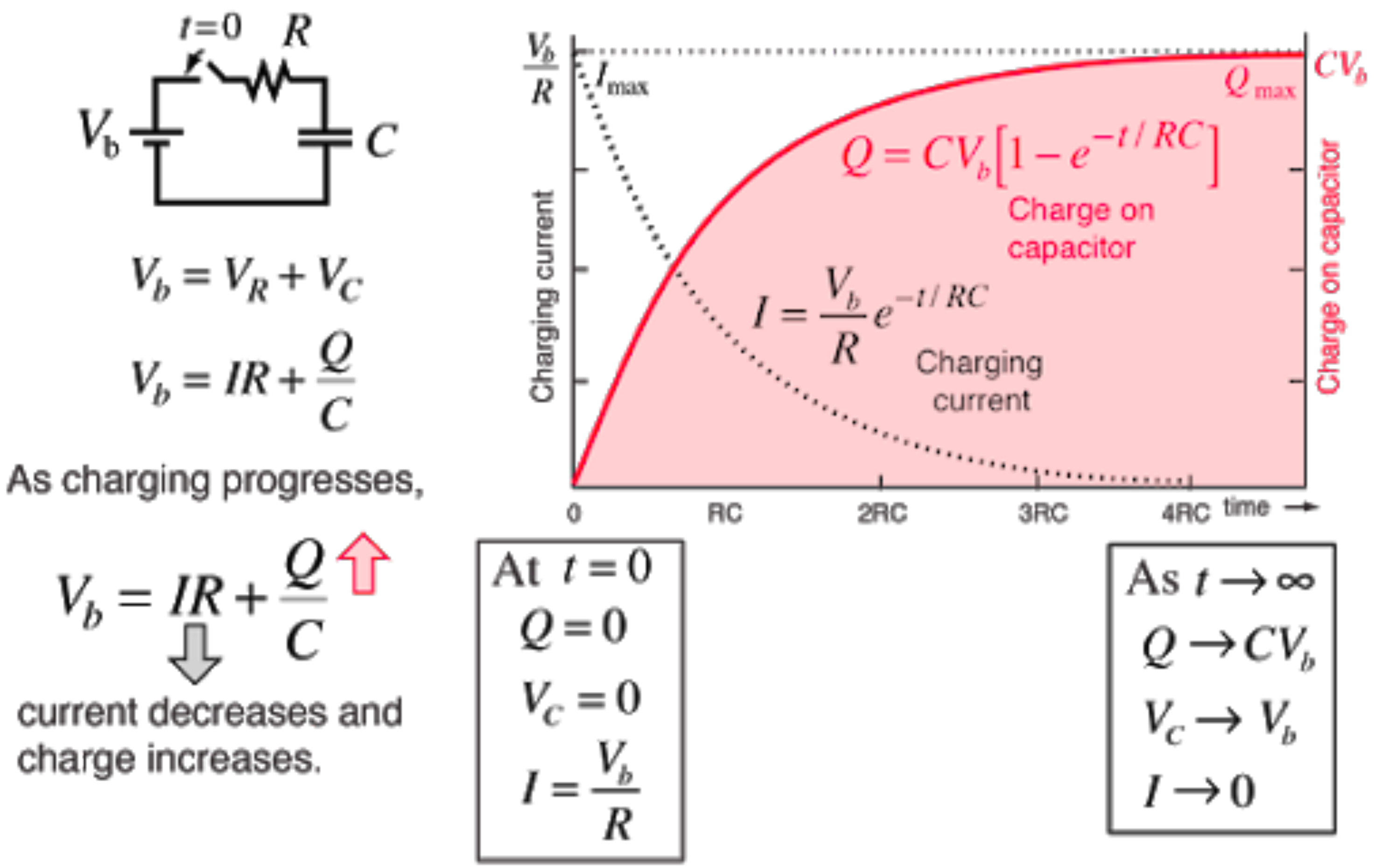


ODT
RESET_n
CKE
CK_t, CK_c
PAR
TEN
CS_n
ACT_n
To ODT/output drivers
ZQ CAL
To ZQ Control
Control logic
VrefDQ
BC4
OTF
CRC
Parity
2 (A12,A10)
Command decode RAS_n, CAS_n, WE_n
Mode registers
21
Row-address MUX
16
Refresh counter
Bank 0 BG0 Row-address latch and decoder
65,536
Bank 0 BG1
Bank 0 BG2
Bank 0 BG3
Bank 0 Bank Group 0
Bank 0 Bank Group 1
Bank 0 Bank Group 2
Bank 0 Bank Group 3
Bank 1
Bank 2
Bank 3
Memory array (65,536 x 128 x 64)
Sense amplifiers
8192
I/O gating DM mask logic
Global I/O gating
BG and BA control logic
A[16:0]
BA[1:0]
BG[1:0]
Address register
Column address counter/ latch
Column decoder
128 x64
Columns 0, 1 and 2
Columns 0, 1, and 2
READ FIFO and data MUX
Data interface
CK_t,CK_c
Read drivers
Write drivers and input logic
Column 2 (BC4 nibble)
VrefDQ
DLL
CRC and parity control
ZQ control
ODT control
ALERT
ZQ
DQ[7:0]
DQS_t / DQS_c
(0 . . . 7)
TDQS_c
DBI_n / DM_n / TDQS_t
DBI

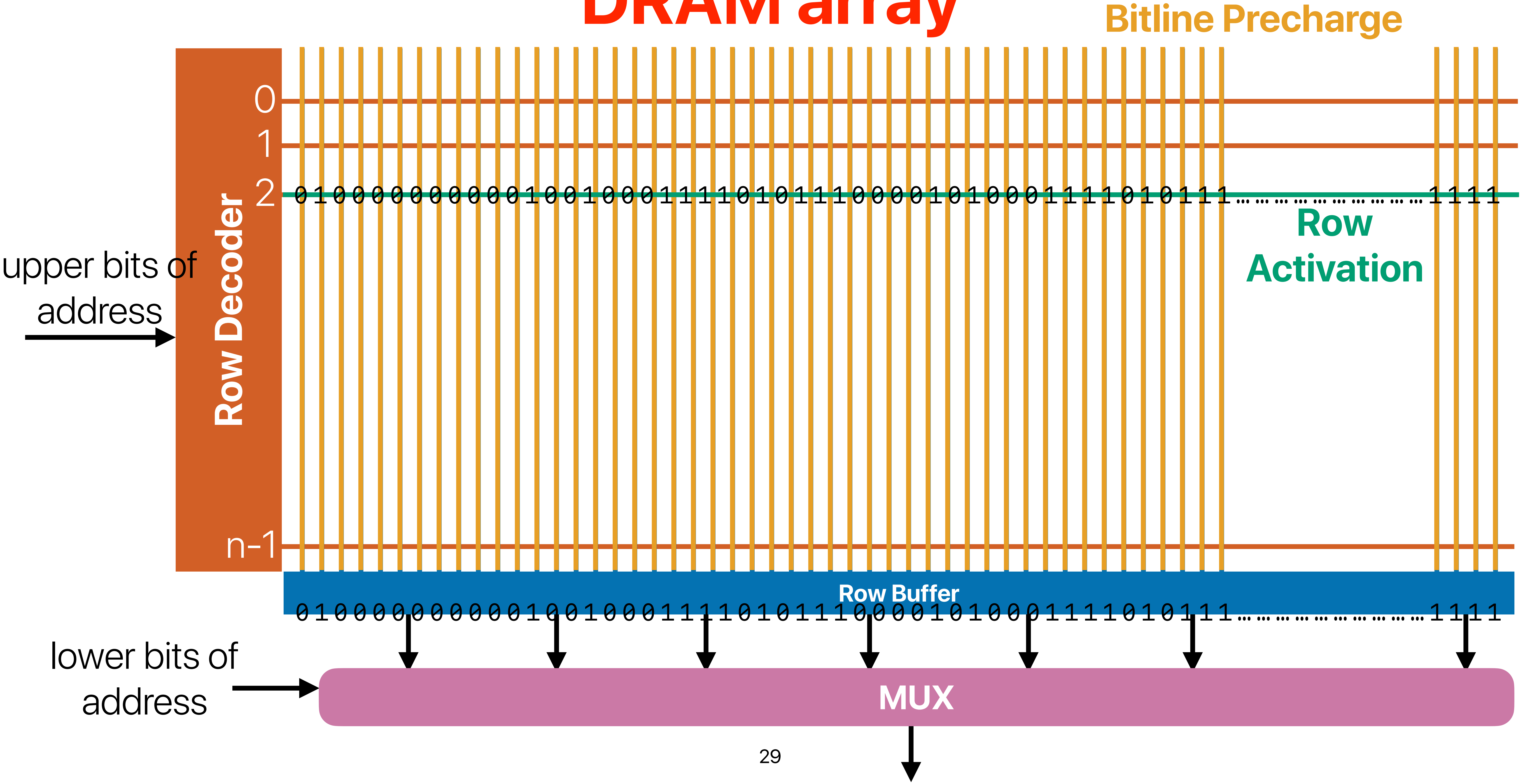
An DRAM cell



- 1 transistor (rather than 6)
- Relies on large capacitor to store bit
  - Write: transistor conducts, data voltage level gets stored on top plate of capacitor
  - Read: look at the value of data voltage

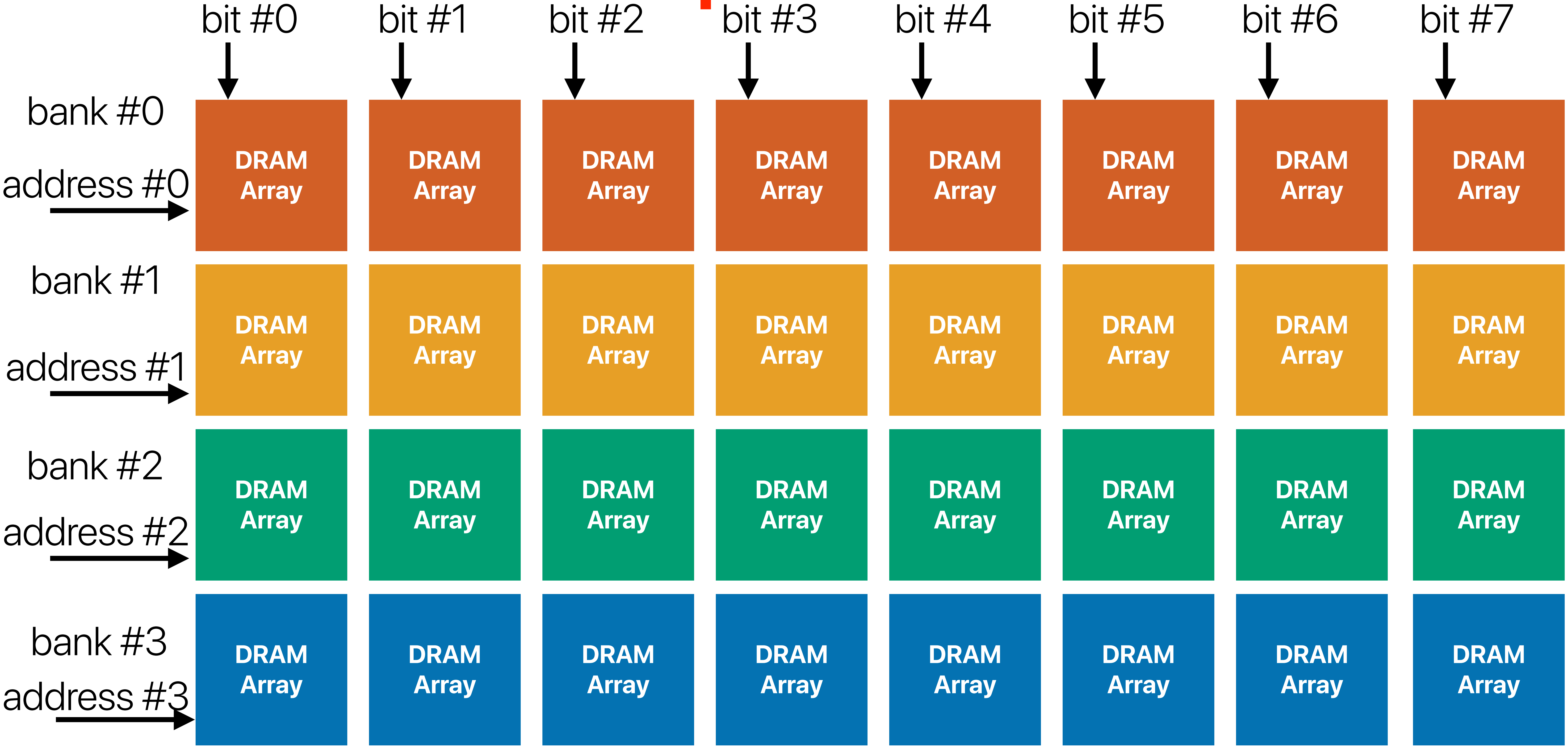


DRAM array



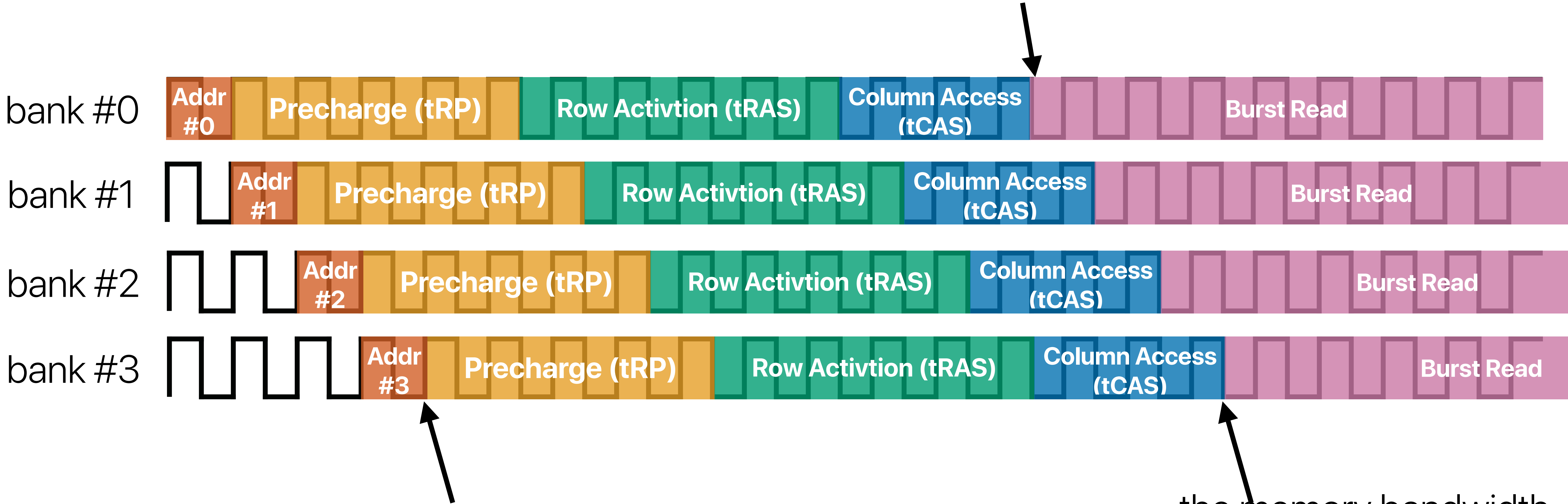
Bitline Precharge
Row Decoder
0
1
2
n-1
01000000000001001000111101011100001010001111010111 …………………… 1111
Row Activation
upper bits of address
Row Buffer
01000000000001001000111101011100001010001111010111 …………………… 1111
lower bits of address
MUX

Multiple Banks



Multi-bank access

we can start output a "byte" from every 8 chips each cycle after this

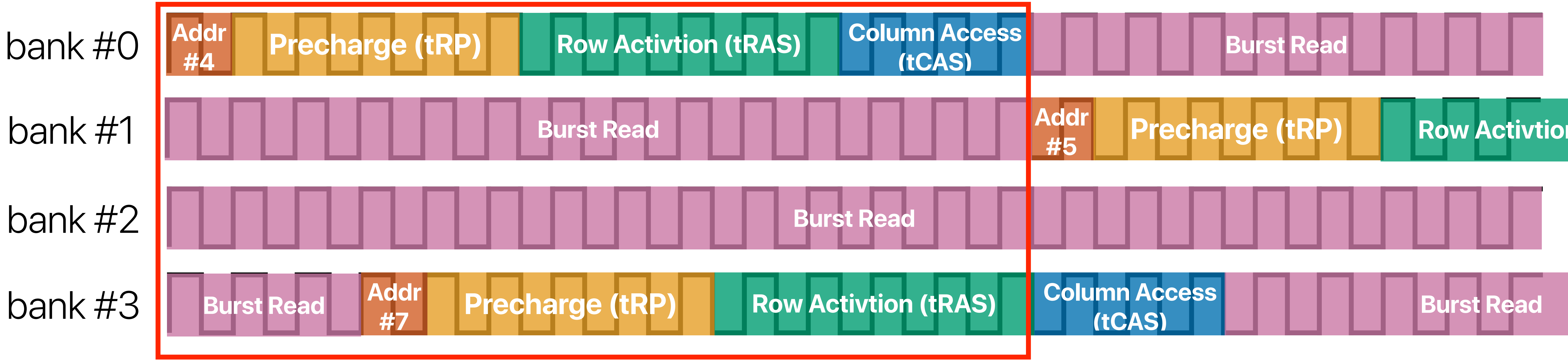


only one bank can accept request each cycle

the memory bandwidth can be fully utilized after this

Multi-bank access

The latency of pre-charge, row/column accesses is fully covered!



DRAM Performance

- Latency per "8-bit" — 0.75 ns (if it's row-buffered)
- Bandwidth per die = $\frac{1}{0.75ns} = 1.33GB/sec$
- 16 chips = $16 \times \frac{1}{0.75ns} = 21.33GB/sec$

2. Key Features

[Table 2] 8Gb DDR4 C-die Speed bins

Speed	DDR4-1600	DDR4-1866	DDR4-2133	DDR4-2400	DDR4-2666	Unit
	11-11-11	13-13-13	15-15-15	17-17-17	19-19-19	
tCK(min)	1.25	1.071	0.937	0.833	0.75	ns
CAS Latency	11	13	15	17	19	nCK
tRCD(min)	13.75	13.92	14.06	14.16	14.25	ns
tRP(min)	13.75	13.92	14.06	14.16	14.25	ns
tRAS(min)	35	34	33	32	32	ns
tRC(min)	48.75	47.92	47.06	46.16	46.25	ns

- JEDEC standard 1.2V (1.14V~1.26V)
- V_{DDQ} = 1.2V (1.14V~1.26V)
- V_{PP} = 2.5V (2.375V~2.75V)
- 800 MHz f_{CK} for 1600Mb/sec/pin, 933 MHz f_{CK} for 1866Mb/sec/pin, 1067MHz f_{CK} for 2133Mb/sec/pin, 1200MHz f_{CK} for 2400Mb/sec/pin, 1333MHz f_{CK} for 2666Mb/sec/pin
- 8 Banks (2 Bank Groups)
- Programmable CAS Latency (posted CAS): 10,11,12,13,14,15,16,17,18,19,20
- Programmable CAS Write Latency (CWL) = 9,11 (DDR4-1600), 10,12 (DDR4-1866),11,14 (DDR4-2133),12,16 (DDR4-2400) and 14,18 (DDR4-2666)
- 8-bit pre-fetch
- Burst Length: 8, 4 with tCCD = 4 which does not allow seamless read or write [either On the fly using A12 or MRS]
- Bi-directional Differential Data-Strobe
- Internal (self) calibration: Internal self calibration through ZQ pin (RZQ: 240 ohm ± 1%)
- On Die Termination using ODT pin
- Average Refresh Period 7.8us at lower than T_{CASE} 85°C, 3.9us at 85°C < $T_{CASE} \leq$ 95 °C
- Connectivity Test Mode (TEN) is Supported

The 8Gb DDR4 SDRAM C-die is organized as a 64Mbit x 16 I/Os x 8banks device. This synchronous device achieves high speed double-data-rate transfer rates of up to 2666Mb/sec/pin (DDR4-2666) for general applications.

The chip is designed to comply with the following key DDR4 SDRAM features such as posted CAS, Programmable CWL, Internal (Self) Calibration, On Die Termination using ODT pin and Asynchronous Reset.

All of the control and address inputs are synchronized with a pair of externally supplied differential clocks. Inputs are latched at the crosspoint of differential clocks (CK rising and $\overline{CK}$ falling). All I/Os are synchronized with a pair of bidirectional strobes (DQS and $\overline{DQS}$) in a source synchronous fashion. The address bus is used to convey row, column, and bank address information in a $\overline{RAS}/\overline{CAS}$ multiplexing style. The DDR4 device operates with a single 1.2V (1.14V~1.26V) power supply, 1.2V(1.14V~1.26V) V_{DDQ} and 2.5V (2.375V~2.75V) V_{PP}.

The 8Gb DDR4 C-die device is available in 96ball FBGAs(x16).

https://semiconductor.samsung.com/resources/user-manual/x16%20only_8G_C_DDR4_Samsung_Spec_Rev1.5_Apr.17.pdf

Roogle Claimed ideas

- Claimed ideas
  - Setup and benchmarking SmartSSDs
  - Parallel processing using matrix parallelism
  - Benchmarking fault-tolerant mechanisms in machine learning
  - Benchmarking and optimizing multimedia machine learning applications on heterogeneous computing resources
  - LLM on Edge TPUs/Embedded Systems
- Contact me if you've confirmed your project ideas but not seeing it in the slide — also send me your team member names
- Make an appointment through Google Calendar if you need consultant on projects

Roogle Other interesting ideas

- Can we use RT cores to accelerate other problems beyond identified ones?
- Can you improve the efficiency of existing problems on Tensor Cores (matrix decompositions, security algorithms)?
- Can you use other accelerated libraries (e.g., cuDNN, cuFFT) for applications beyond their domains?
- Can you think about some data representations that can make important compute kernels (e.g., machine learning algorithms, matrix multiplications) more efficient or secure?

Roogle Project Presentations

- Make an appointment on 2/1 and 2/6 through the Google Calendar
- 12 minute presentation with 3 minute Q & A
- Why & what & how!!! — considering you're giving a presentation at Apple's keynote
  - 6-minute why — why should everyone care about this problem? Why is this still a problem?
  - 4-minute what — what are you proposing in this project to address the problem?
  - 2-minute how — expected platforms/engineering efforts, milestones and workload distribution among members
  - Please reference this article to make a good presentation https://cseweb.ucsd.edu//~swanson/GivingTalks.html

Electrical
Computer Science
Engineering

つづく