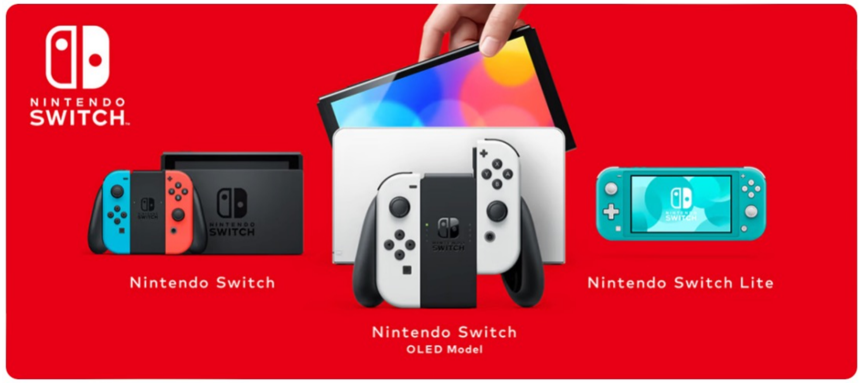
AI ASIC: Design and Practice (ADaP)

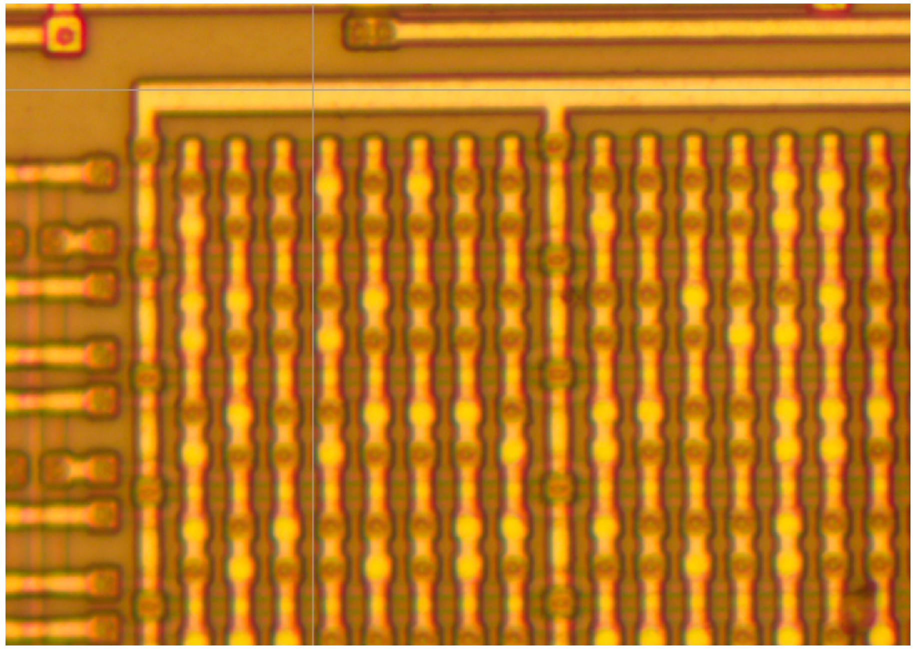
Fall 2023

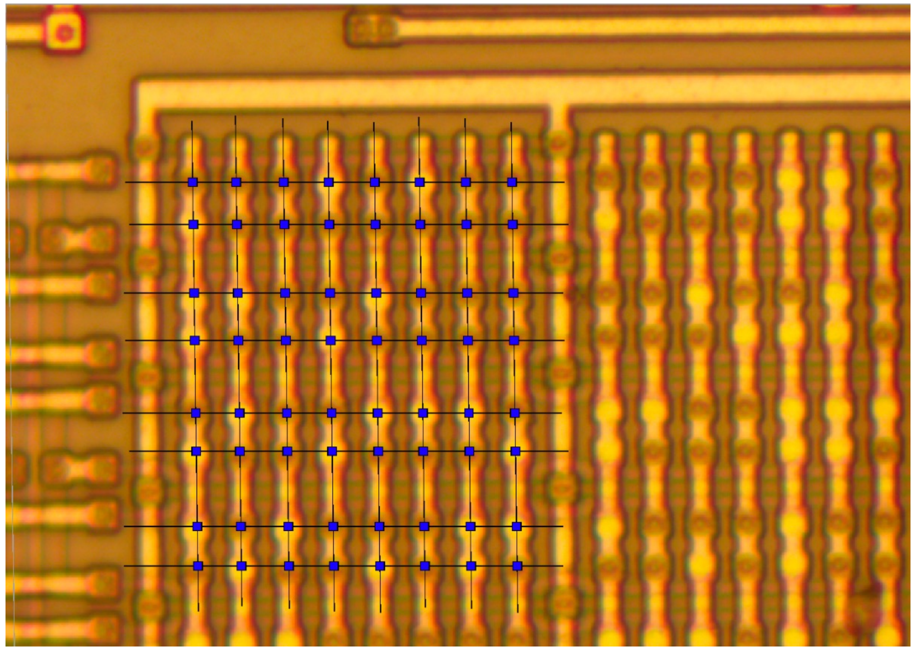
Memory Technologies

燕博南

Introduction







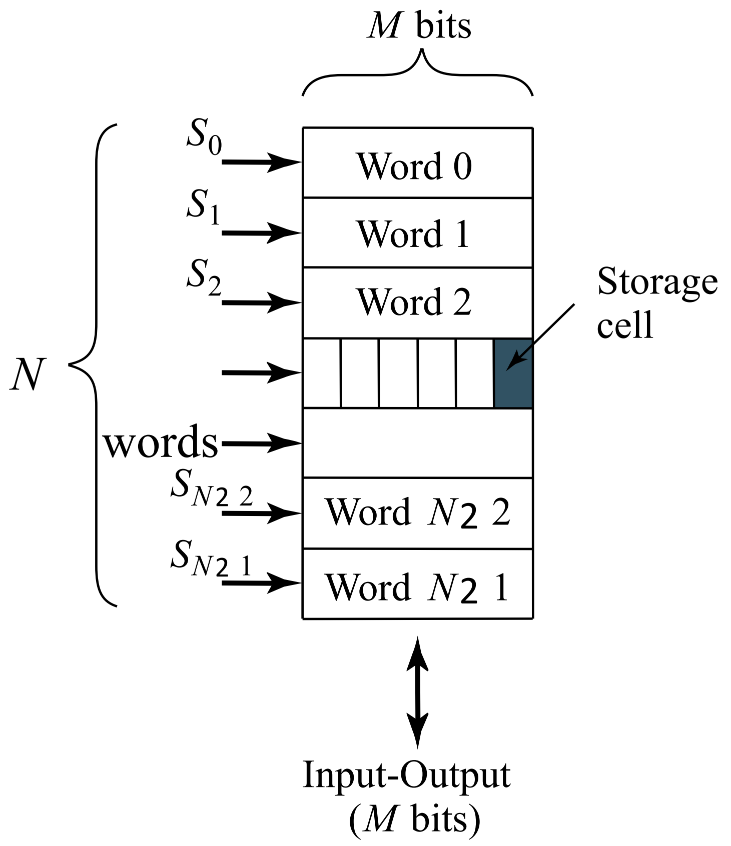
Outline

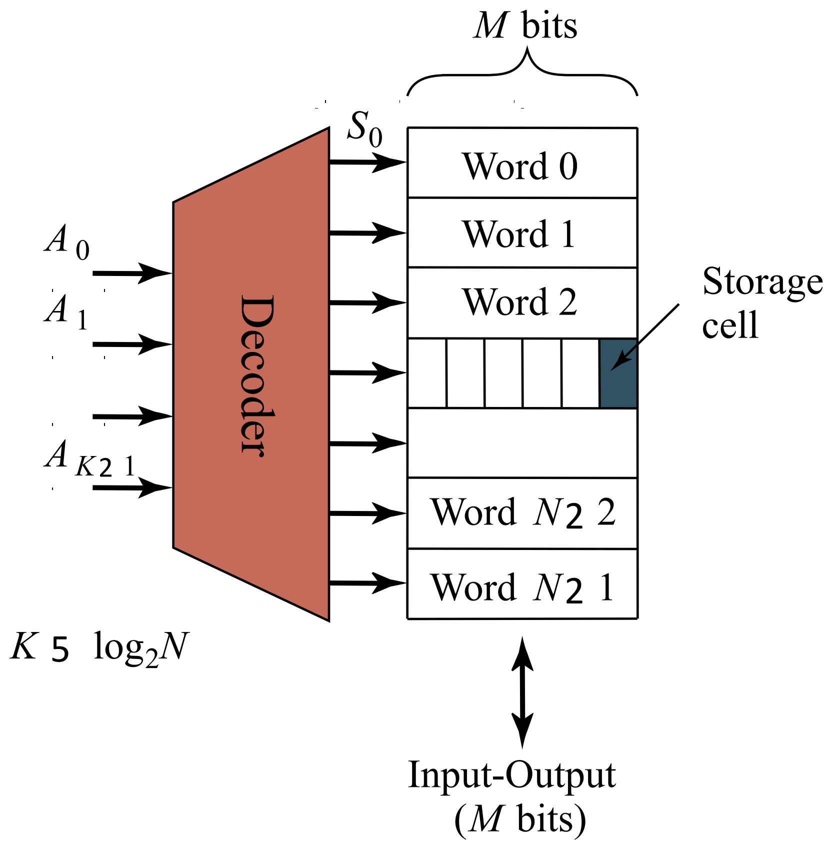
- Memory Types
- Memory Organization
- ROM design
- RAM design
- PLA design

Memory Types

Read-Write Memory		**Non-Volatile Read-Write Memory**	**Read-Only Memory**
Random Access	**Non-Random Access**	EPROM	Mask-Programmed
		E^2PROM	Programmable (PROM)
SRAM	FIFO	FLASH	
DRAM	LIFO	RRAM	
	Shift Register	*MRAM*	
	CAM	*PCM*	

Memory Spatial Abstraction

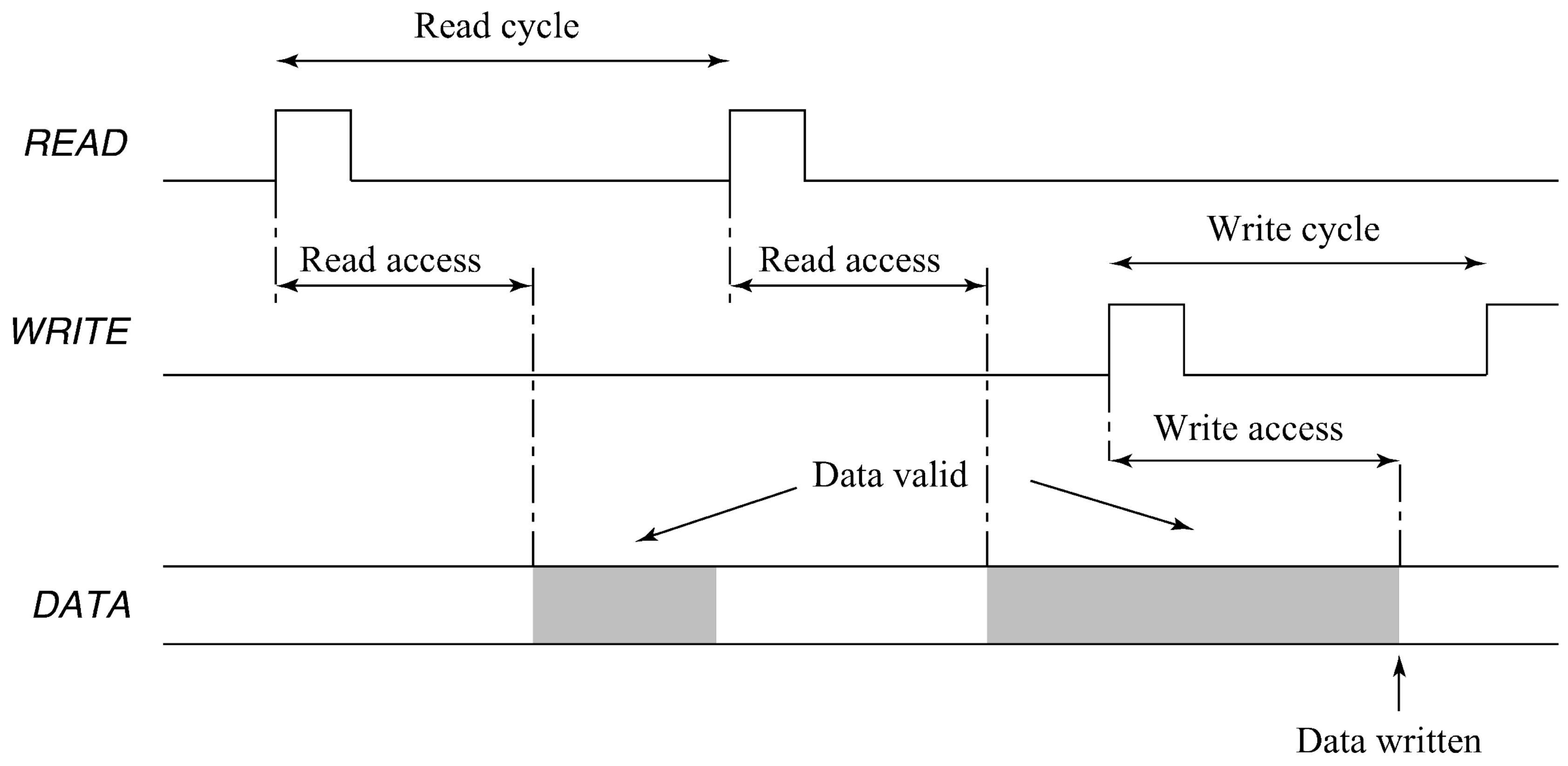




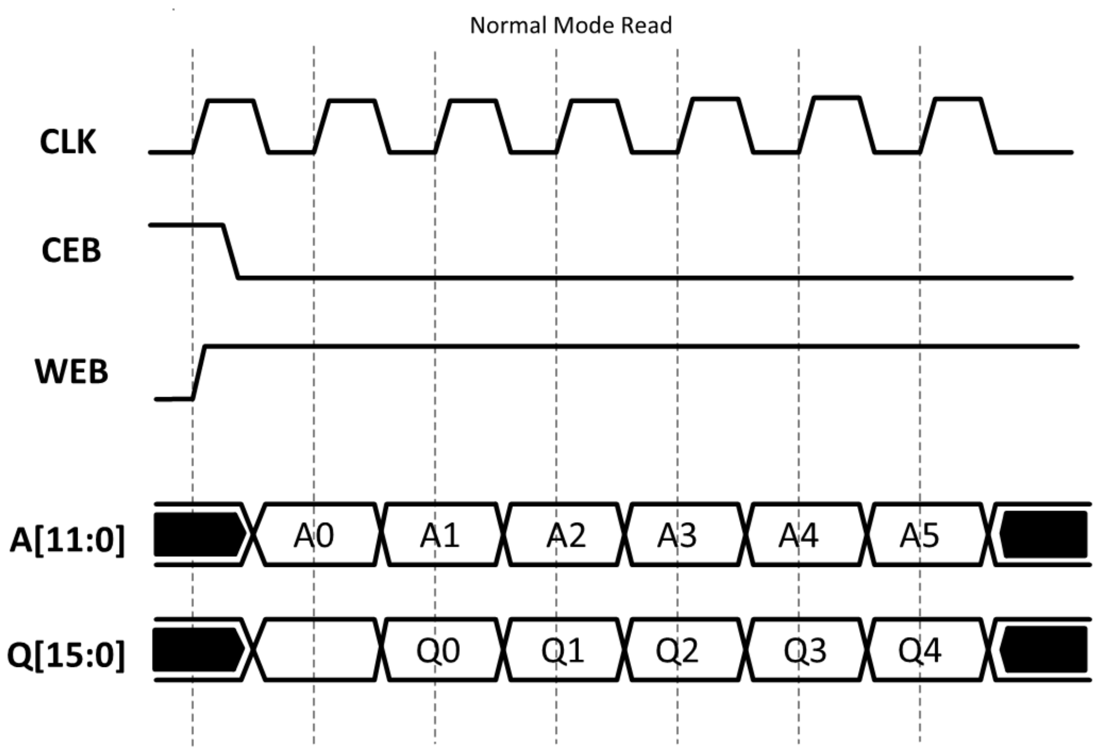
Intuitive architecture for N x M memory
Too many select signals:
N words == N select signals

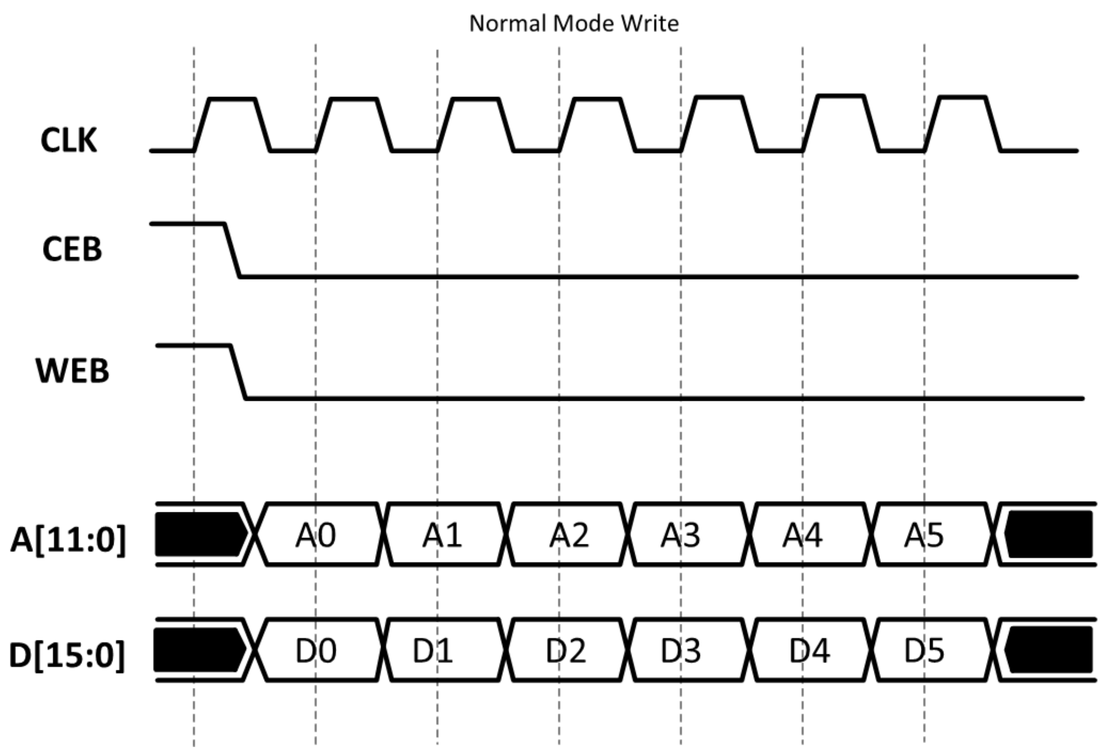
Decoder reduces the number of select signals
$K = log_2 N$

Memory Timing Behavior

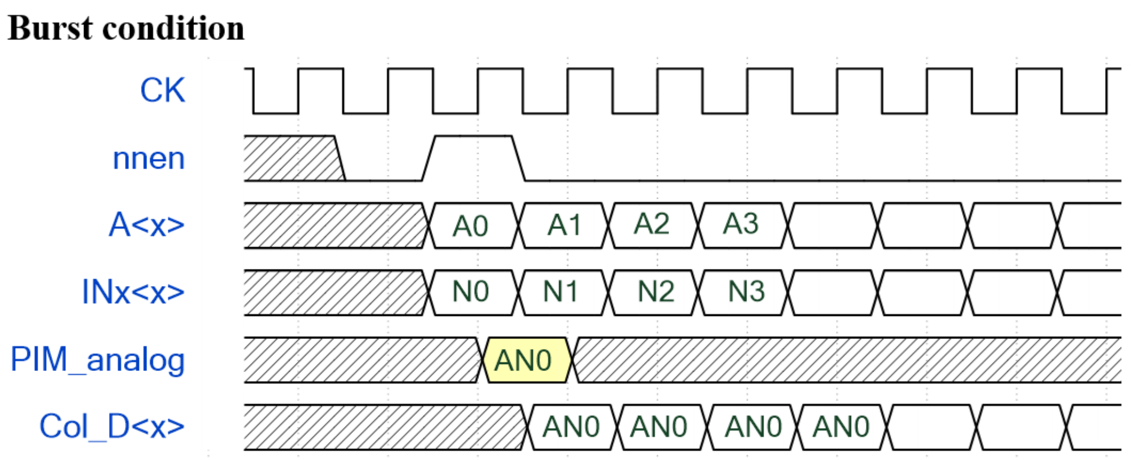


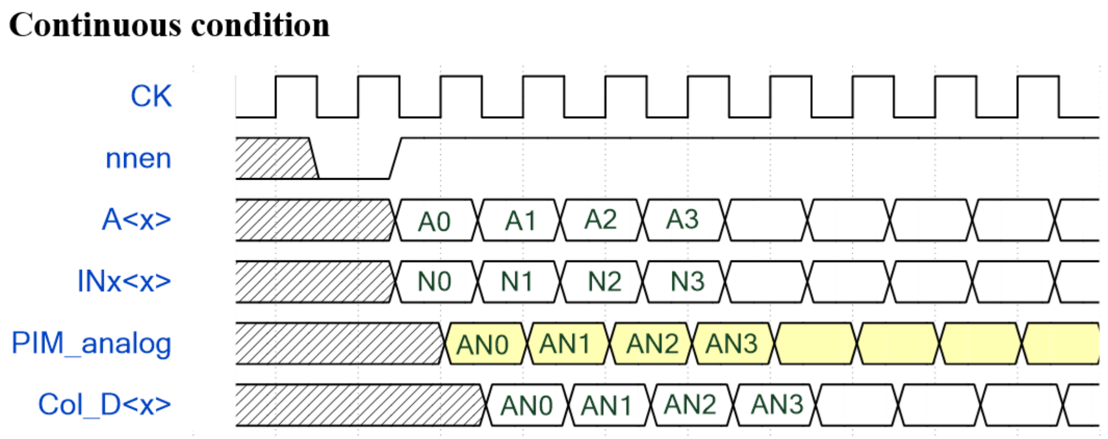
Memory Timing Behavior





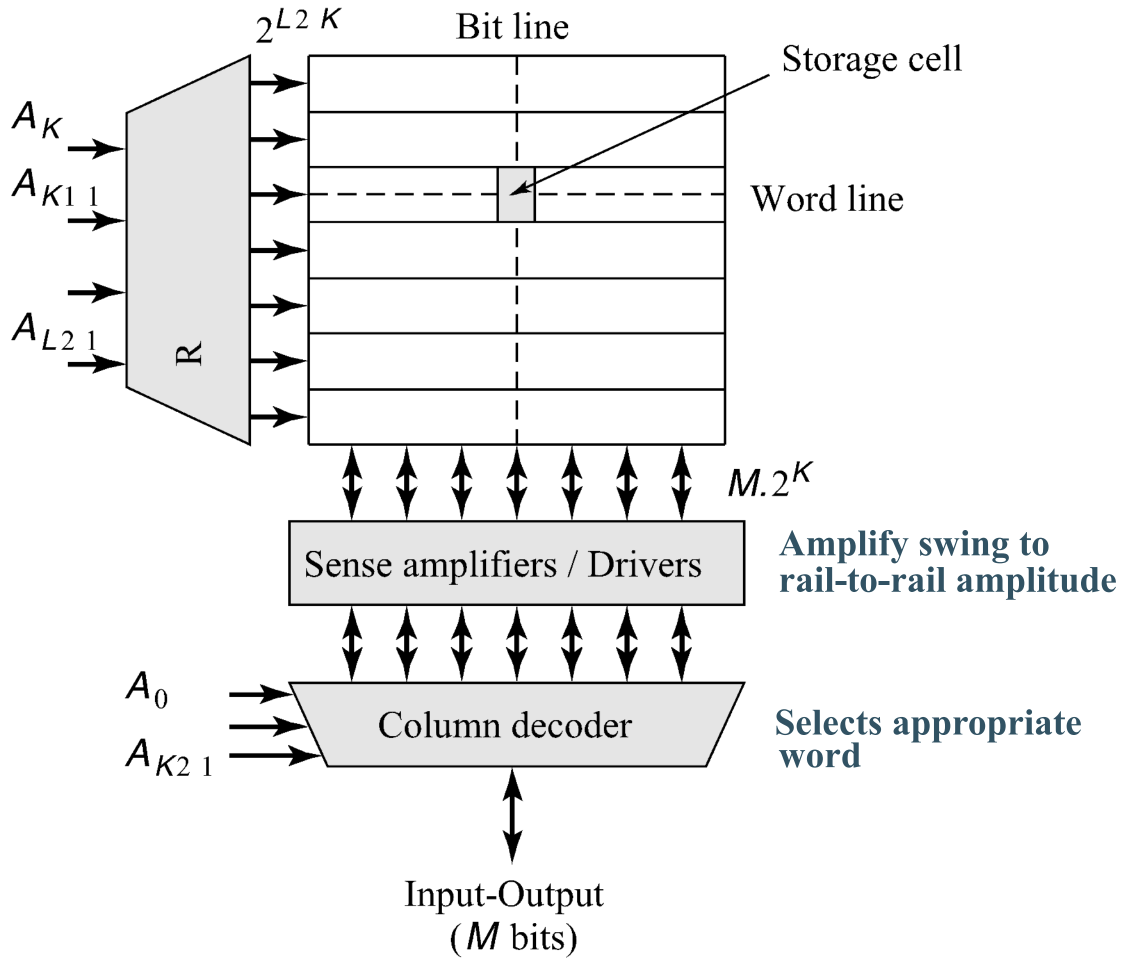
Memory Timing Behavior / Compute-In-Memory

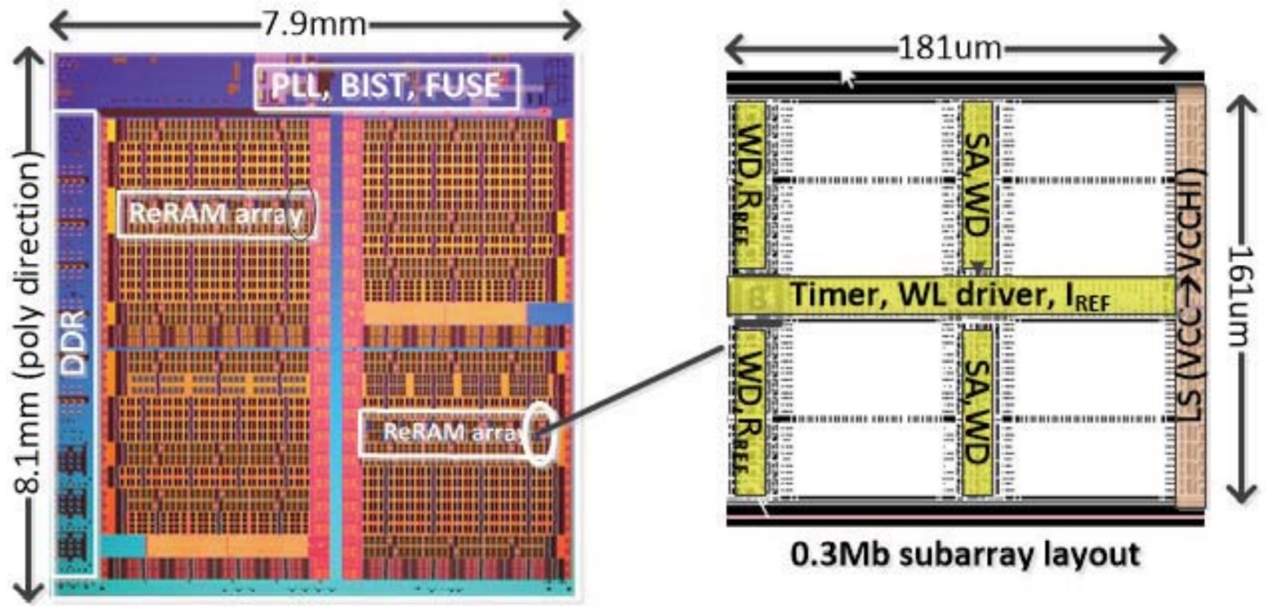




- **Add additional (compute mode) inputs**
- **Perhaps additional address bits**

Memory Architecture

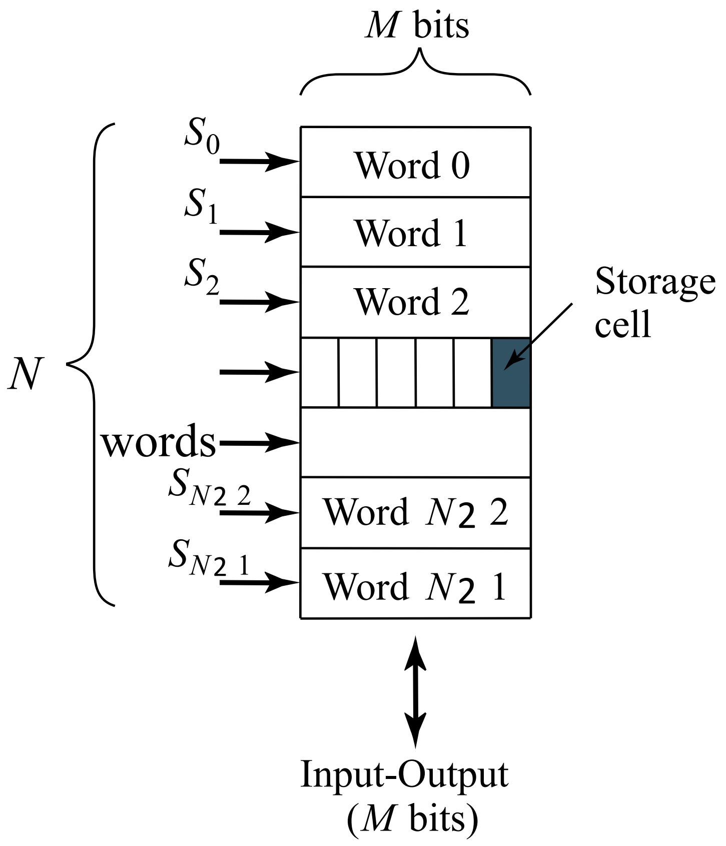




How it can achieve large capacity?

Jain, Pulkit, et al. "13.2 A 3.6 Mb 10.1 Mb/mm 2 embedded non-volatile ReRAM macro in 22nm FinFET technology with adaptive forming/set/reset schemes yielding down to 0.5 V with sensing time of 5ns at 0.7 V." *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019.

Memory Address



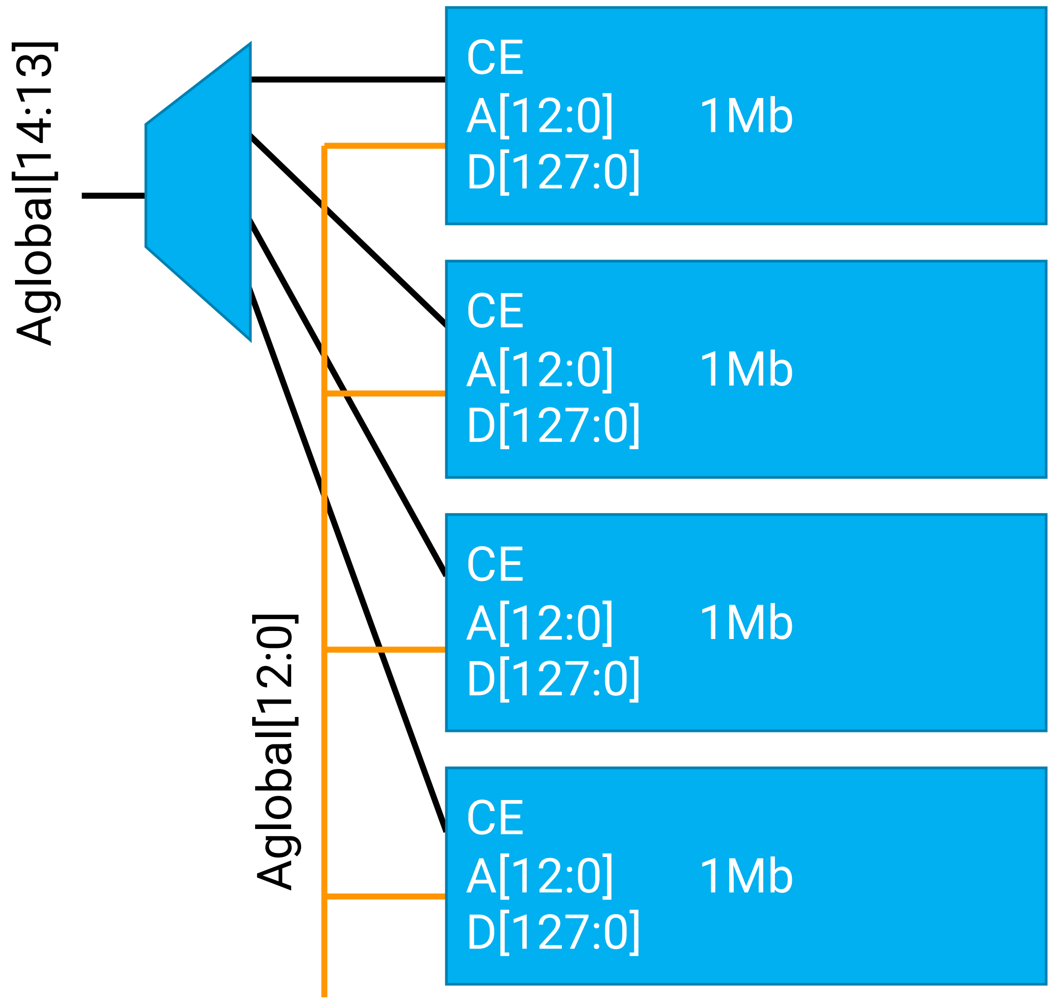
Each memory I/O bit width is 128bit

For 1Mb memory, what is the range of memory address?

1Mb/128b
= $(2^{20}$ bit$)/(2^{7}$ bit$)$
= 2^{13}

13-wire address is necessary

Memory Architecture (inside a memory block)



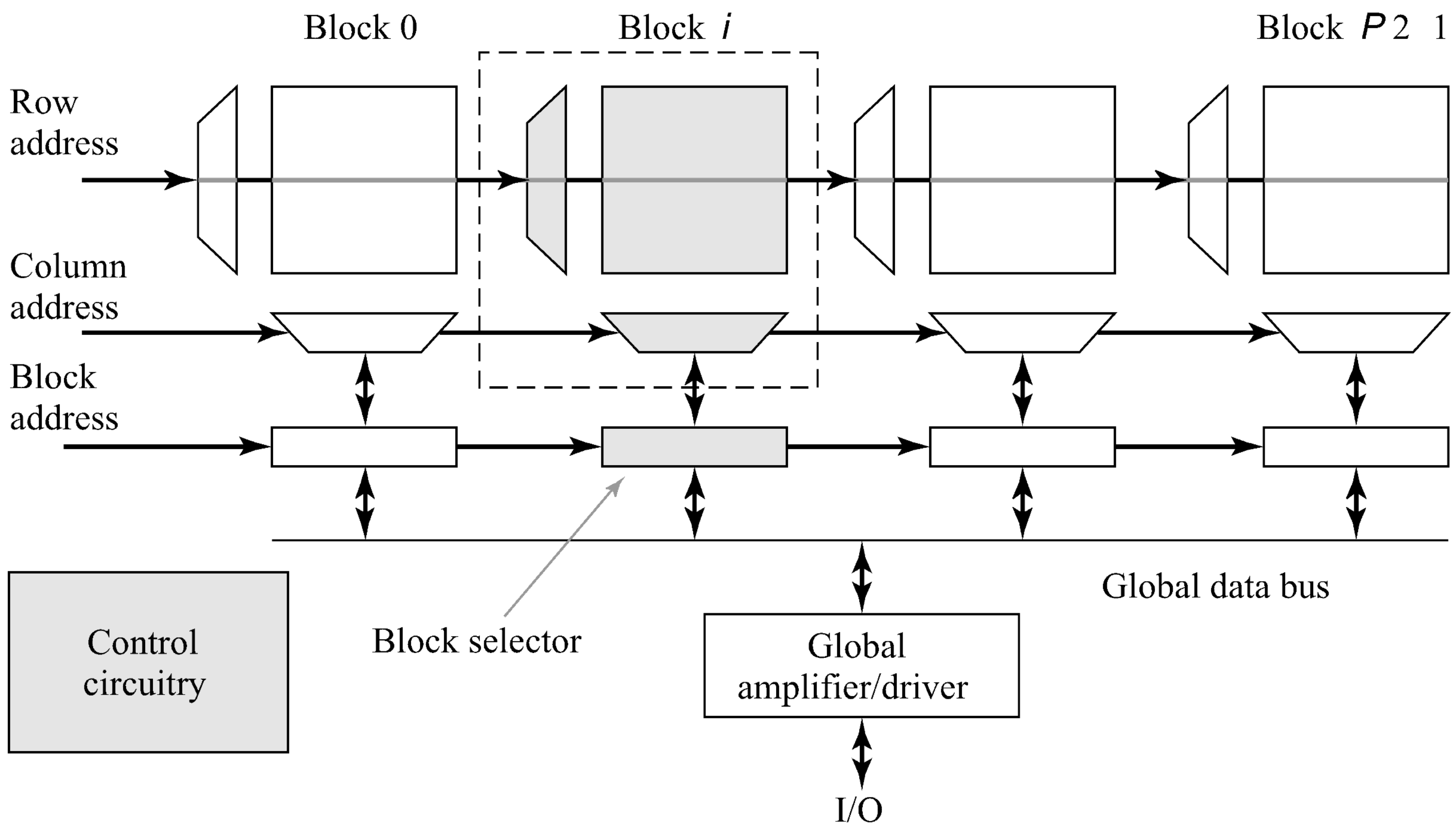
Assume 1Mb is one subarray

Configuration:

- Memory I/O bit width is 128b
- 1Mb/subarray
- Address width: 13b

How get 4Mb memory?

Hierarchical Memory Architecture



Advantages:

1. Shorter wires within blocks
2. Block address activates only 1 block => power savings

Various Memories

- Read-Only-Memory (ROM)
- Random-Access-Memory (RAM) : Read/Write Memory
  - SRAM
  - DRAM

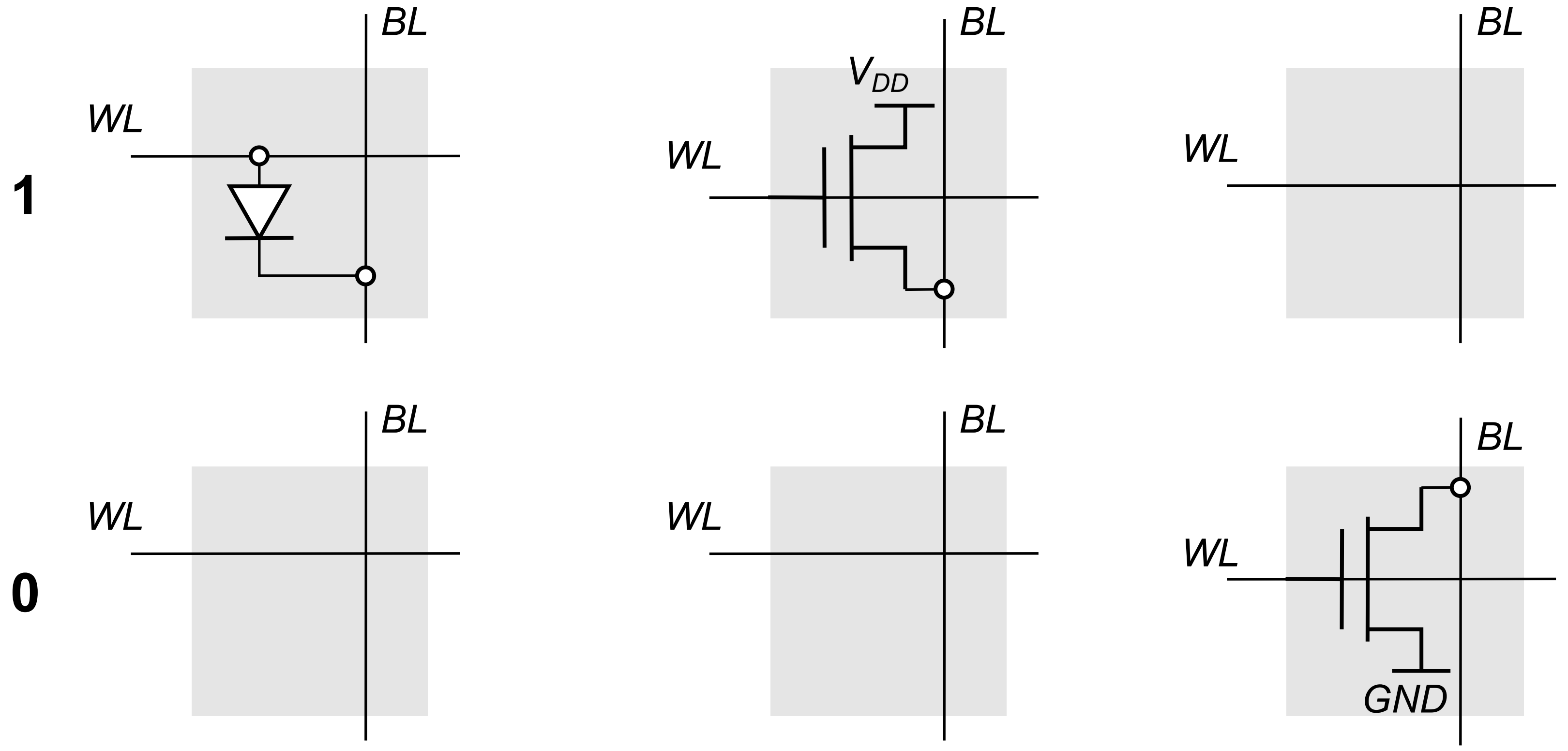
Before we introduce them, think first: what are good memories?

Density, R/W Speed, Endurance, Retention, Nonvolatility, …

Various Memories

- Read-Only-Memory (ROM)
- Random-Access-Memory (RAM) : Read/Write Memory
  - SRAM
  - DRAM

ROM Cells

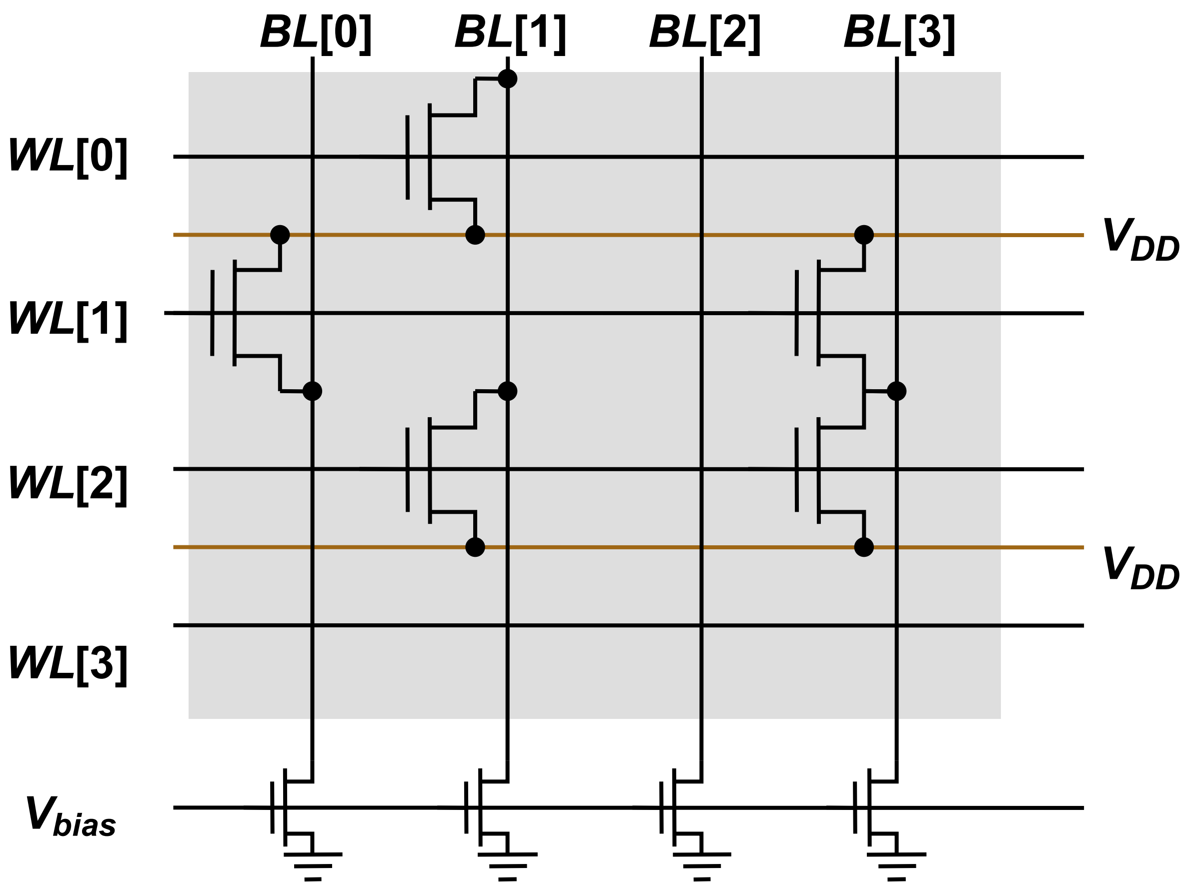


Diode ROM

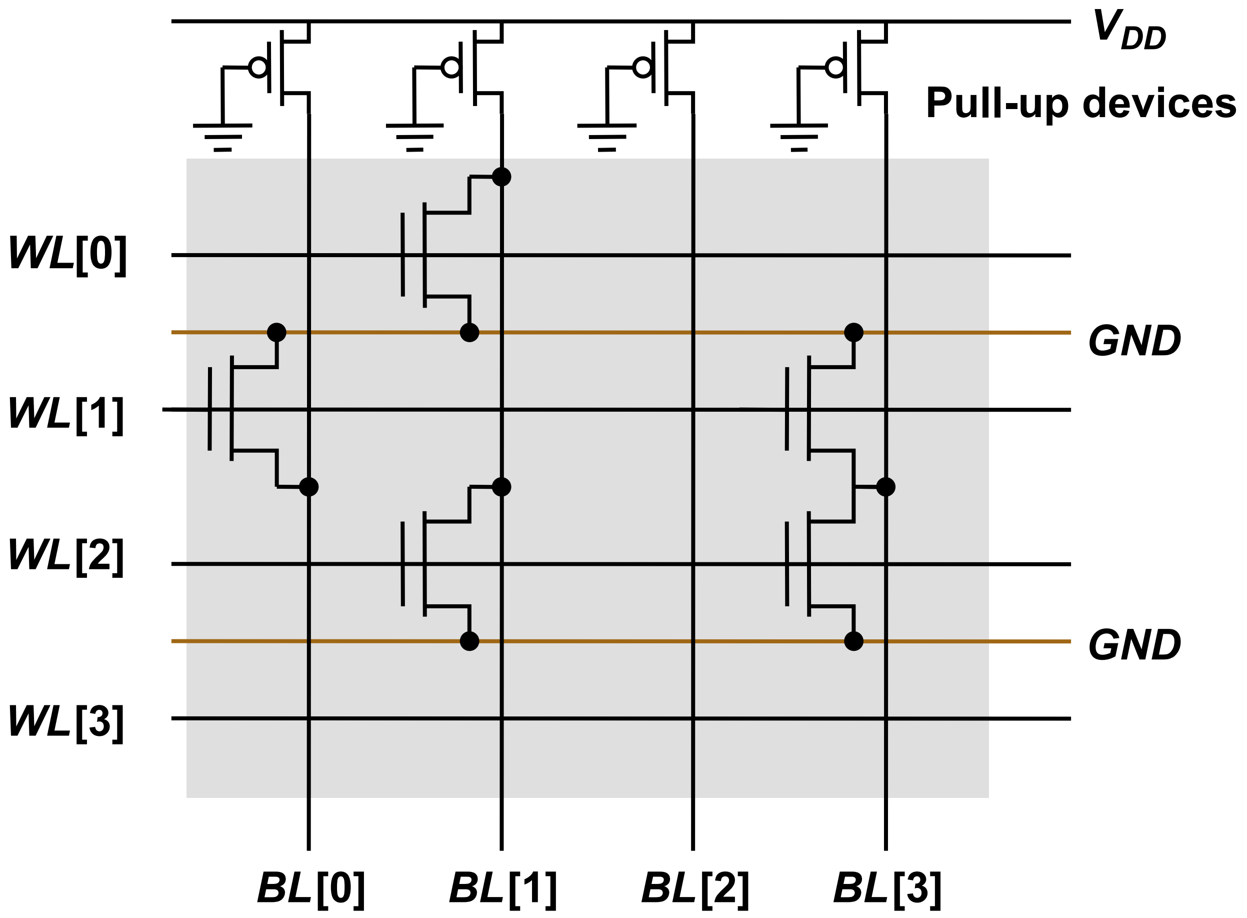
MOS ROM 1

MOS ROM 2

MOS ROM



$BL[0]$
$BL[1]$
$BL[2]$
$BL[3]$
$WL[0]$
$WL[1]$
$WL[2]$
$WL[3]$
V_{DD}
V_{DD}
V_{bias}

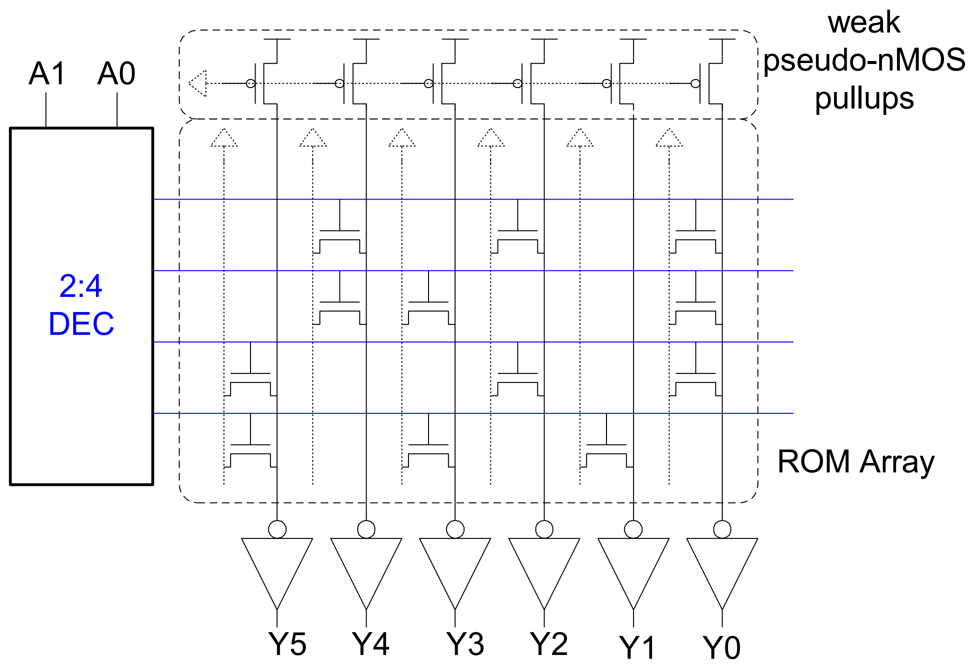


V_{DD}
Pull-up devices
$WL[0]$
$WL[1]$
$WL[2]$
$WL[3]$
GND
GND
$BL[0]$
$BL[1]$
$BL[2]$
$BL[3]$

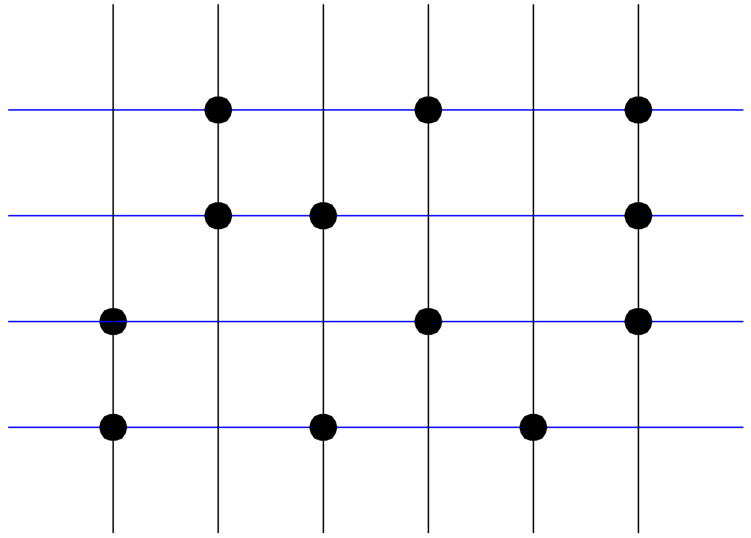
MOS ROM Example

- 4-word x 6-bit ROM
  - Represented with dot diagram
  - Dots indicate 1's in ROM

Word 0:	010101
Word 1:	011001
Word 2:	100101
Word 3:	101010

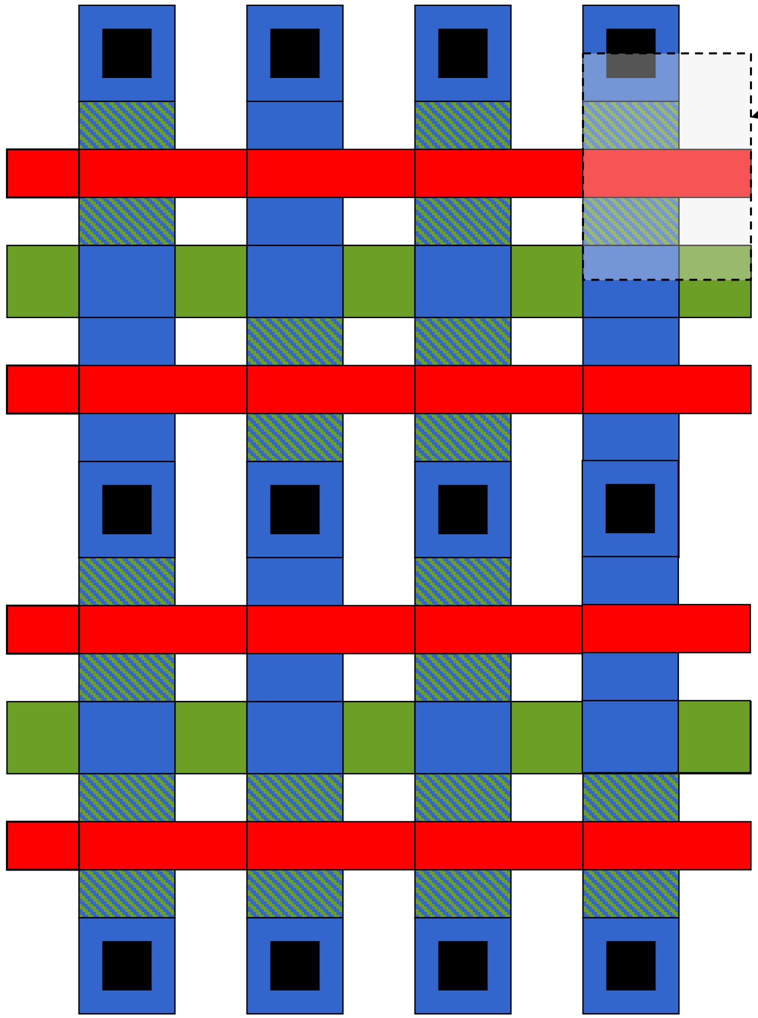


Looks like 6 4-input pseudo-nMOS NORs



Layout

北京大學 PEKING UNIVERSITY



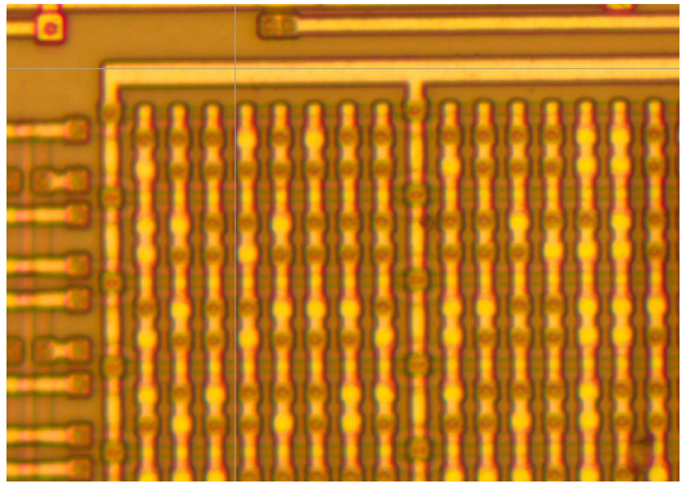
Programmming using the Active Layer Only

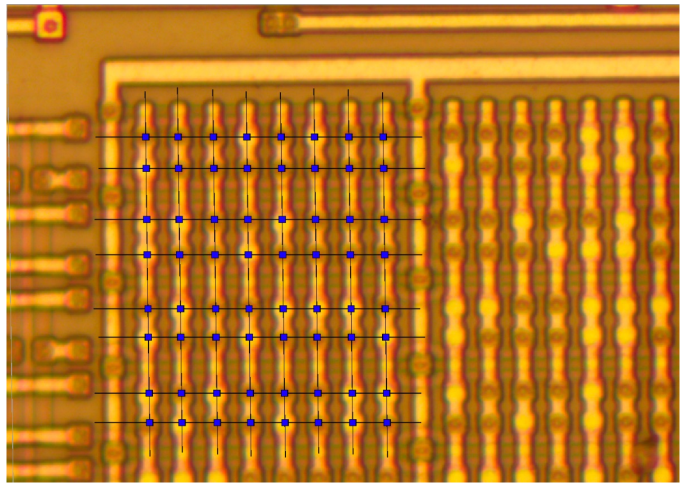
🟥 Polysilicon

🟦 Metal1

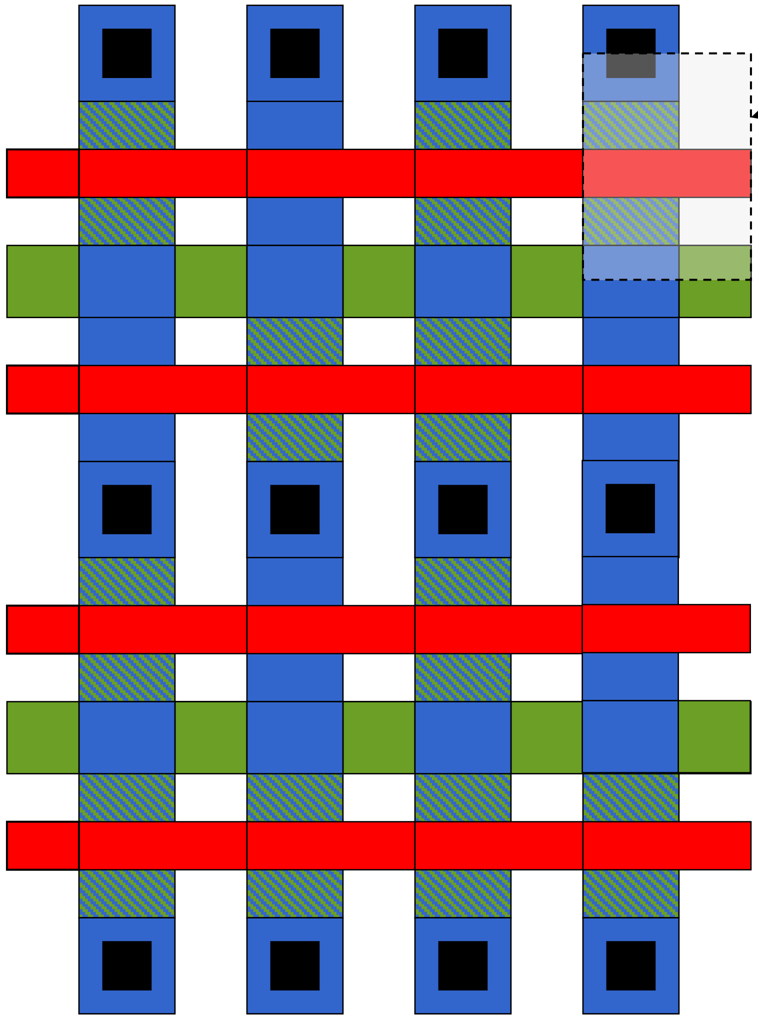
🟩 Diffusion

▨ Metal1 on Diffusion





Layout



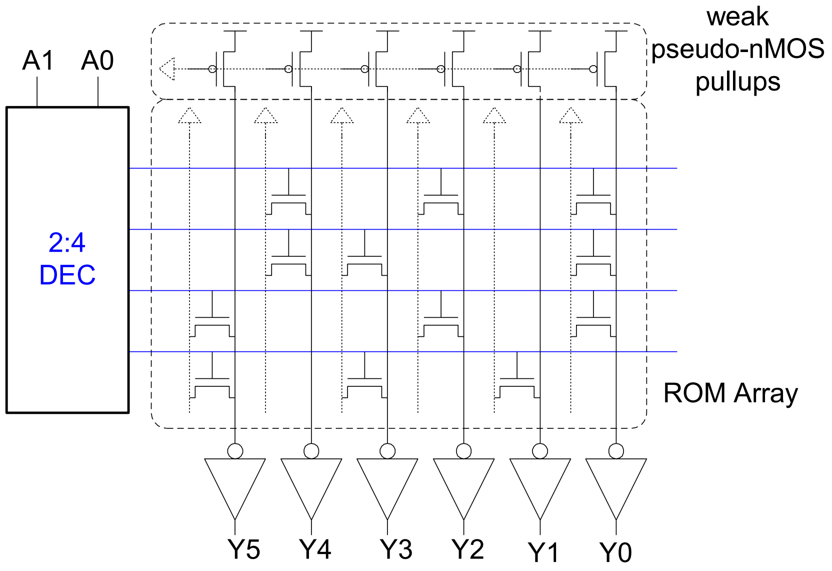
Cell (*19f* x 14*f*)

f: feature size

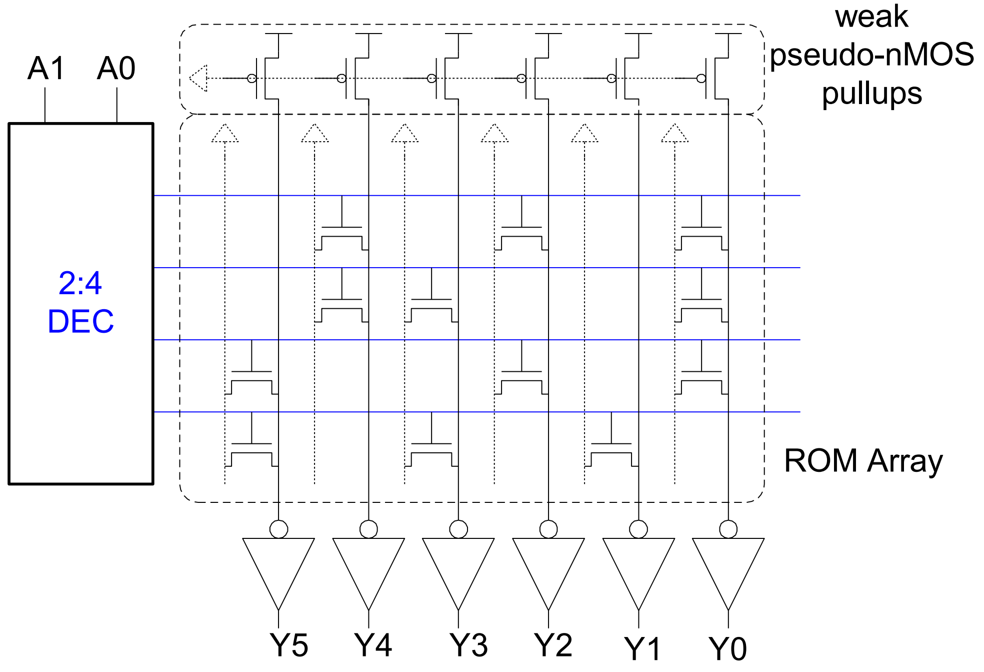
Programmming using the Active Layer Only

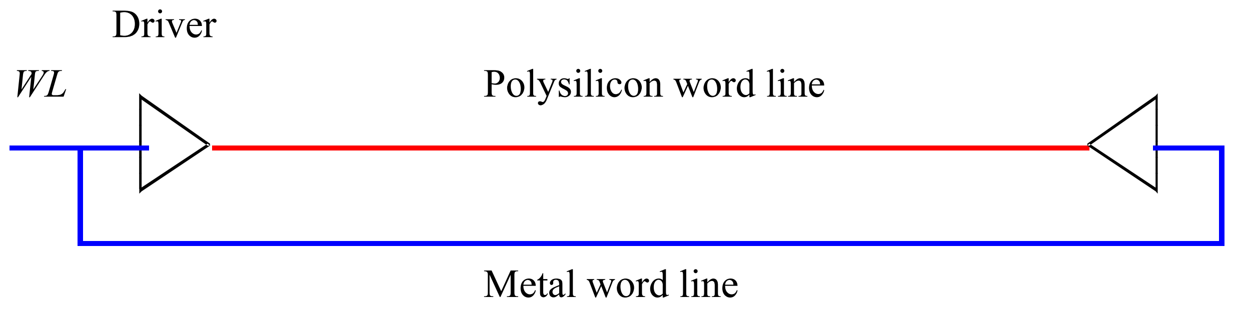
Q: Can this array expand to infinite size?

🟥	Polysilicon
🟦	Metal1
🟩	Diffusion
▨	Metal1 on Diffusion

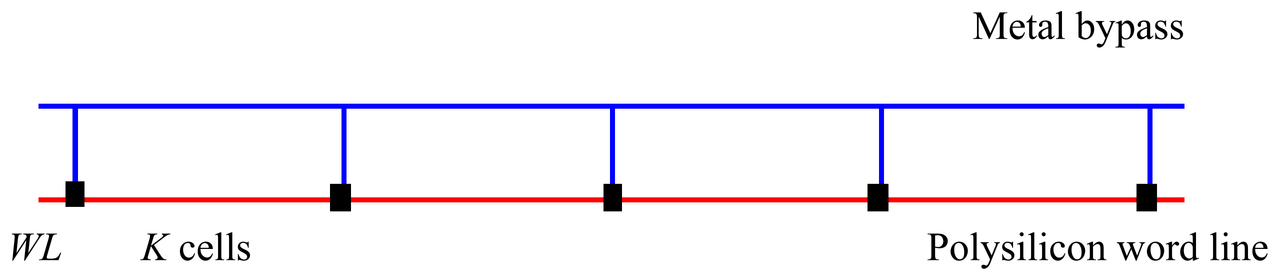


Decreasing Word Line Delay





(a) Driving the word line from both sides



(b) Using a metal bypass

Various Memories

- Read-Only-Memory (ROM)
- Random-Access-Memory (RAM) : Read/Write Memory
  - SRAM
  - DRAM

RAM Types

❑ STATIC (SRAM)

Data stored as long as supply is applied
Large (6 transistors/cell)
Fast
Differential

❑ DYNAMIC (DRAM)

Periodic refresh required
Small (1-3 transistors/cell)
Slower
Single Ended

RAM Types

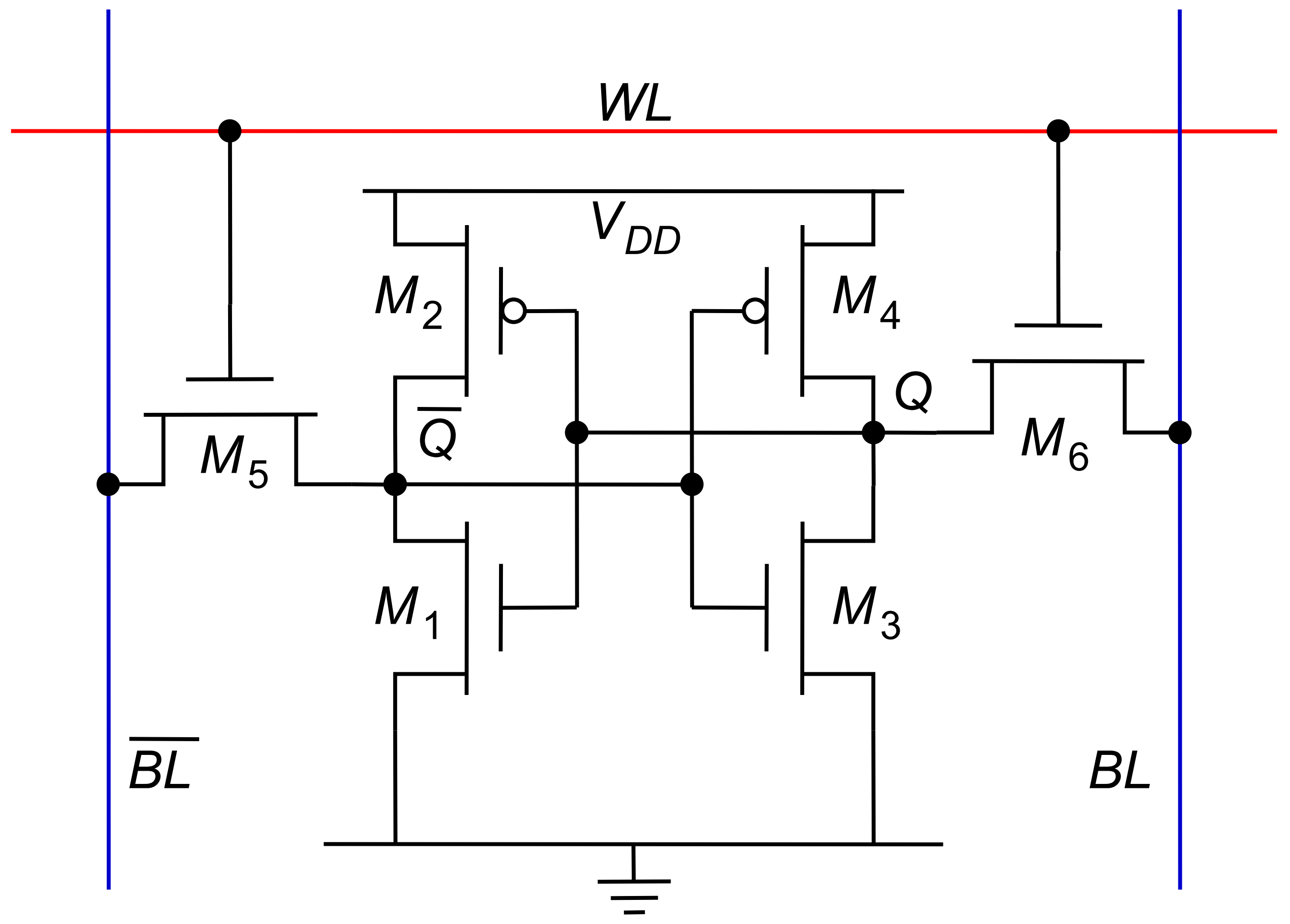
- ❑ STATIC (SRAM)

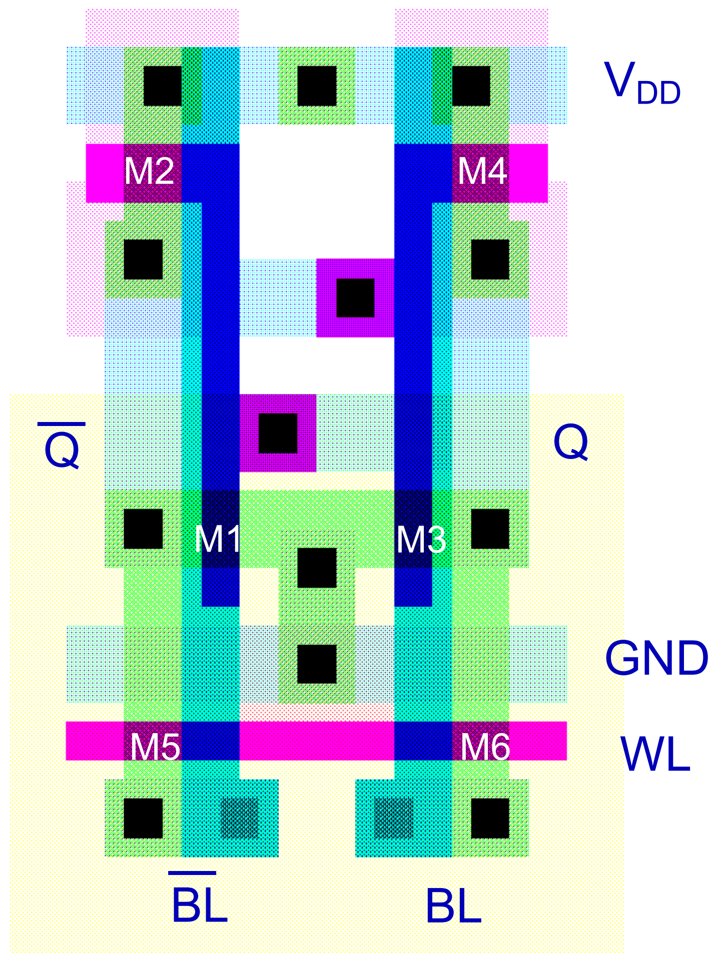
  Data stored as long as supply is applied
  Large (6 transistors/cell)
  Fast
  Differential

- ❑ DYNAMIC (DRAM)

  Periodic refresh required
  Small (1-3 transistors/cell)
  Slower
  Single Ended

SRAM 6T Cell

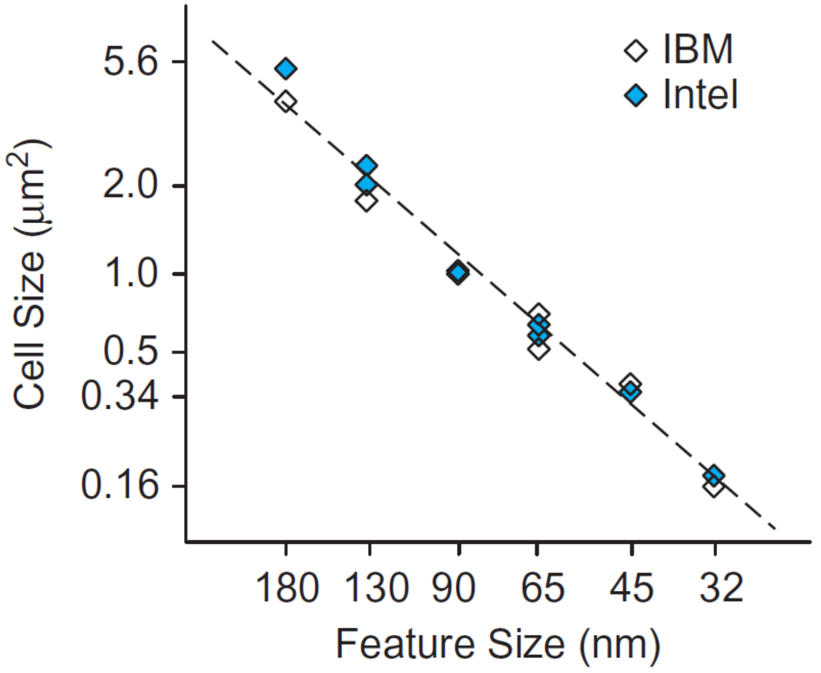


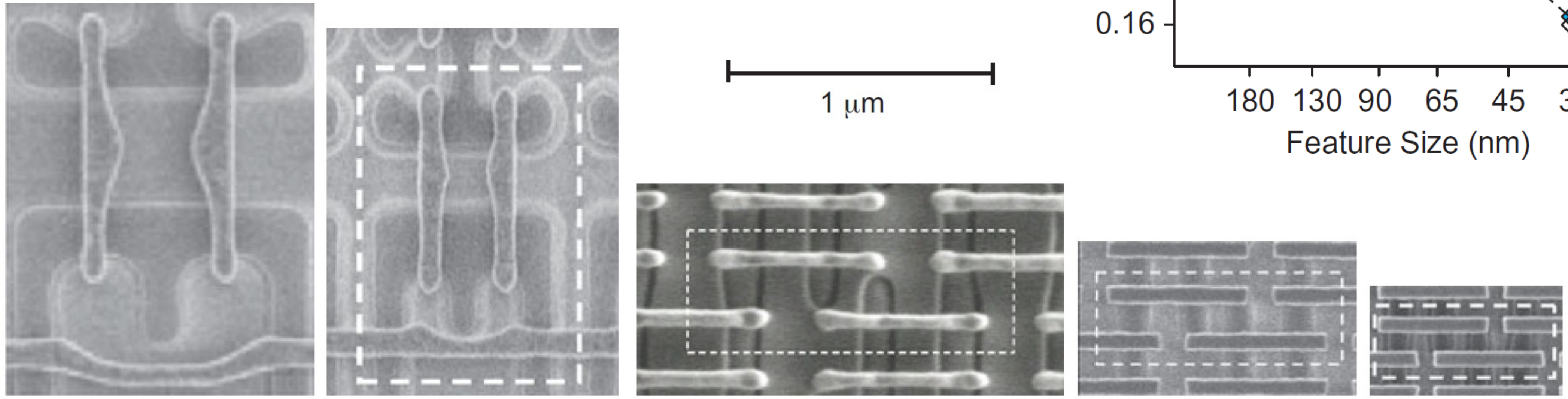


Push Rule: special design rule for SRAM Transistors

SRAM Scaling

SRAM is the first component for foundry to develop





130 nm [Tyagi00]

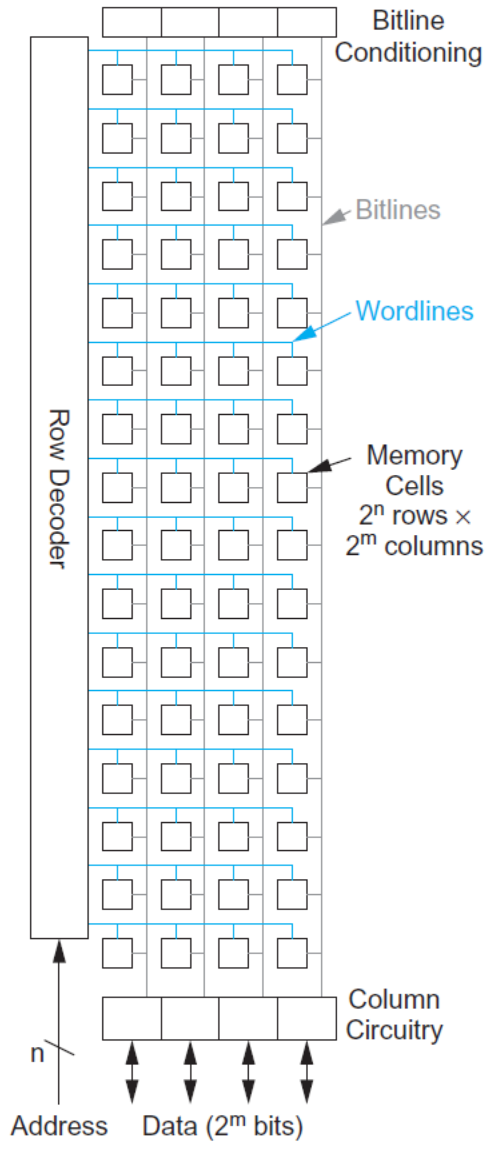
90 nm [Thompson02]

65 nm [Bai04]

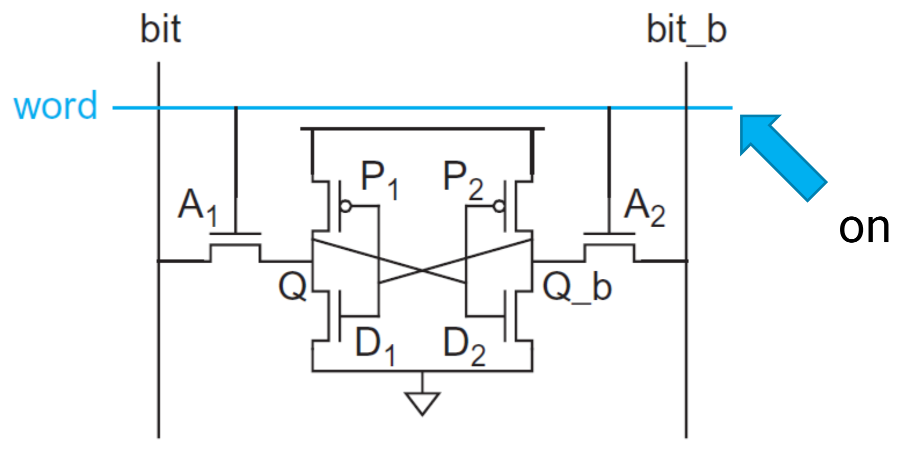
45 nm [Mistry07]

32 nm [Natarajan08]

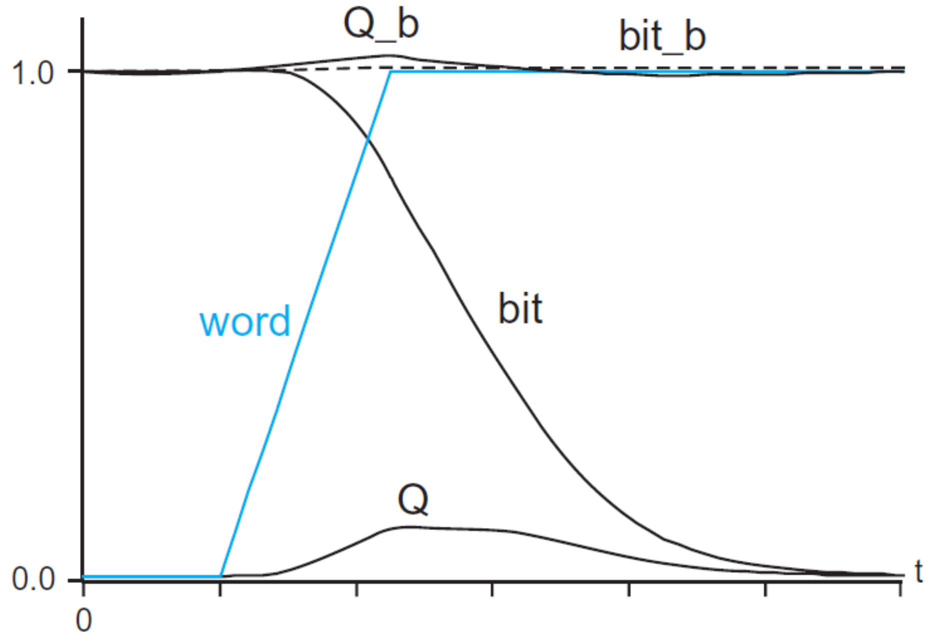
SRAM Bitcell Array



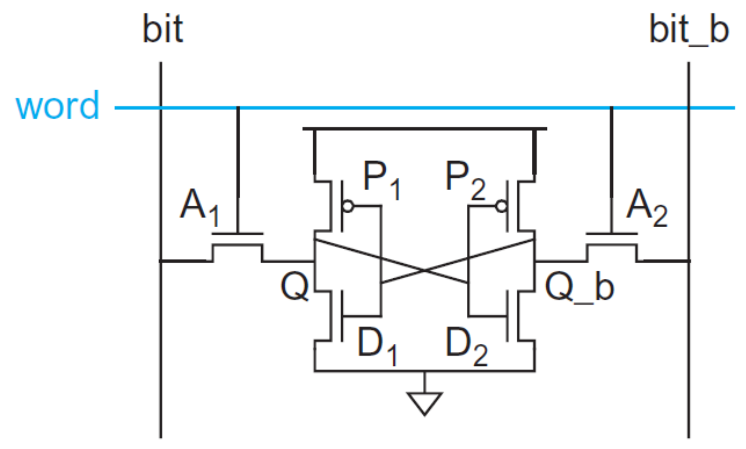
Read Operation:

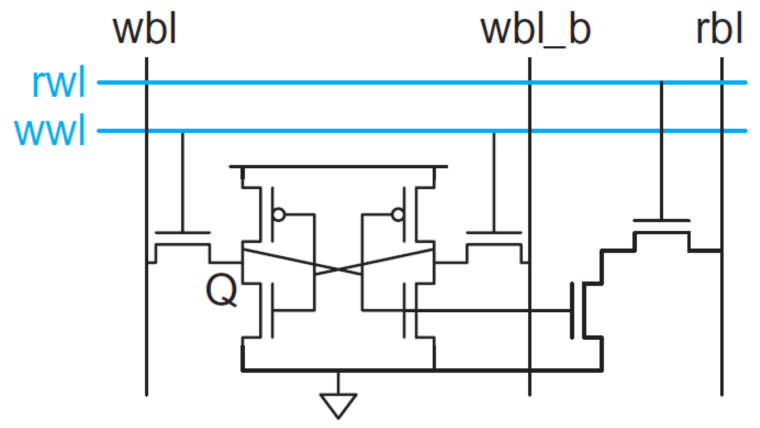


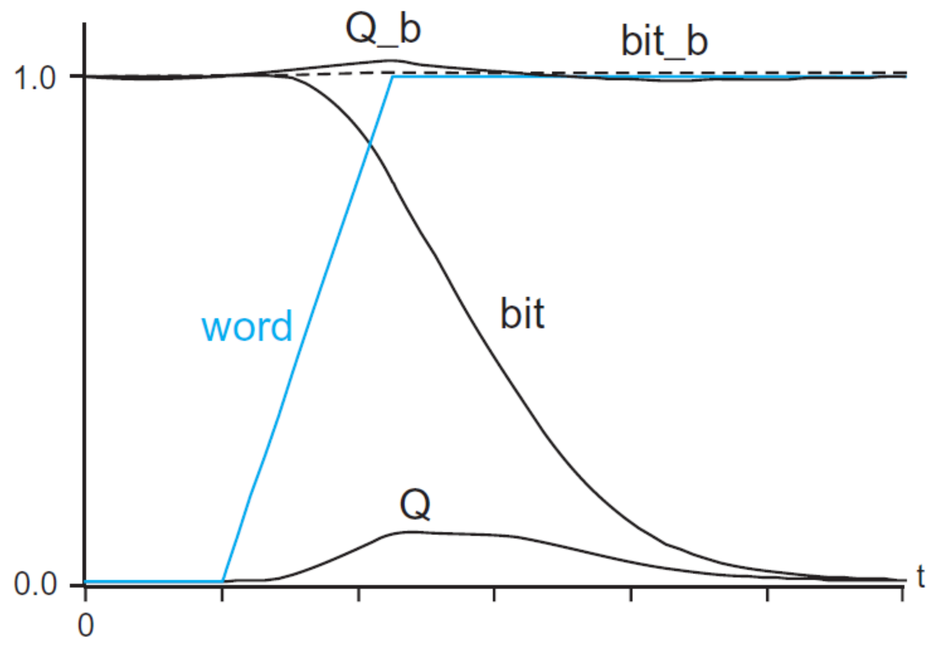
Timing:



8T SRAM







- Advantage: reduce read disturbance
- Disadvantage: Too large
- Read disturbance:
  - Unexpectedly change bitcell data when read

RAM Types

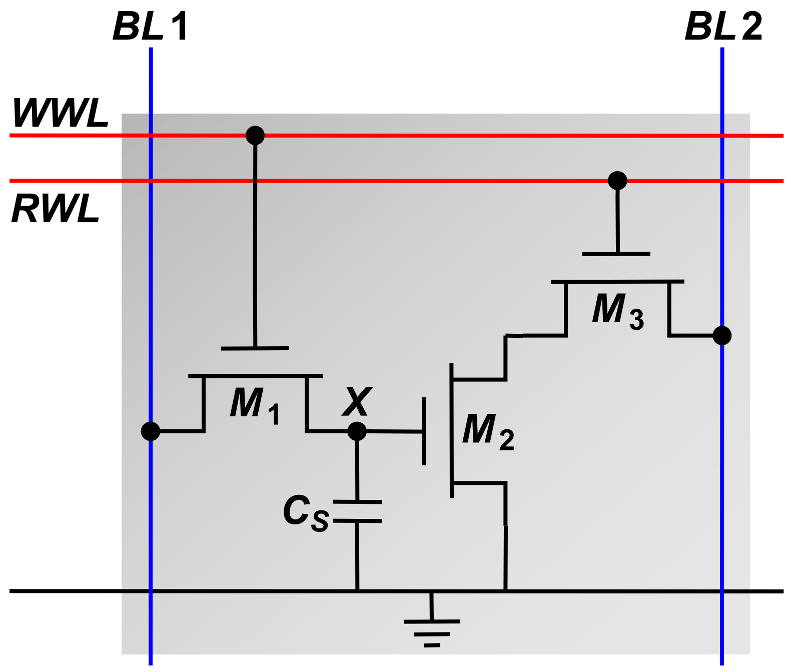
- ❑ STATIC (SRAM)

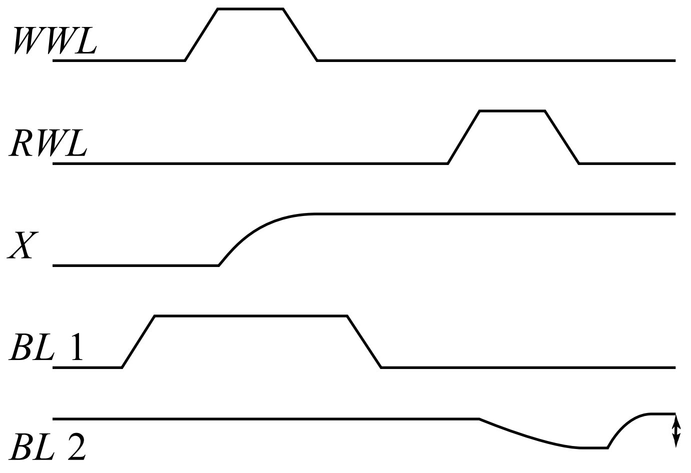
  Data stored as long as supply is applied
  Large (6 transistors/cell)
  Fast
  Differential

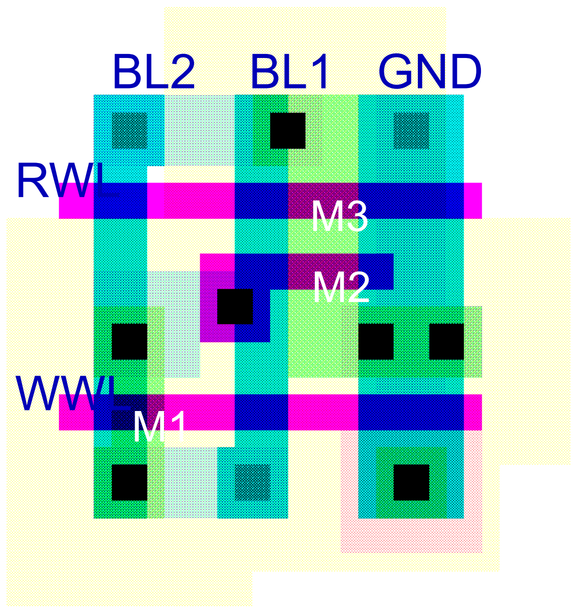
- ❑ DYNAMIC (DRAM)

  Periodic refresh required
  Small (1-3 transistors/cell)
  Slower
  Single Ended

DRAM Cell 3T



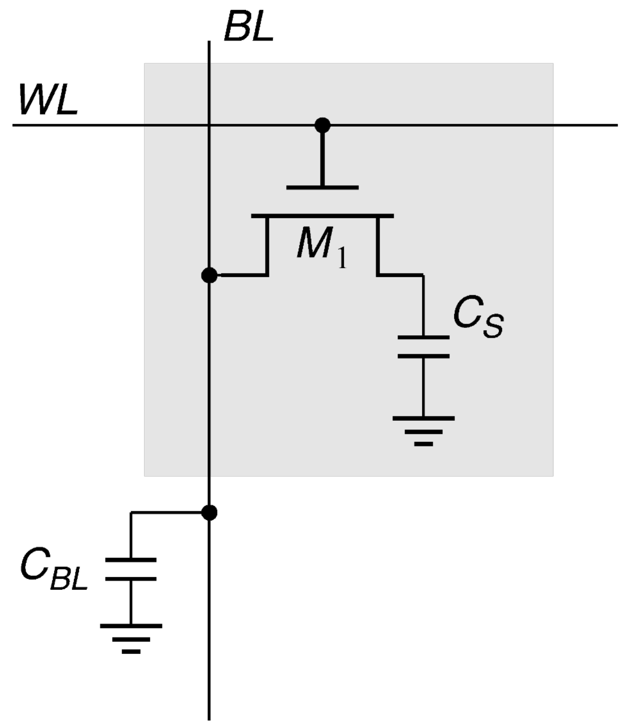


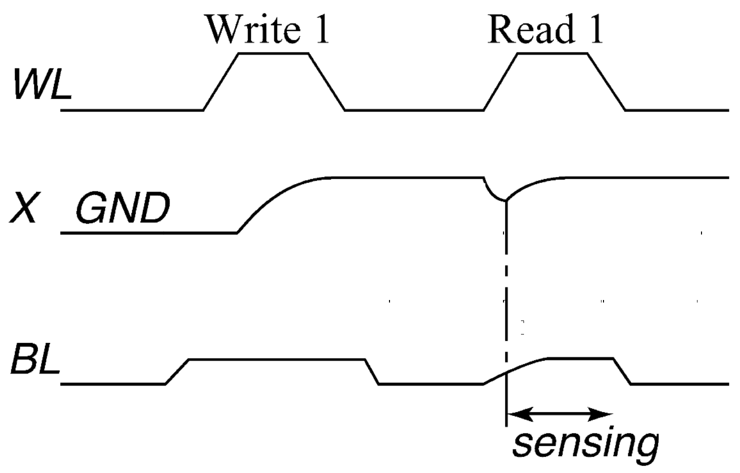


No constraints on device ratios
Reads are non-destructive
Value stored at node X when writing a “1” = V_{WWL}-V_{Tn}

Destructive read: after reading, the data bit changes with 100% probabilities

1T DRAM Cell



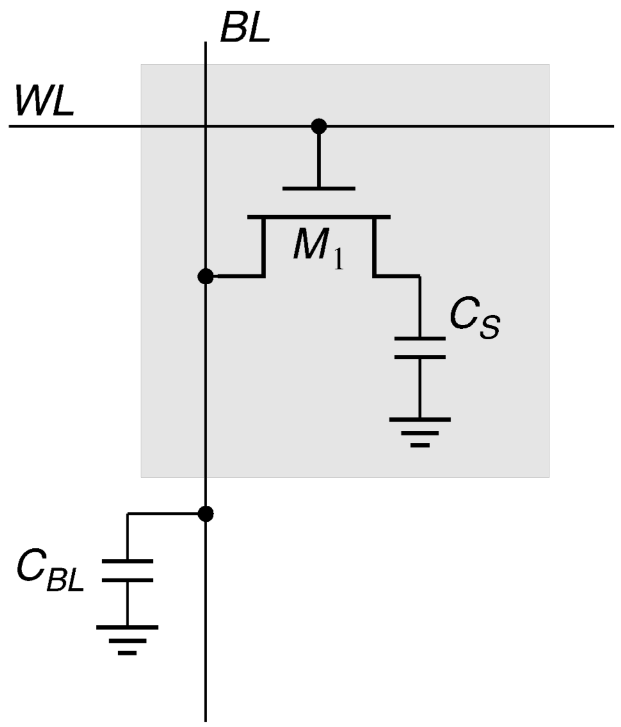


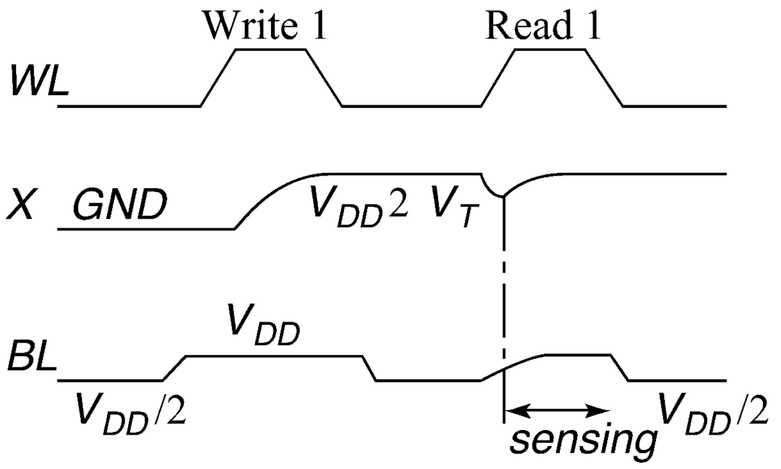
Write: C_S is charged or discharged by asserting WL and BL.
Read: Charge redistribution takes places between bit line and storage capacitance

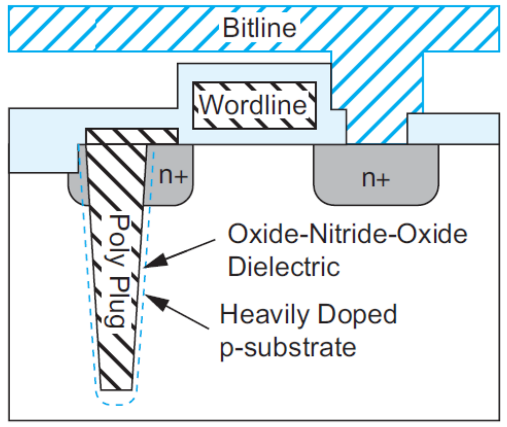
$$\Delta V = V_{BL} - V_{PRE} = V_{BIT} - V_{PRE} \frac{C_S}{C_S + C_{BL}}$$

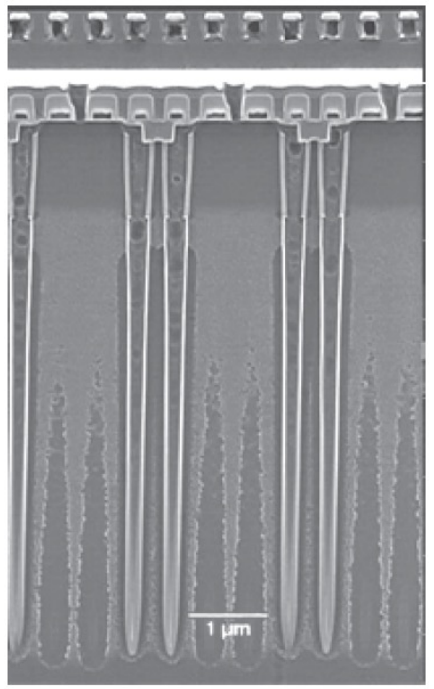
Voltage swing is small; typically around 250 mV.

1T DRAM Cell



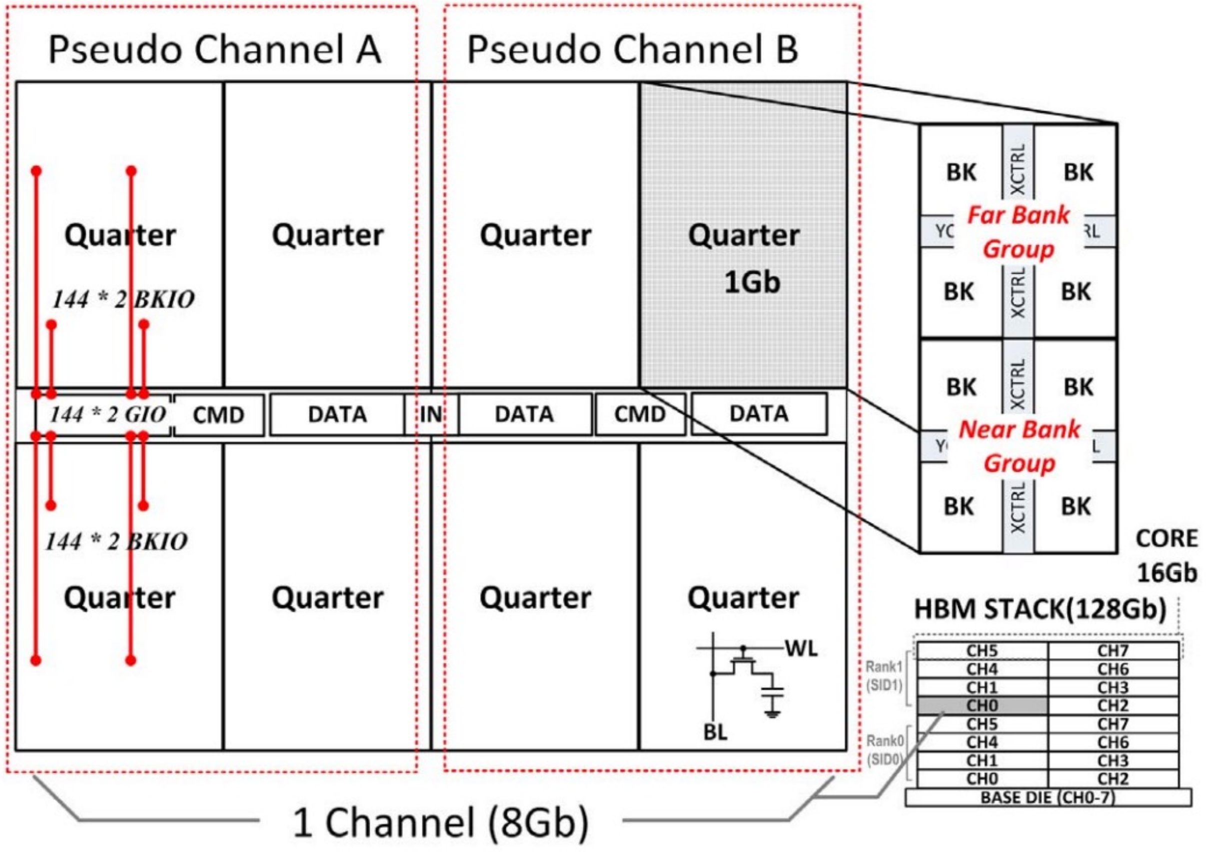


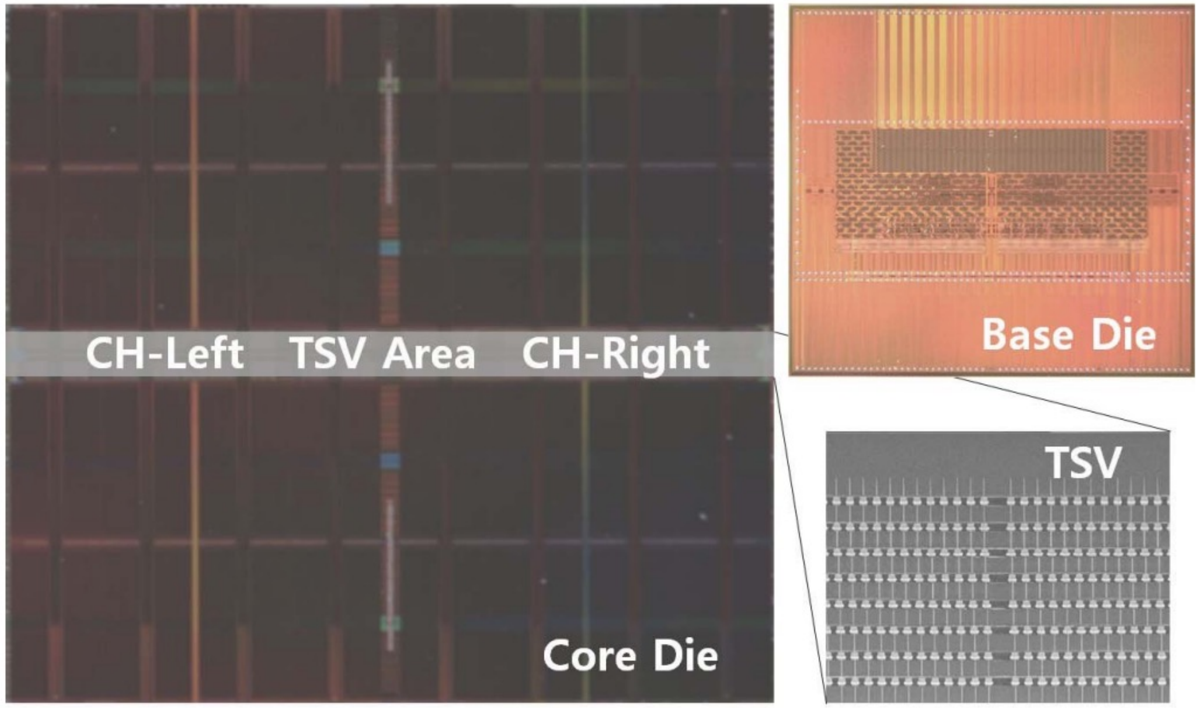




- Often offer 10~20 times higher density than SRAM
- Trench capacitor
  is often in specialized process

Better Density – 3D Integration





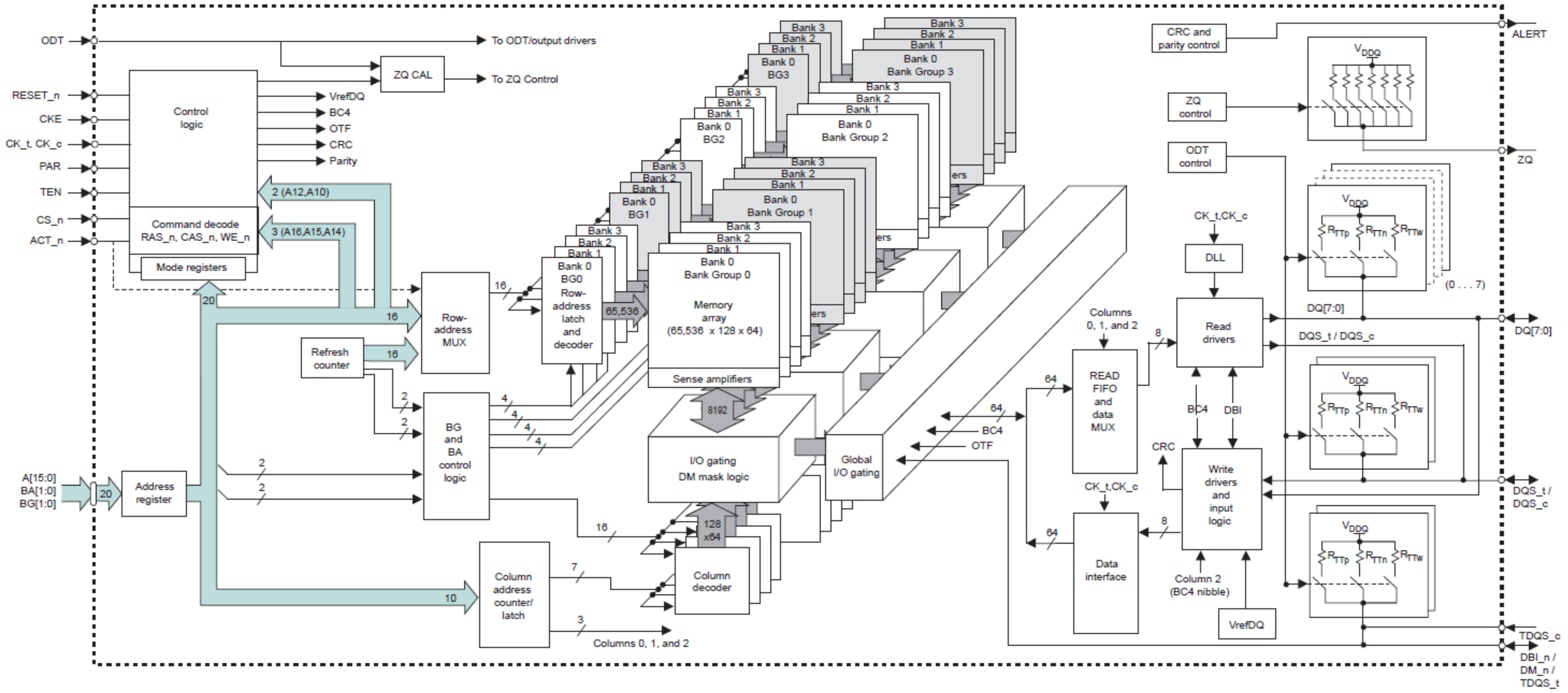
Lee, Dong Uk, et al. "22.3 A 128Gb 8-high 512GB/s HBM2E DRAM with a pseudo quarter bank structure, power dispersion and an instruction-based at-speed PMBIST." *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020.

DRAM vs. eDRAM

- eDRAM: embedded DRAM
- DRAM special process (thick oxide/high threshold) is too slow as logic platform
- eDRAM process incorporates compact-size of capacitors to be compatible with processes for logic (regular threshold transistors)

A Practical DRAM Product

Figure 3: 1 Gig x 8 Functional Block Diagram



Source: Micron

Modern memories

CPUs can also use **registers, caches** and **scratchpad memory**

Register Files & Cache

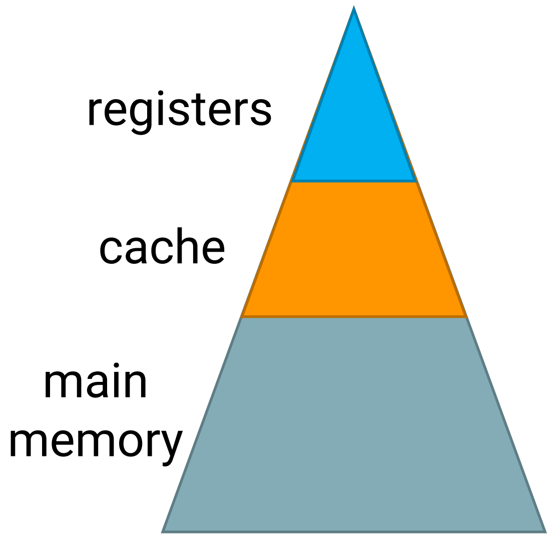
Register file: an array of registers
Cache: “transparent memory” to avoid excessive access to DRAM

Registers and Caches are often made of SRAM

```
int a , b , c;
a = 1;
b = 2;
C = a + b;
```

$\xrightarrow{\text{compiler}}$

where to store a, b and c?



- If registers are enough, they are all in registers
- If registers are not enough, they need to enter main memory (with an address)
- If “cache hit” happens, they will not go off-chip (large latency/low access bandwidth)

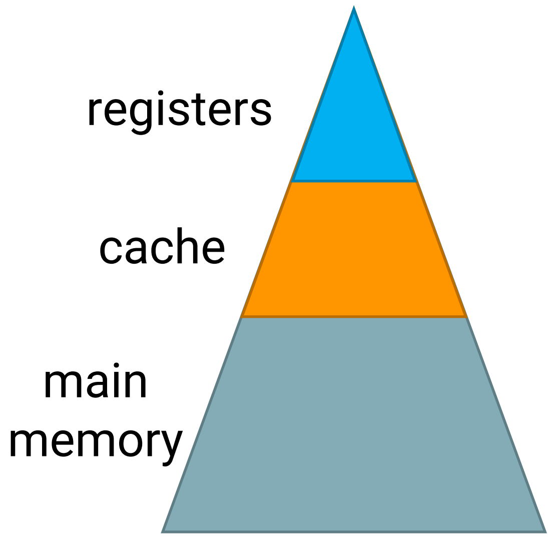
Register files

```
module picorv32_regs (
       input clk, wen,
       input [5:0] waddr,
       input [5:0] raddr1,
       input [5:0] raddr2,
       input [31:0] wdata,
       output [31:0] rdata1,
       output [31:0] rdata2
);
       reg [31:0] regs [0:30];

       always @(posedge clk)
              if (wen) regs[~waddr[4:0]] <= wdata;

       assign rdata1 = regs[~raddr1[4:0]];
       assign rdata2 = regs[~raddr2[4:0]];
endmodule
```

- Is this synthesizable? Yes!
- Do we use synthesis flow to generate memory? Yes!
- Sometimes we use this, often we use full-custom flow to design register files

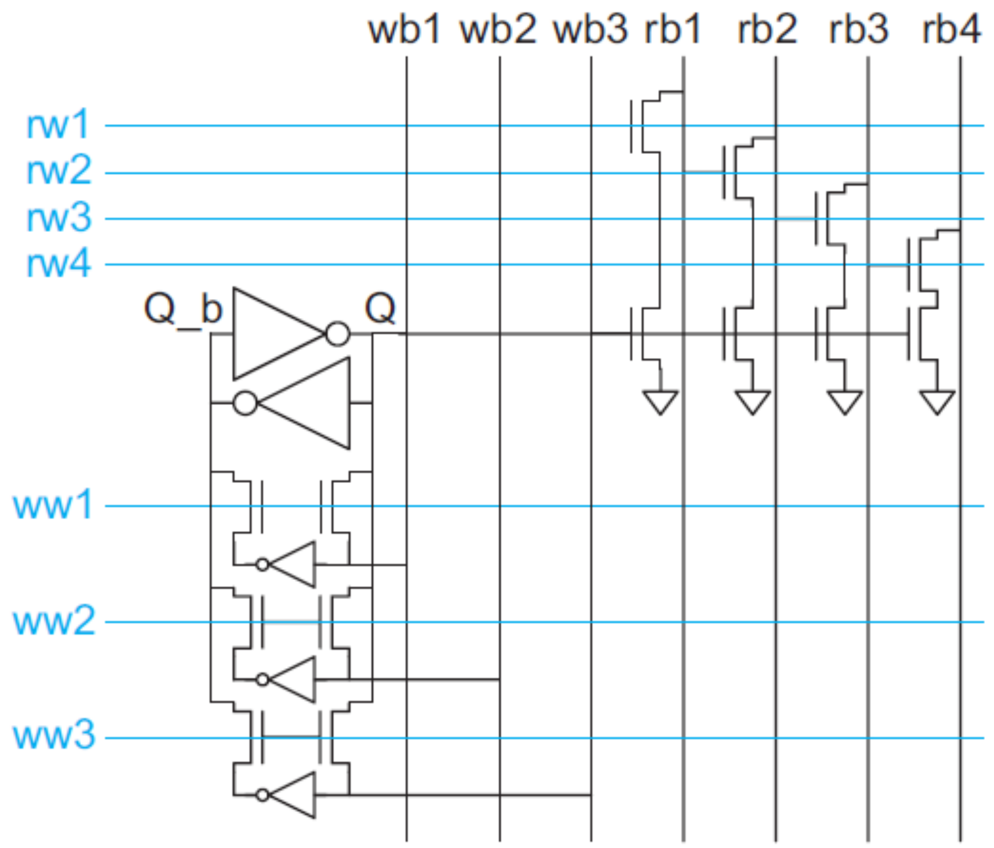


Source: picorv32 CPU

Register files

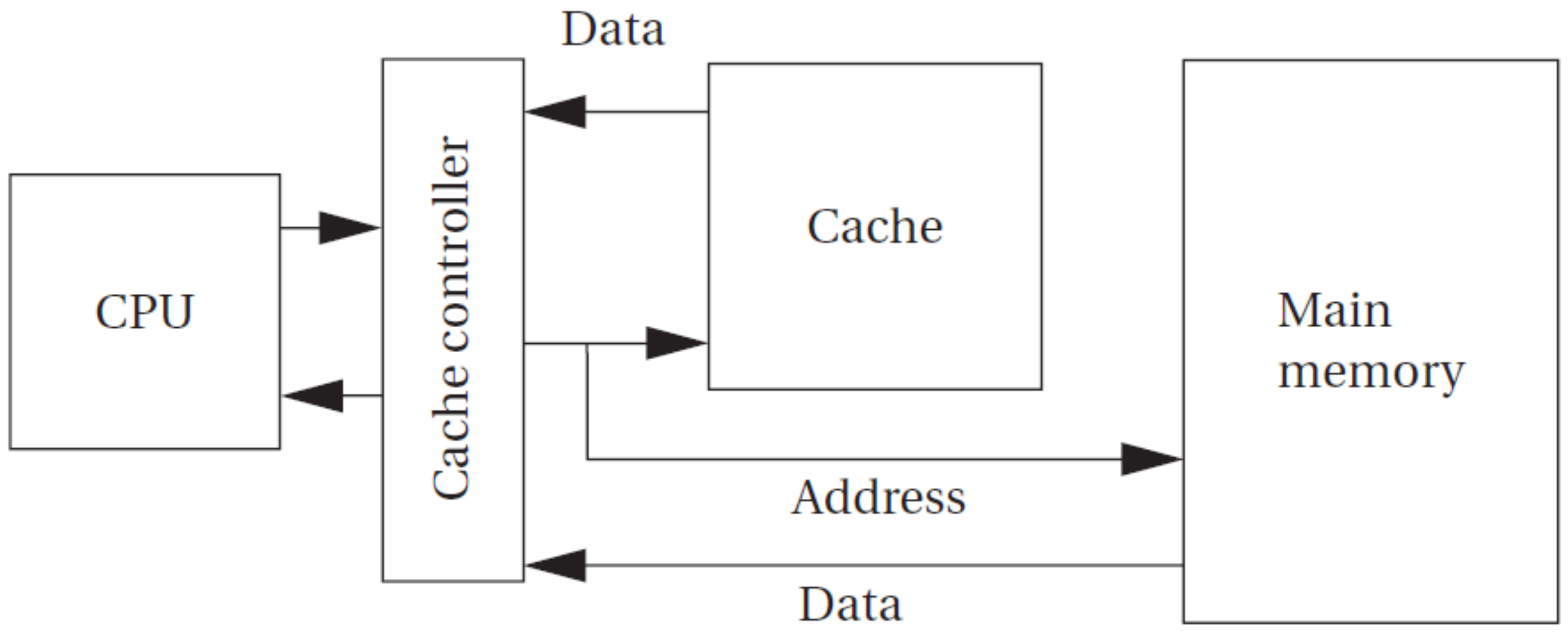
Made from SRAM?
Requirement: Multiport Read/Write

- How to use this cell to build large array of registers and even a register file?
- ww1, ww2, ww3 control need to be one-hot



Multiport SRAM cell

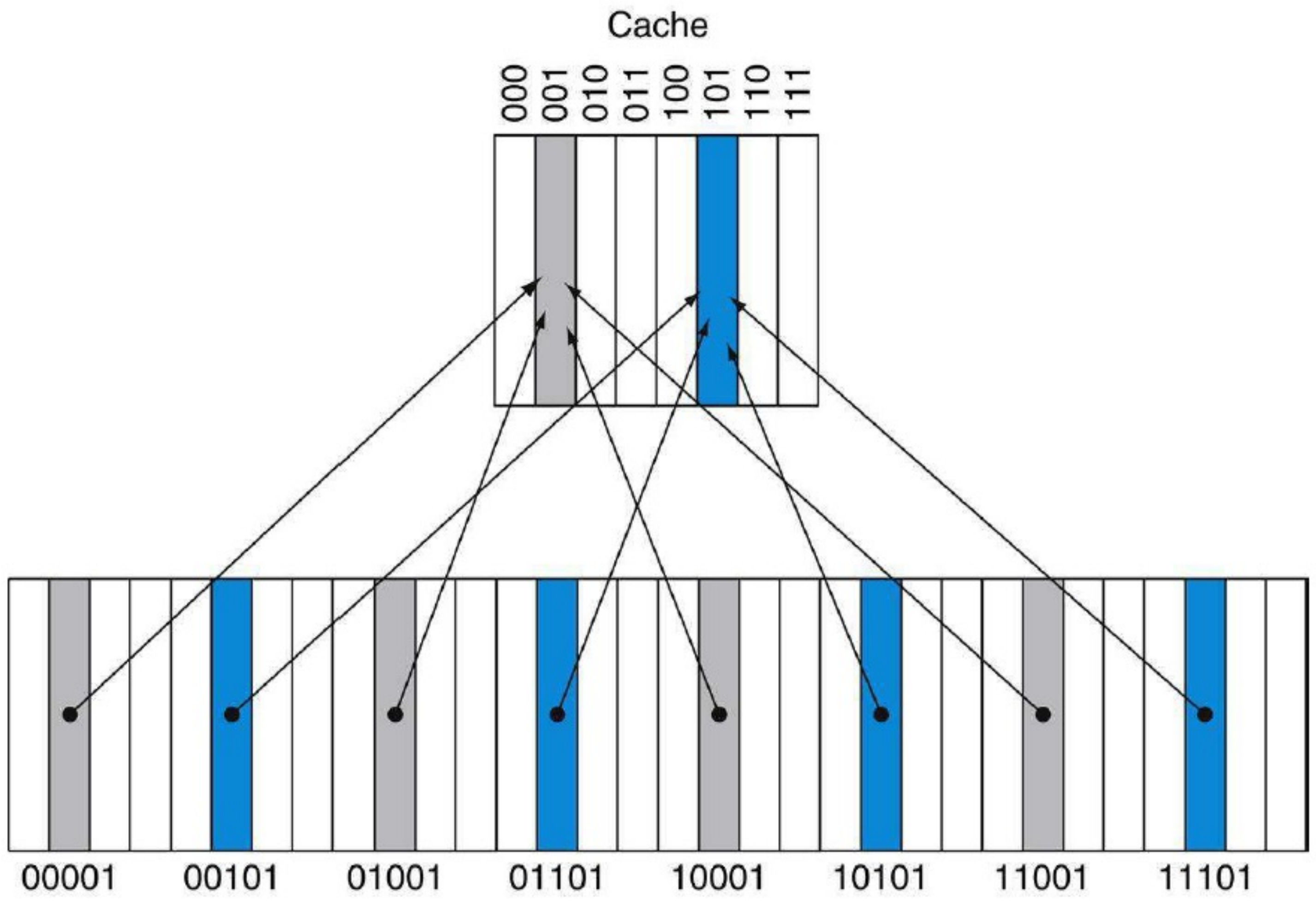
Cache



CPU keeps asking if the data is in the cache:

- If yes, it is a cache hit
- If no, it is a cache miss

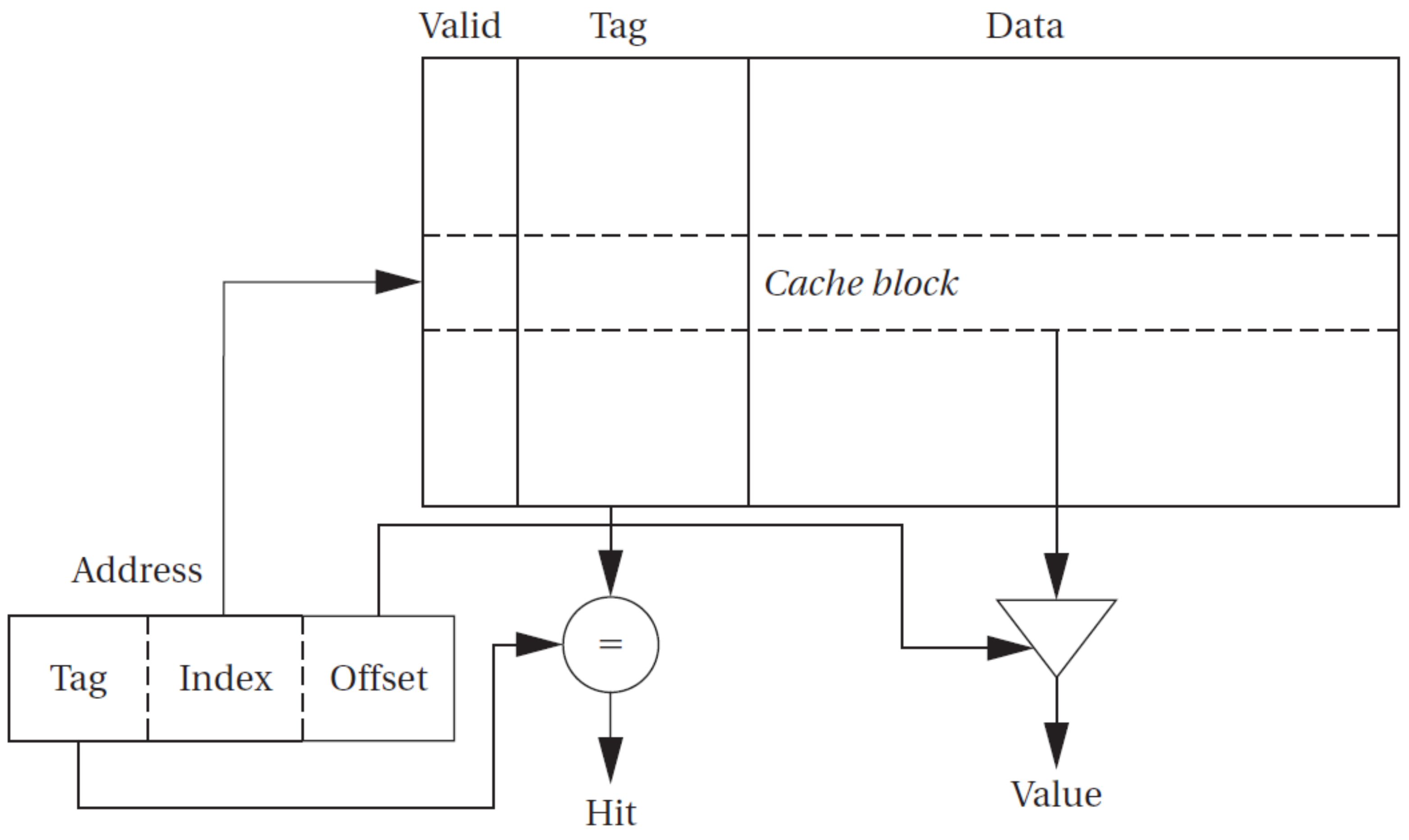
Direct Map



Tag: higher 2b address as the tag

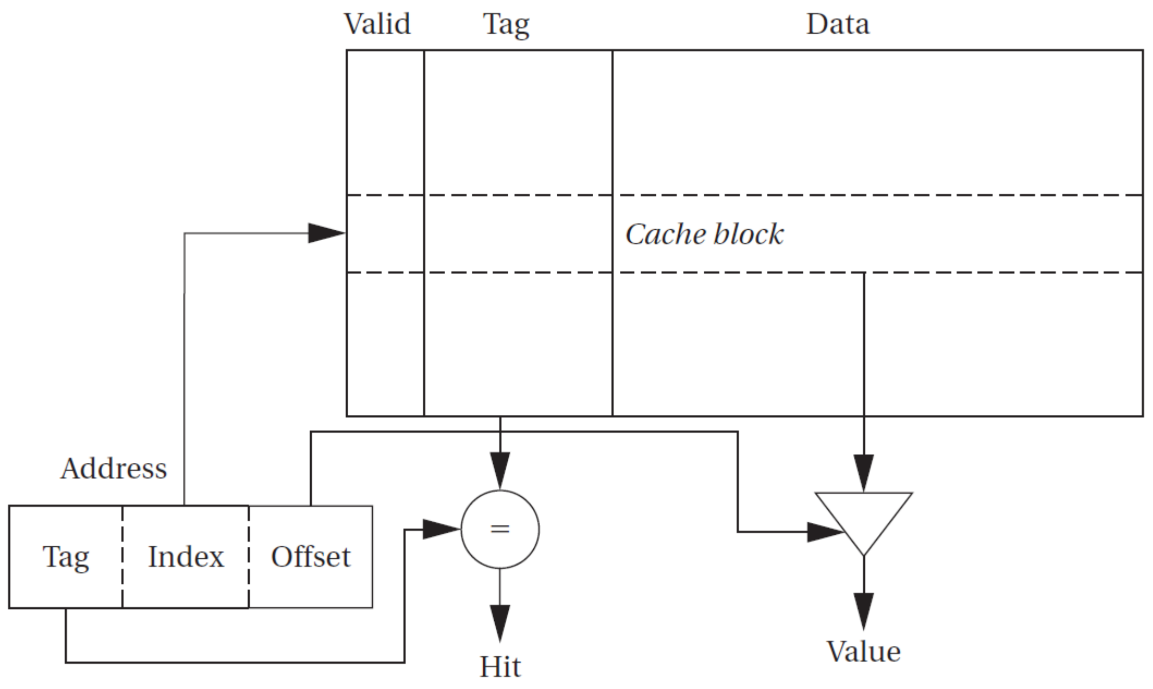
To clarify whether the requested word in memory is in the cache

Inside A Cache

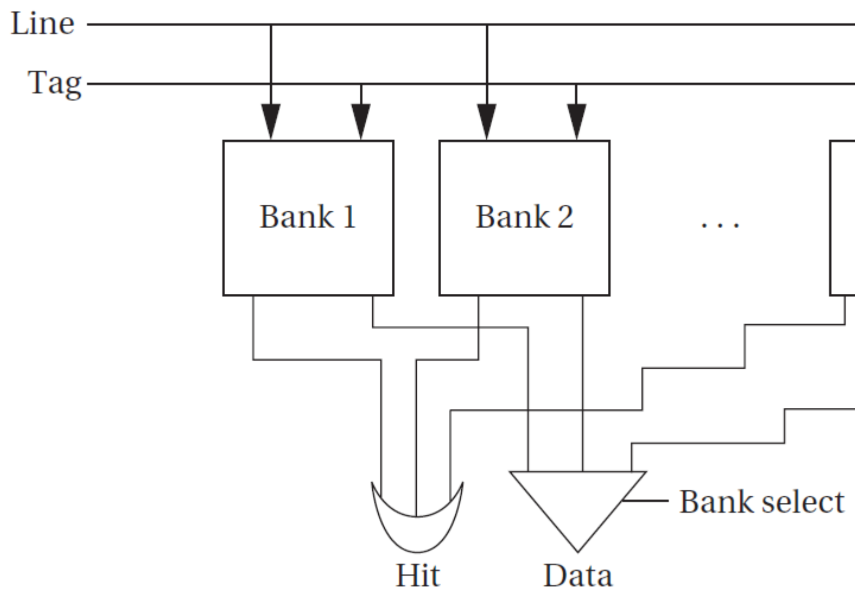


Content-addressable memory (CAM) / associate memory

Direct-associativity & Set-associativity

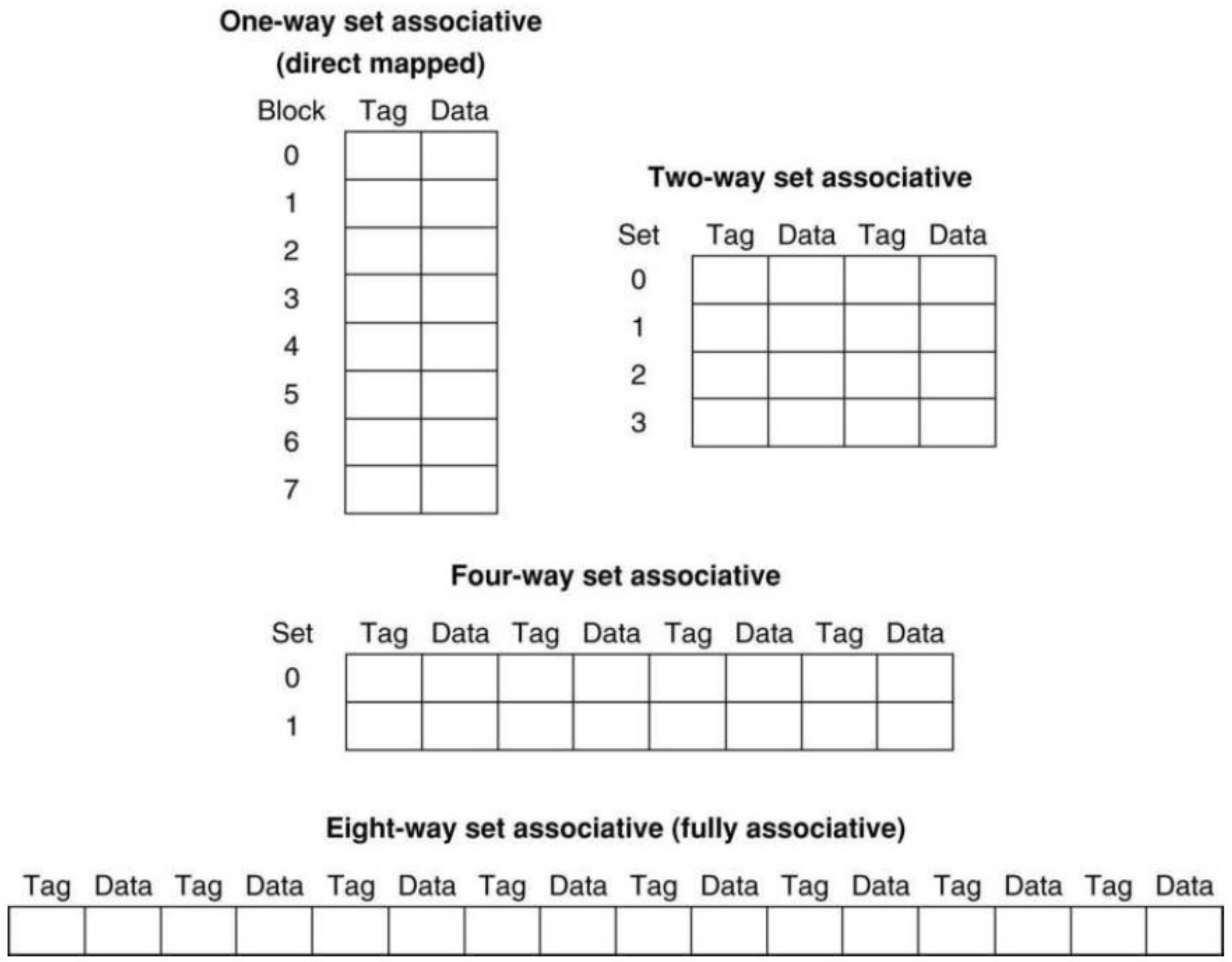


direct-associative



n-way set-associative

Example of direct- and set-associativity


One-way set associative
(direct mapped)
Block Tag Data
0
1
2
3
4
5
6
7
Two-way set associative
Set Tag Data Tag Data
0
1
2
3
Four-way set associative
Set Tag Data Tag Data Tag Data Tag Data
0
1
Eight-way set associative (fully associative)
Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data

Set: increase the cache hit probability

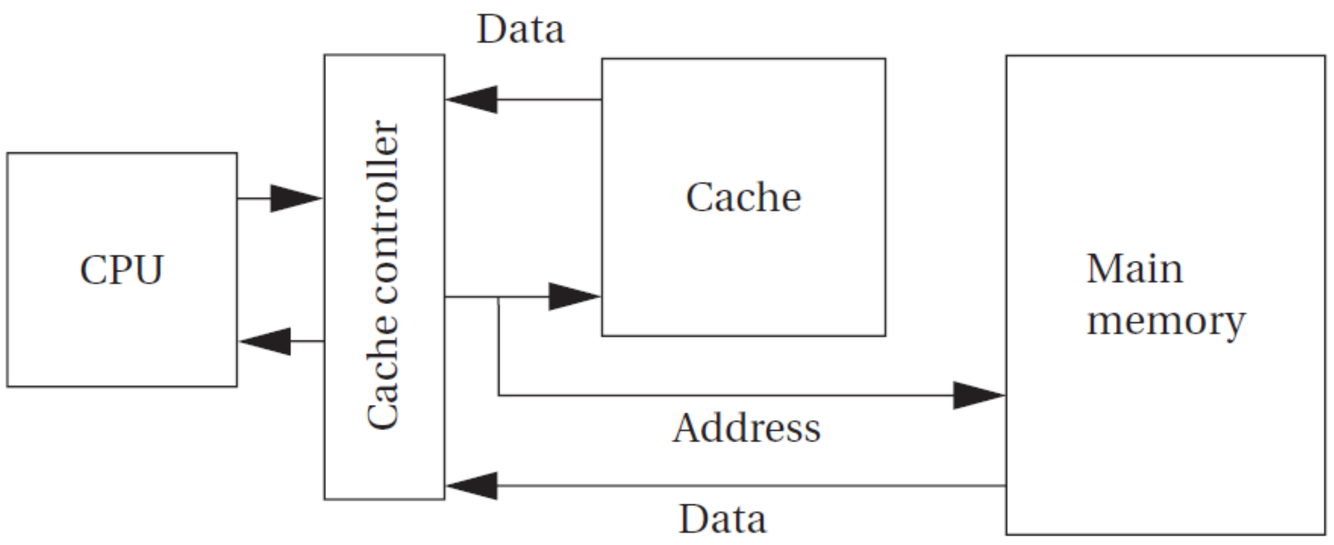
Assumption/Logic behind:
Cache miss leads to very long time of data movement

Challenges for Cache Design

Definition: Scratchpad Memory is just embedded "memory".

Embedded memory: memories that are on the chip, i.e. "integrated memory".

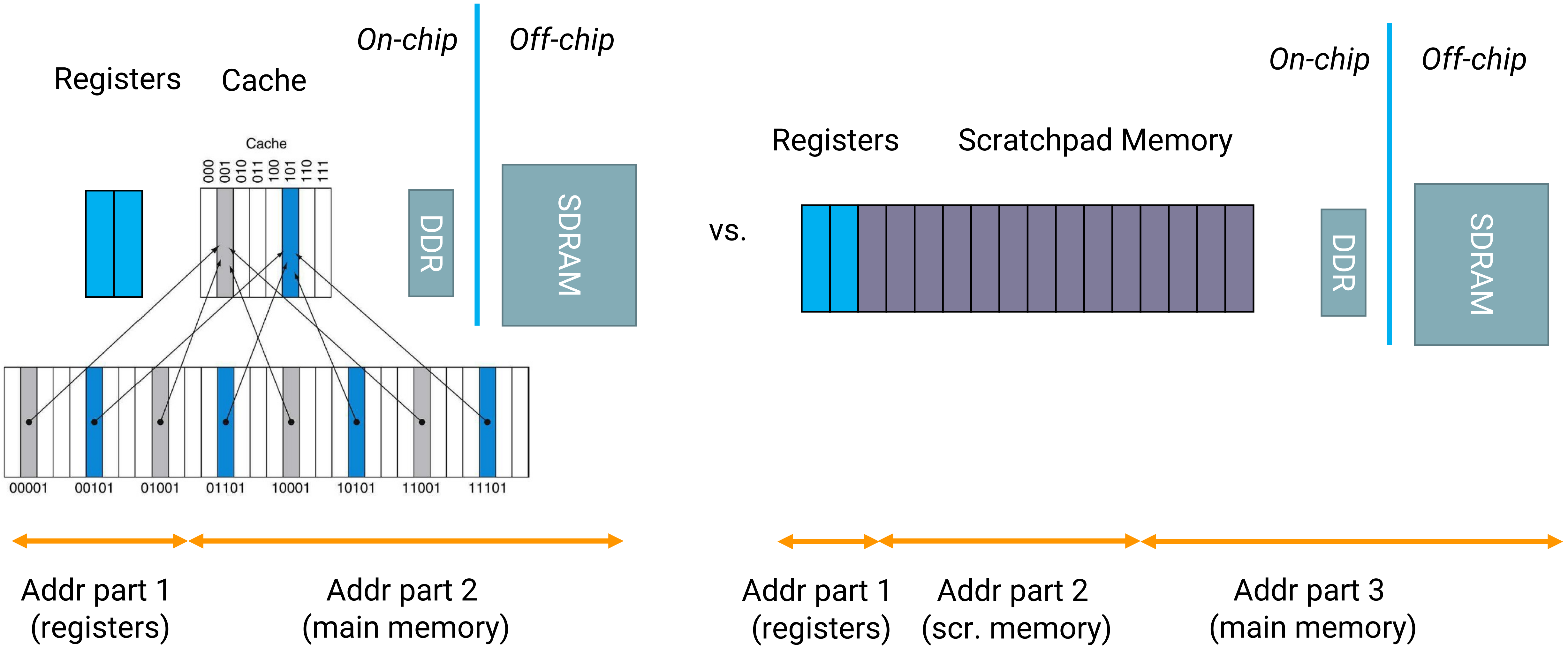
- Complex mechanism
- Large area overhead
- Stochastic characteristics
  - Cache miss penalty is too large!



Banakar, Rajeshwari, et al. "Scratchpad memory: A design alternative for cache on-chip memory in embedded systems." *Proceedings of the Tenth International Symposium on Hardware/Software Codesign. CODES 2002 (IEEE Cat. No. 02TH8627)*. IEEE, 2002.

Scratchpad Memory vs. Cache

Definition: Scratchpad Memory is just embedded “memory”.



Review for Memory

- Memories
  - Operations: Read, Write
  - Basic circuits: bitcell array, column/row decoders, sense amplifiers (SA)
  - Hierarchy:
    - registers, cache, scratchpad memory, main memory
  - Memory technologies:
    - Transistor-based ROMs, SRAM, DRAM, eNVM (embedded nonvolatile memory)
  - Other concepts:
    - cache miss, cache hit, address space