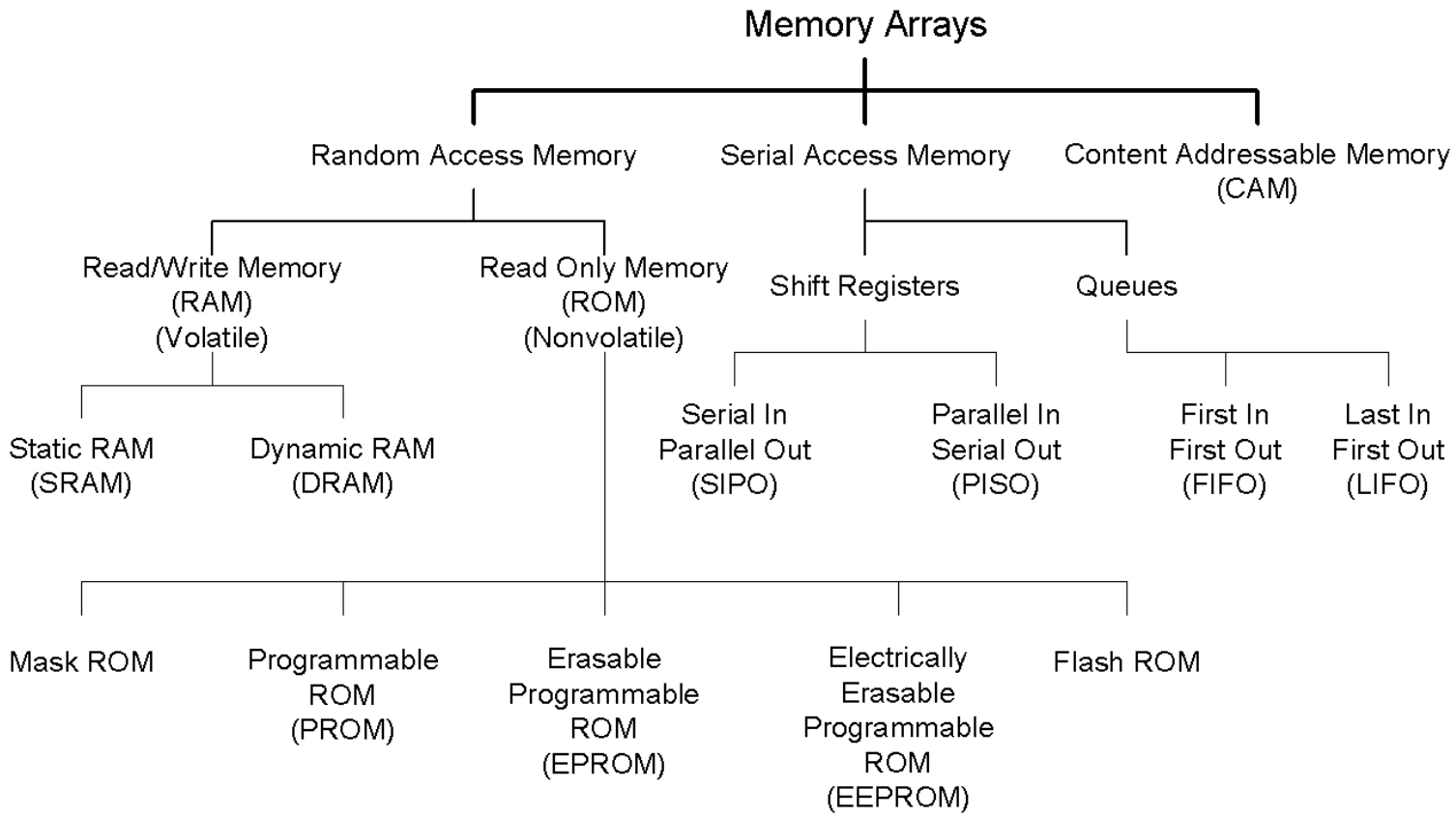
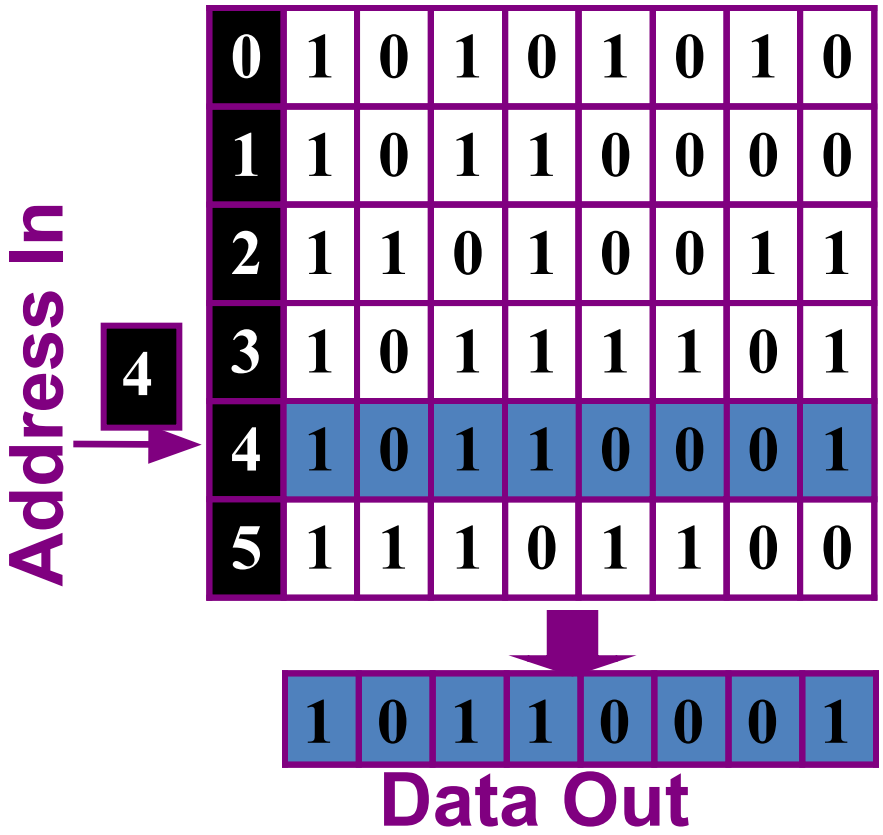
Memory


DRAM
CPU
CPU
5:30
9:00
72°
DRAM
CPU
DRAM
CPU
CPU
DRAM
CPU
DRAM
DRAM
CPU
DRAM
DRAM

Memory

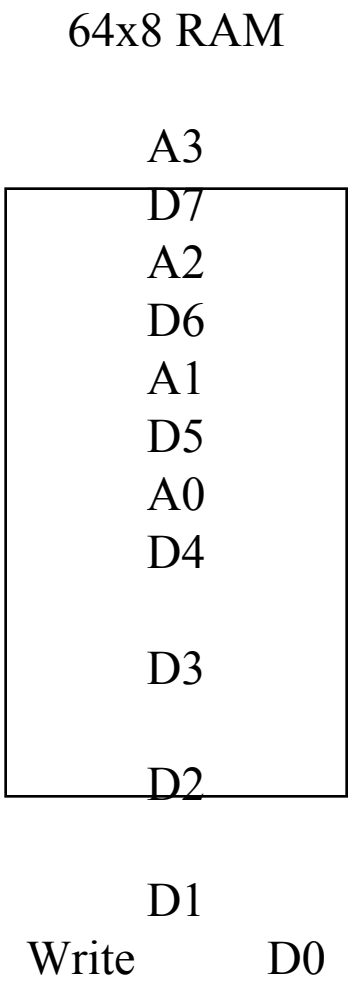

Memory Arrays
Random Access Memory
Serial Access Memory
Content Addressable Memory (CAM)
Read/Write Memory (RAM) (Volatile)
Read Only Memory (ROM) (Nonvolatile)
Shift Registers
Queues
Static RAM (SRAM)
Dynamic RAM (DRAM)
Serial In Parallel Out (SIPO)
Parallel In Serial Out (PISO)
First In First Out (FIFO)
Last In First Out (LIFO)
Mask ROM
Programmable ROM (PROM)
Erasable Programmable ROM (EPROM)
Electrically Erasable Programmable ROM (EEPROM)
Flash ROM

RAM

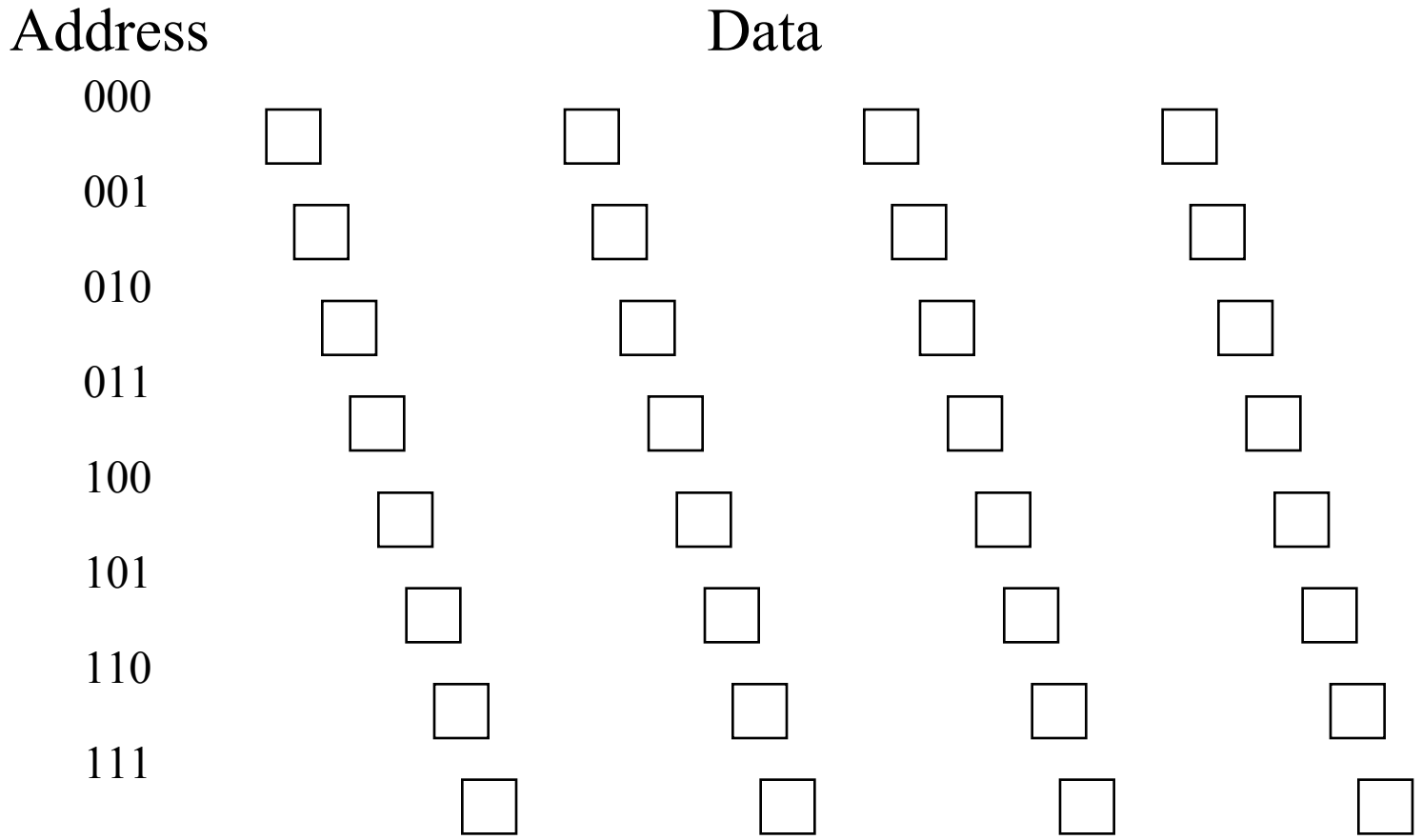

Address In
4
0 1 0 1 0 1 0 1 0
1 1 0 1 1 0 0 0 0
2 1 1 0 1 0 0 1 1
3 1 0 1 1 1 1 0 1
4 1 0 1 1 0 0 0 1
5 1 1 1 0 1 1 0 0
1 0 1 1 0 0 0 1
Data Out

Memory

Address	Data
000000	00111110
000001	01101011
000010	01011101
000011	01100011
000100	00111110
000101	00000000
000110	11111111
000111	01010101
001000	10101010
001001	00100001
001010	11011010

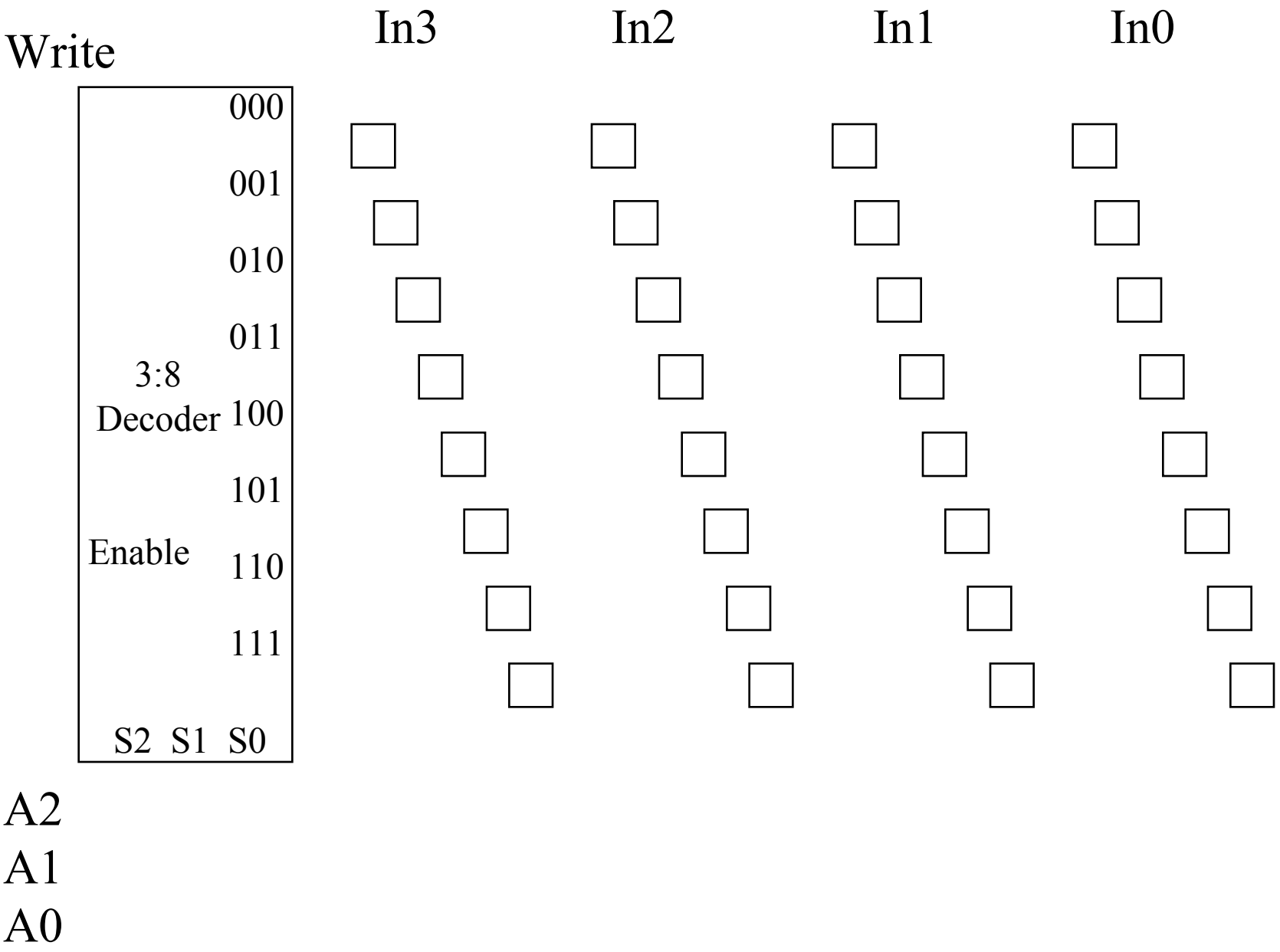


8x4 RAM



A2
A1
A0

8x4 RAM


In3
In2
In1
In0
Write
000
001
010
011
3:8
Decoder
100
101
Enable
110
111
S2 S1 S0
A2
A1
A0

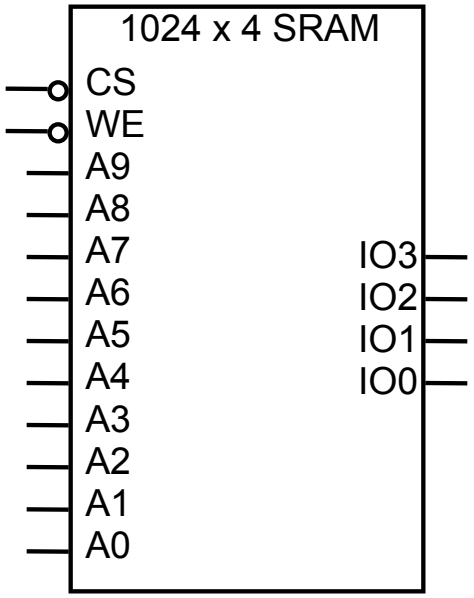
Static RAM Organization

Chip Select Line (active lo)

Write Enable Line (active lo)

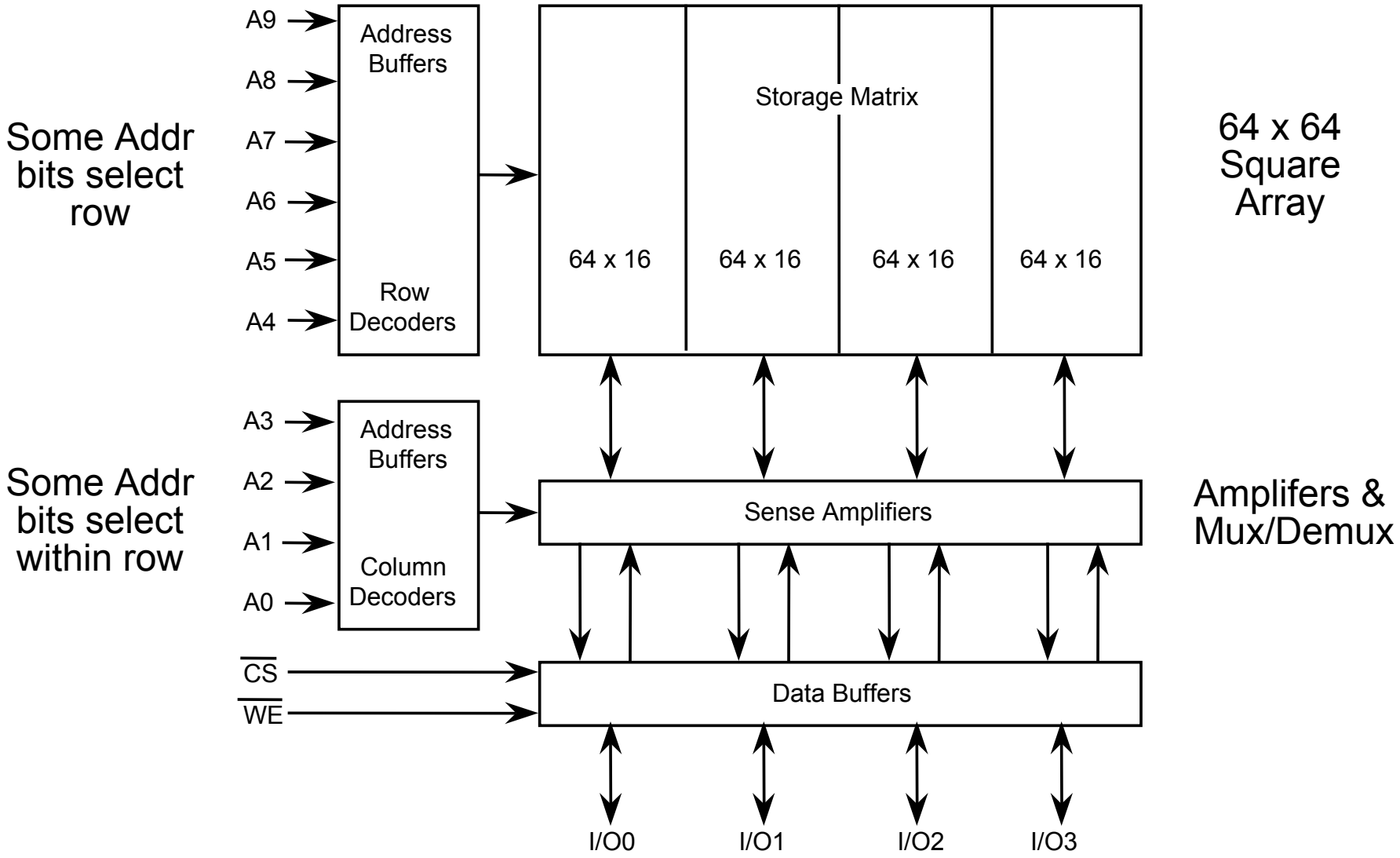
10 Address Lines

4 Bidirectional Data Lines

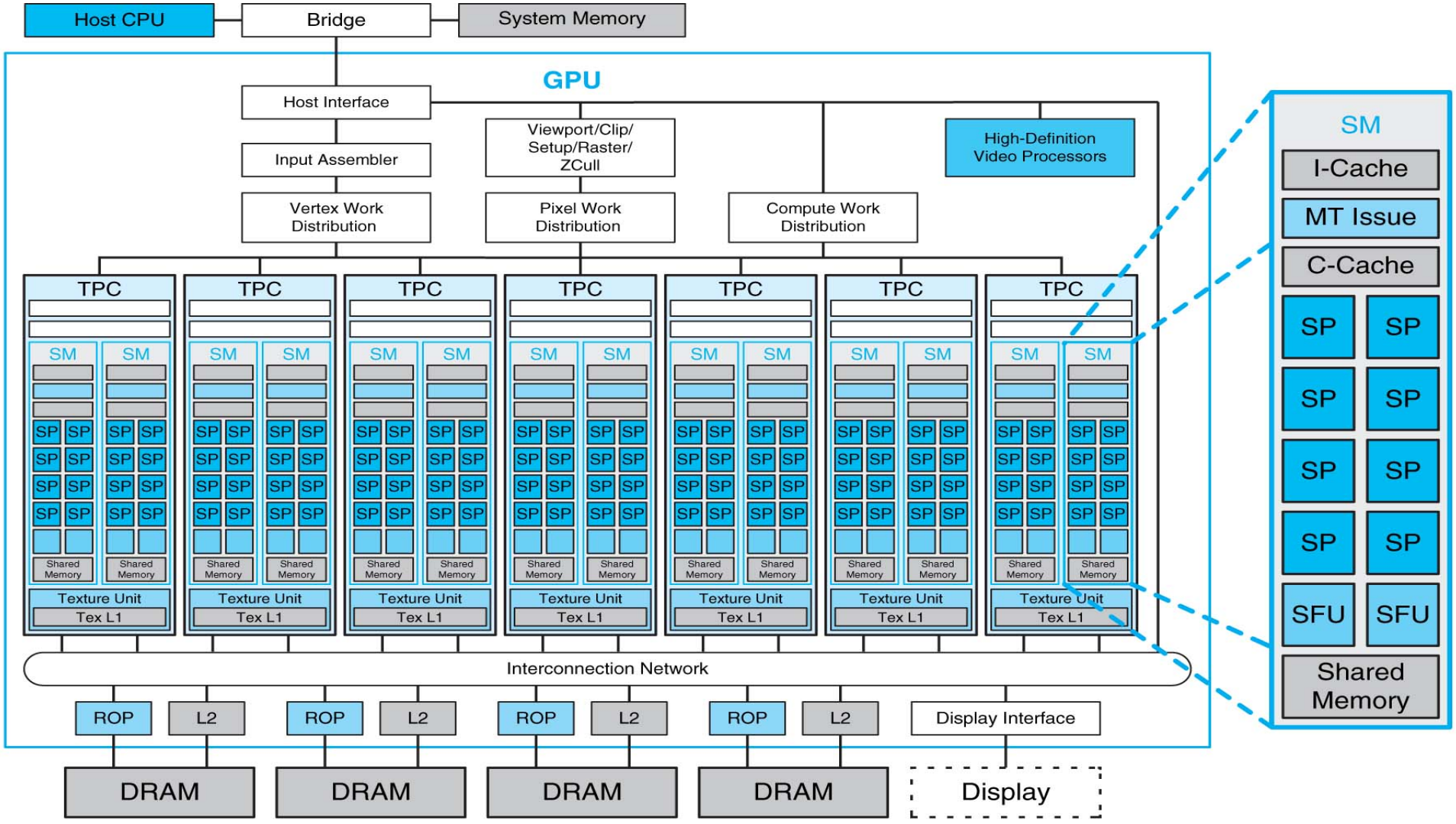


RAM Organization

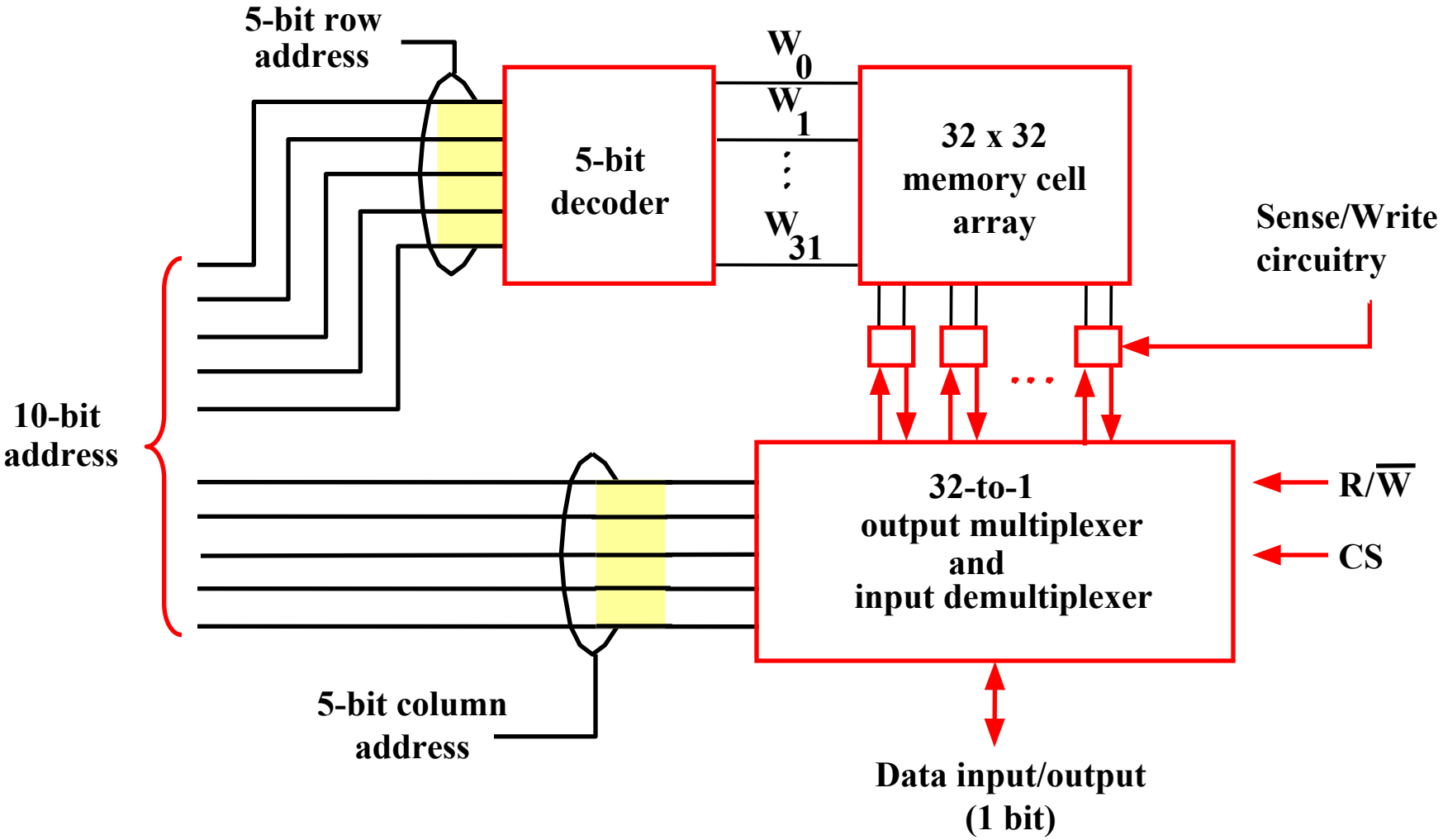
Long thin layouts are not the best organization for a RAM



Processor Array



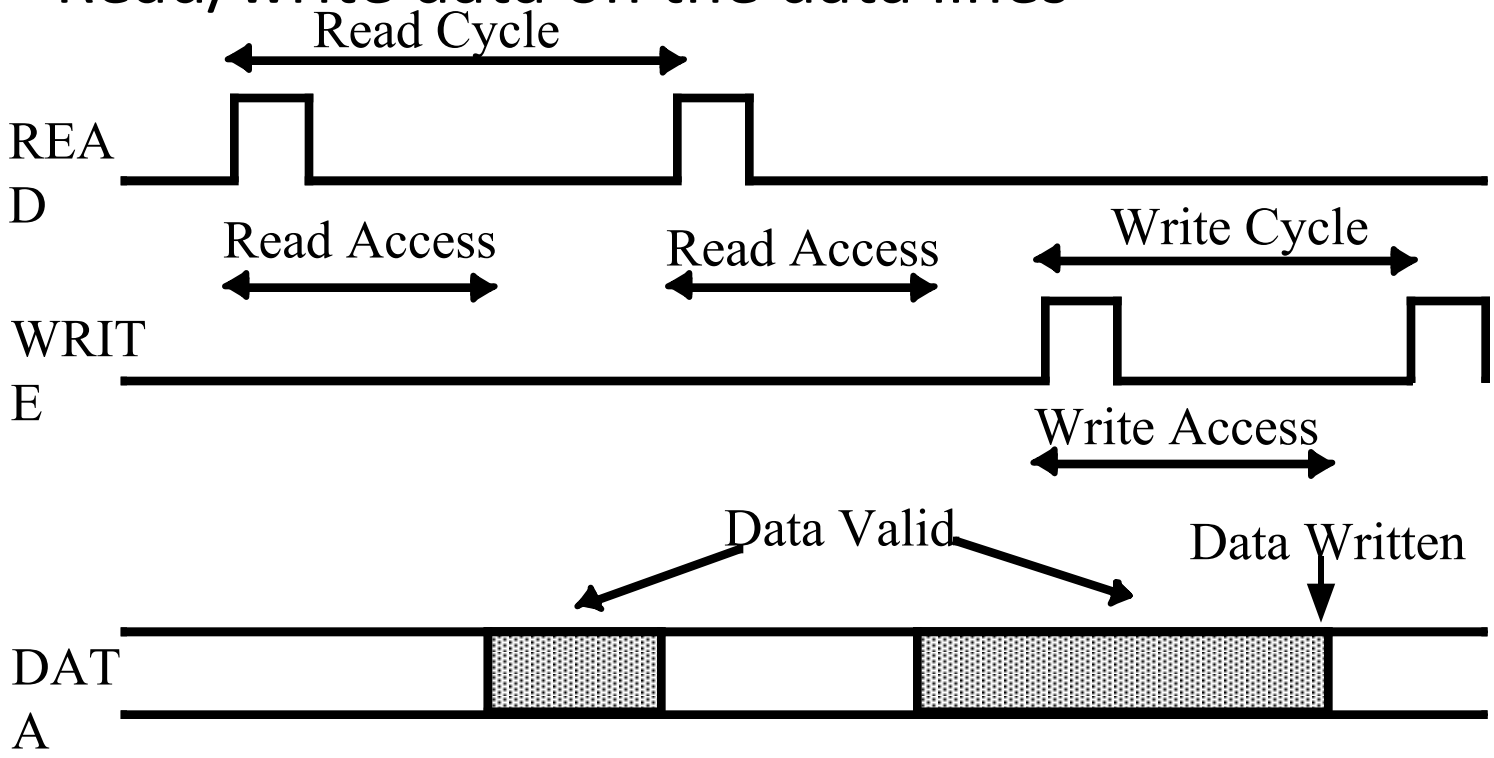
1Kx 1 bit RAM Organization



Organization of a 1K × 1 memory chip.

Memory Access Timing: the Big Picture

- Timing:
  - Send address on the address lines, wait for the word line to become stable
  - Read/write data on the data lines



Content of a memory

- Each word in memory is assigned an identification number, called an address, starting from 0 up to 2^k-1, where k is the number of address lines.
- The number of words in a memory with one of the letters K=2^{10}, M=2^{20}, or G=2^{30}.

  64K = 2^{16} 2M = 2^{21}

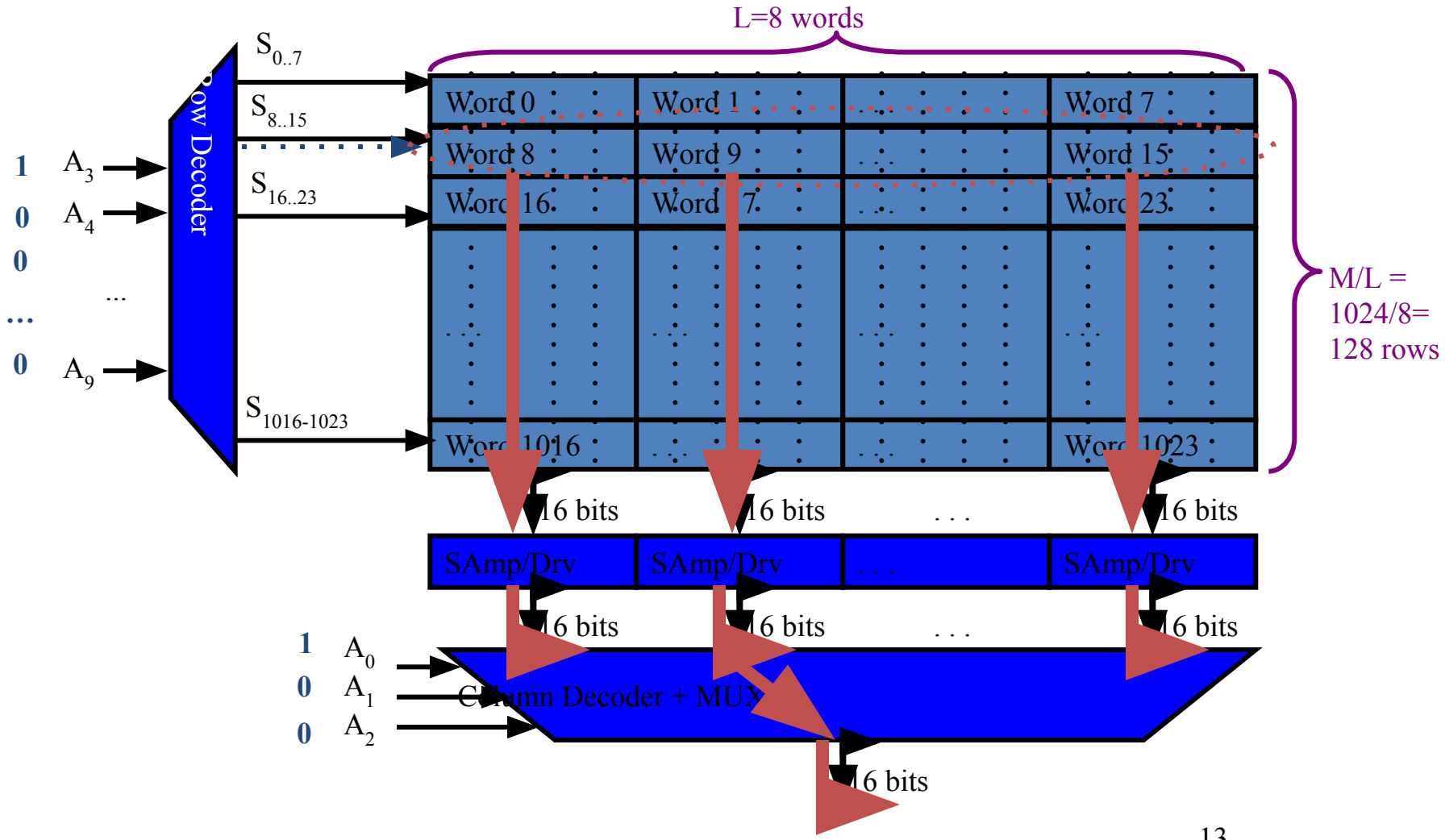
  4G = 2^{32}

Memory address		
Binary	decimal	Memory contest
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
⋮	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

1024x16 Memory Module

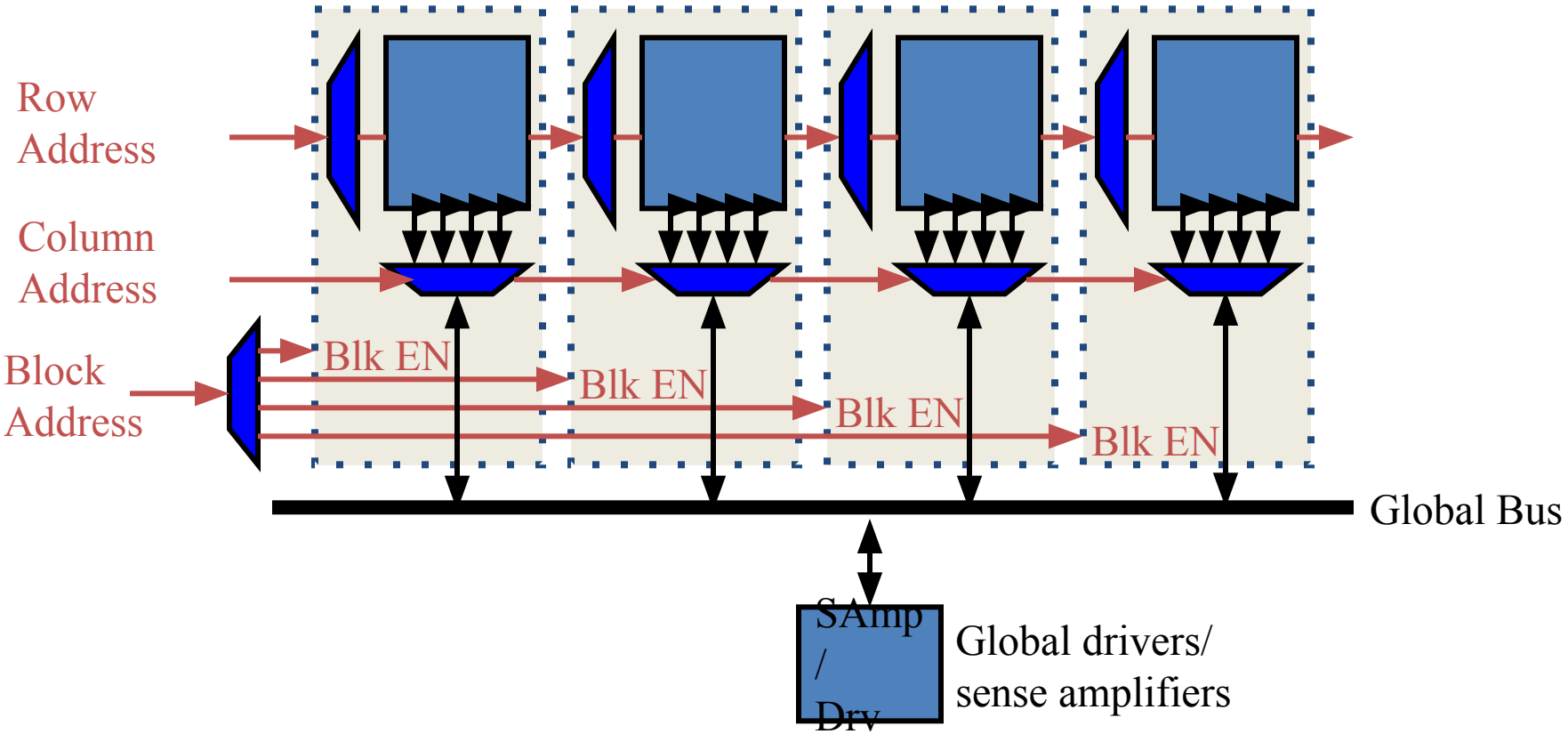
Memory Cell Array Access Example

- word=16-bit wide(N), row=8 words(L), address=10 bits (k)
- Accessing word 9= 0000001001_2

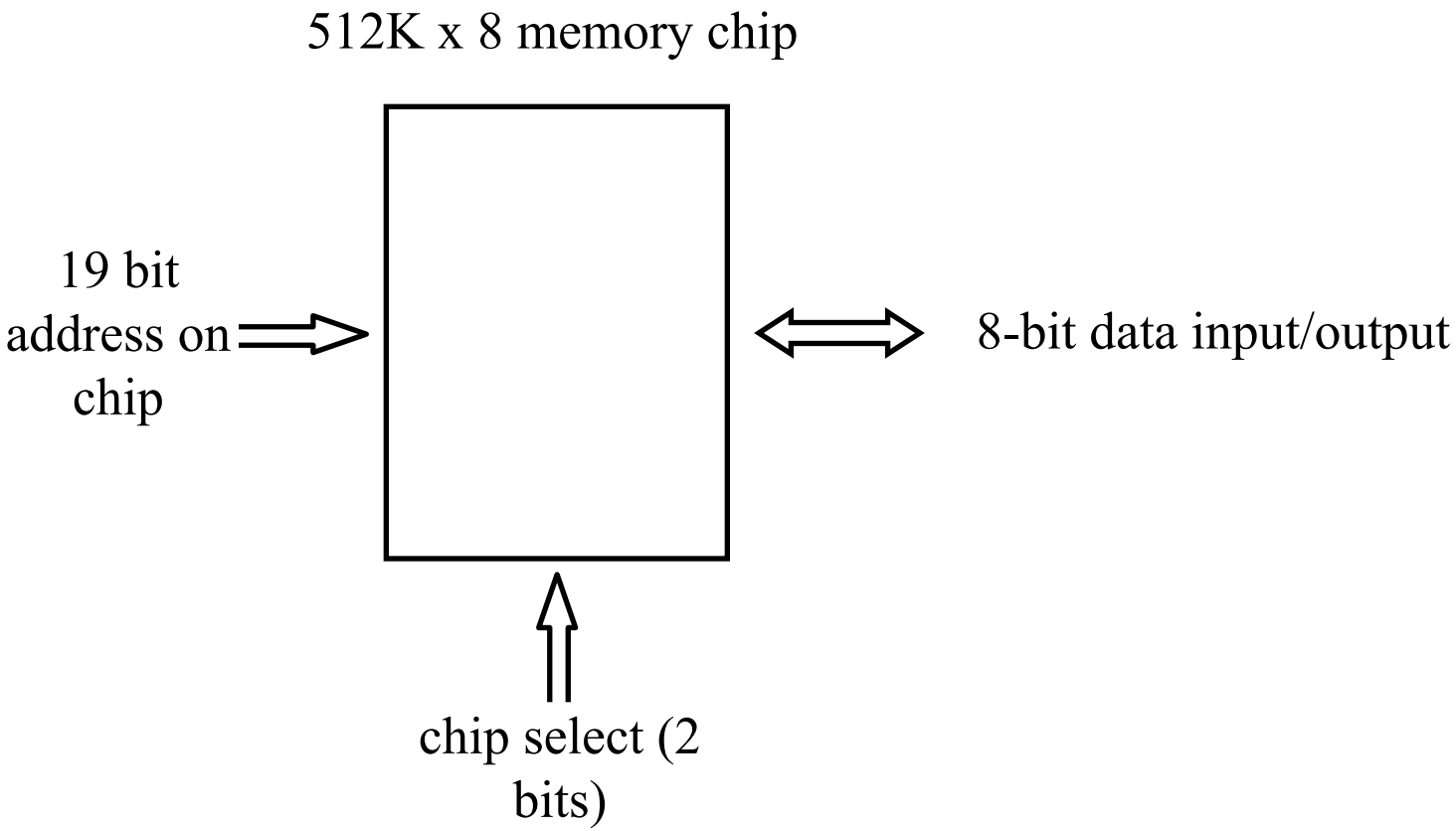


Memory Structures

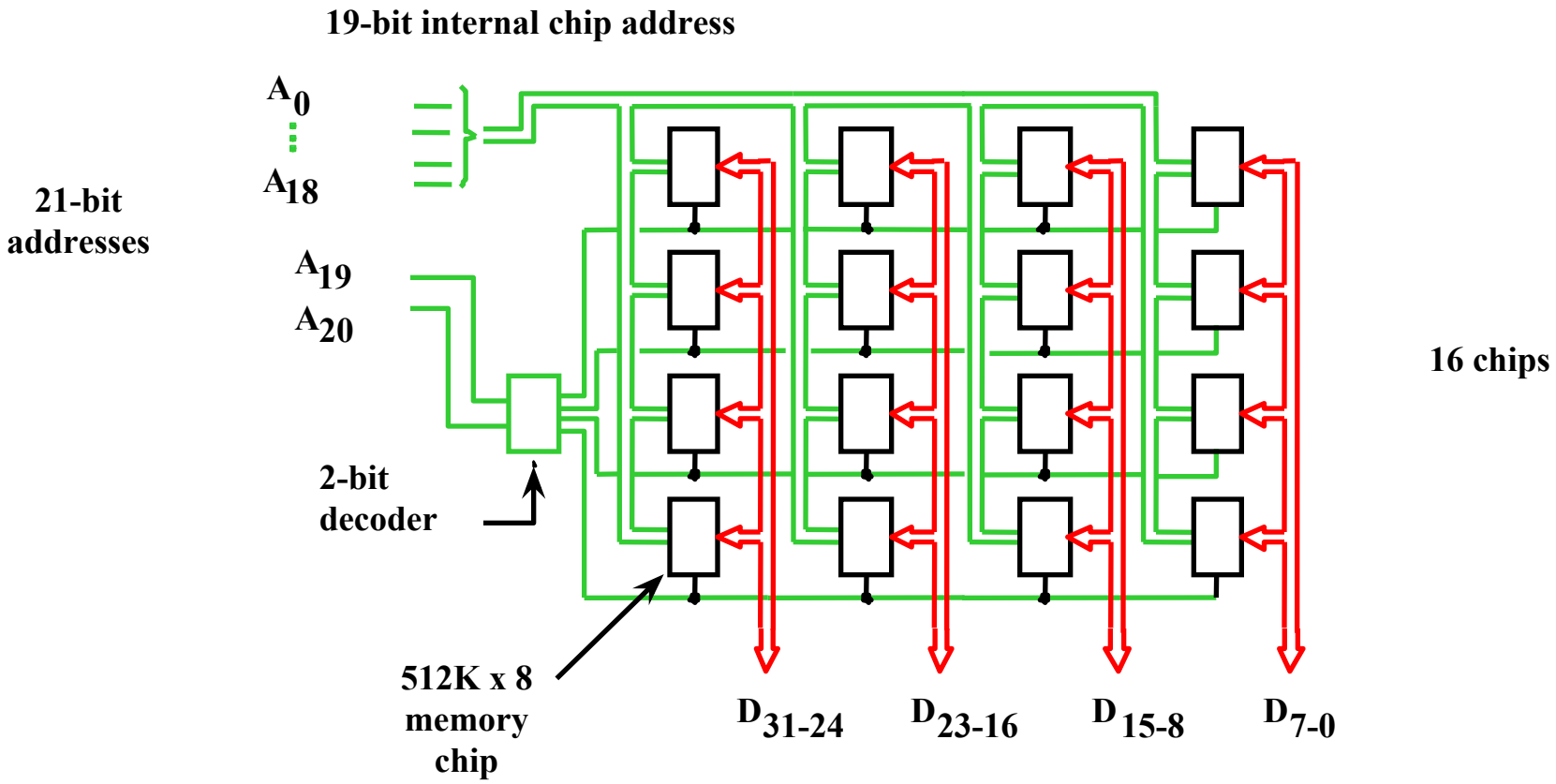
- Taking the idea one step further
  - Shorter wires within each block
  - Enable only one block addr decoder⍰ power savings



Larger Memories Using Multiple Chips

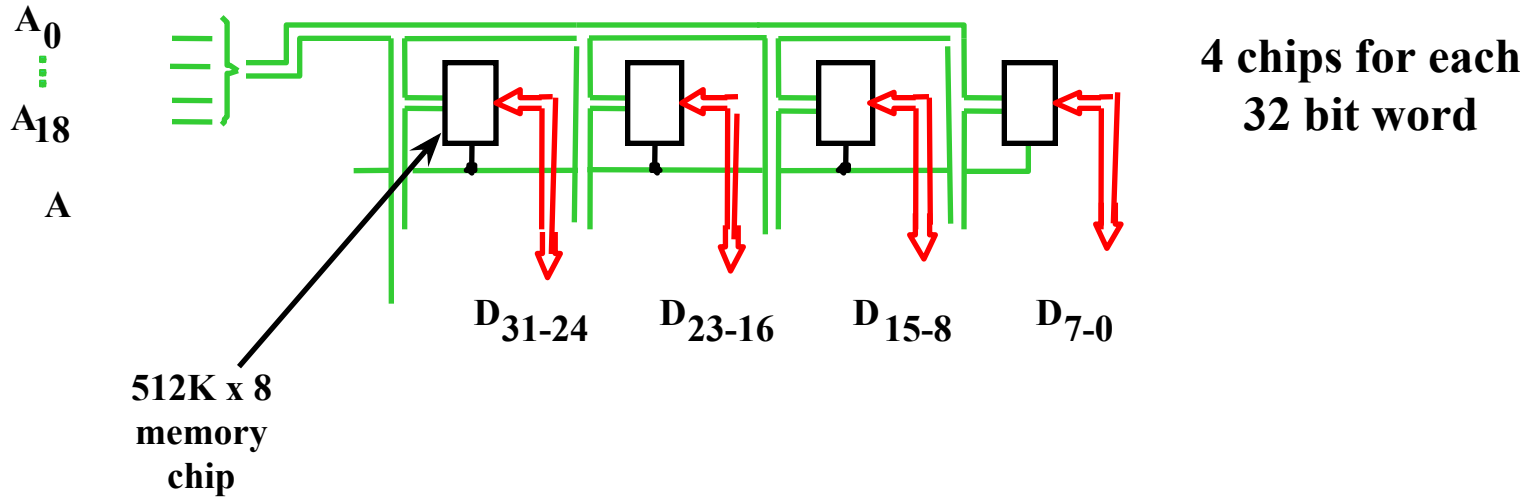


Question ? Design a 2Mx32 given: 512kx8

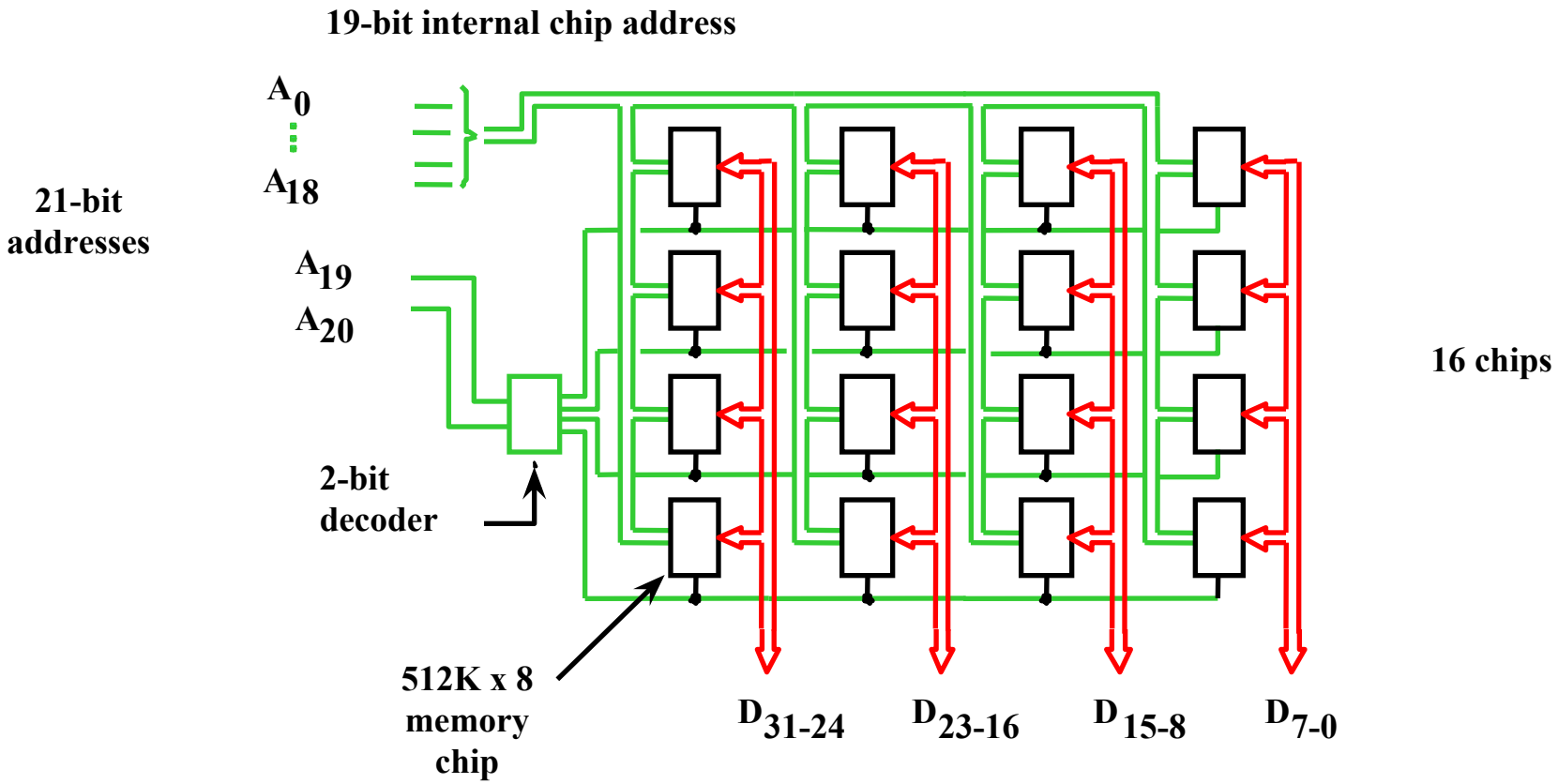


Organization of a 2M × 32 memory module using 512K × 8 static memory chips (16 chips).

19-bit internal chip address



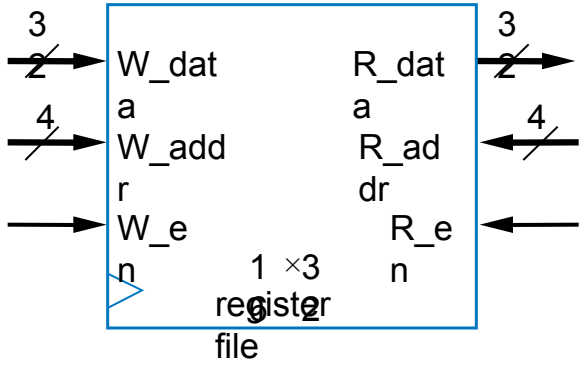
Organization of a 2M × 32 memory module using 512K × 8 static memory chips (16 chips).

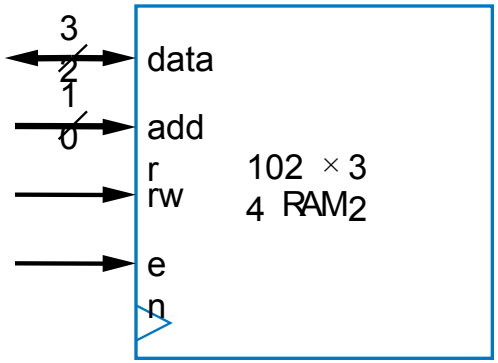


Organization of a 2M × 32 memory module using 512K × 8 static memory chips (16 chips).

Random Access Memory (RAM)

- RAM – Readable and writable memory
  - "Random access memory"
    - Strange name—Created several decades ago to contrast with sequentially-accessed storage like tape drives
  - Logically same as register file—Memory with address inputs, data inputs/outputs, and control
    - RAM usually one port; RF usually two or more
  - RAM vs. RF
    - RAM typically larger than *about* 512 or 1024 words
    - RAM typically stores bits using a bit storage approach that is more efficient than a flip-flop
    - RAM typically implemented on a chip in a square rather than rectangular shape—keeps longest wires (hence delay) short

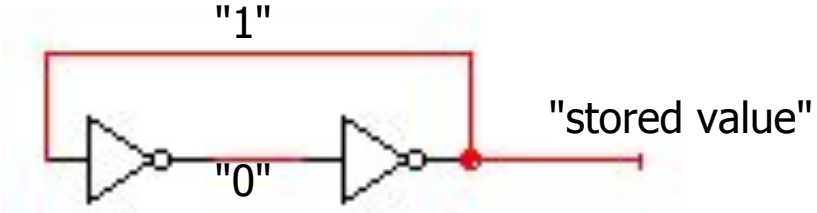




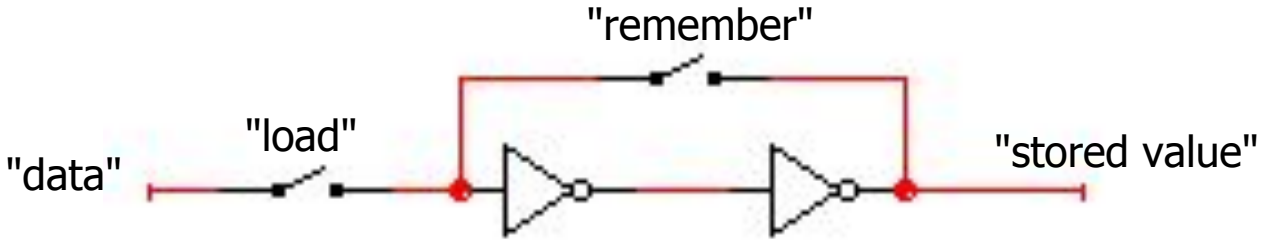
RAM block symbol

Simplest circuits with feedback

- Two inverters form a static memory cell
  - will hold value as long as it has power applied

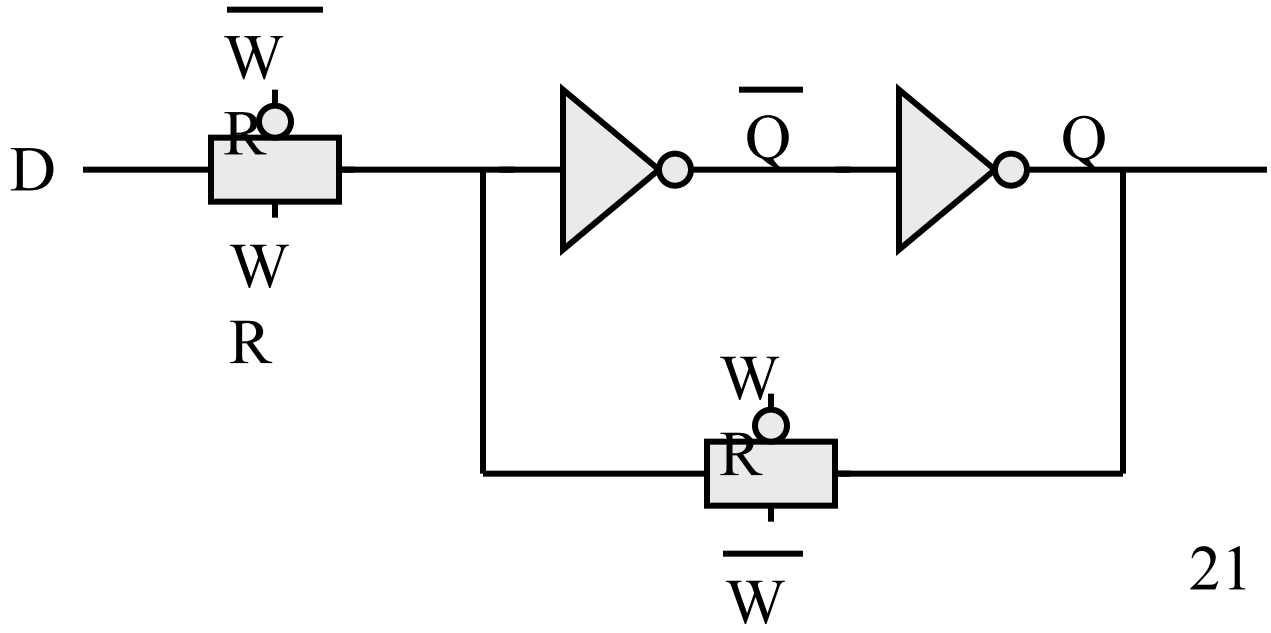


- How to get a new value into the memory cell?
  - selectively break feedback path
  - load new value into cell

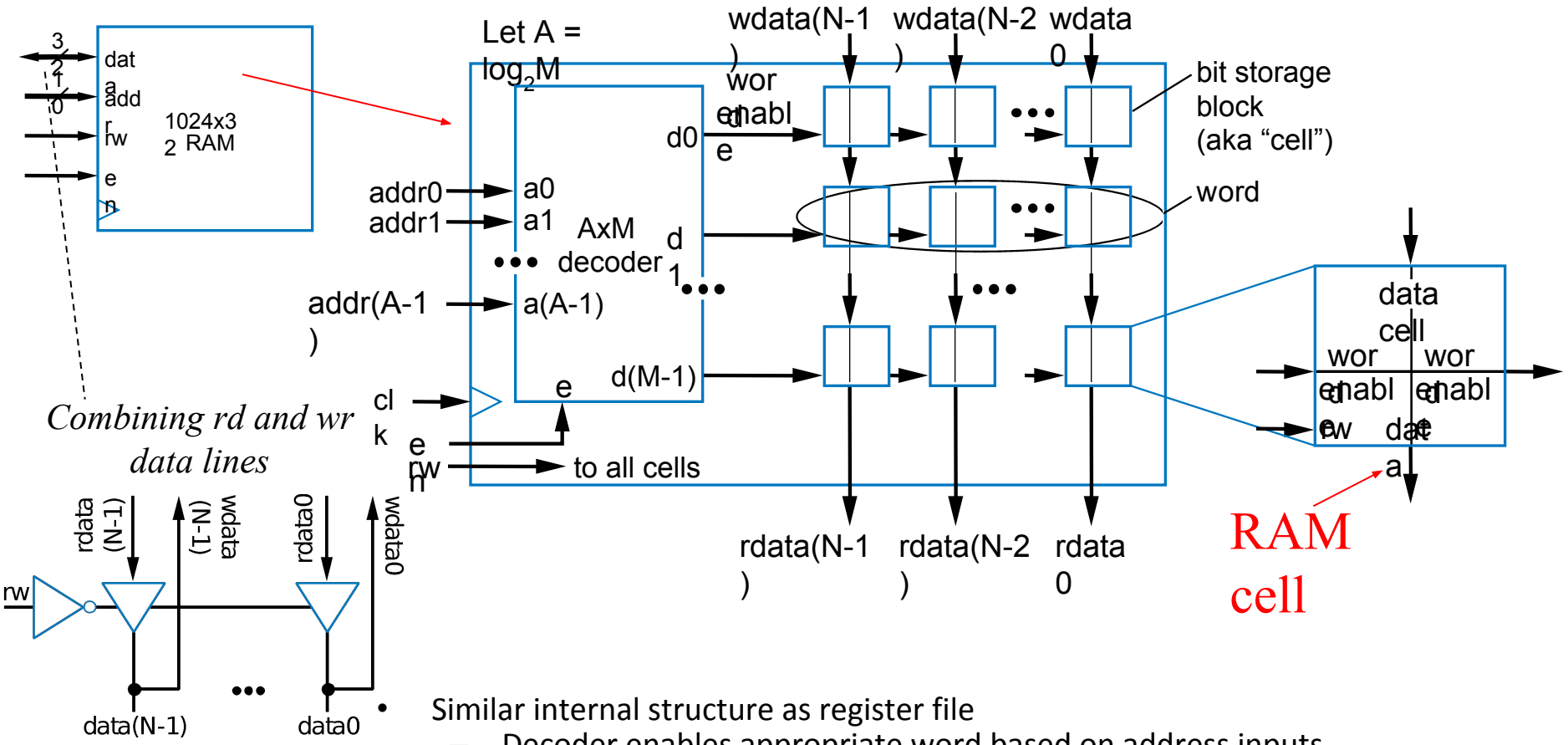


Implementing Registers in CMOS

- Uses transmission gate
  - When “WR” asserted, “write” operation will take place
  - Stack D latch structures to get n-bit register



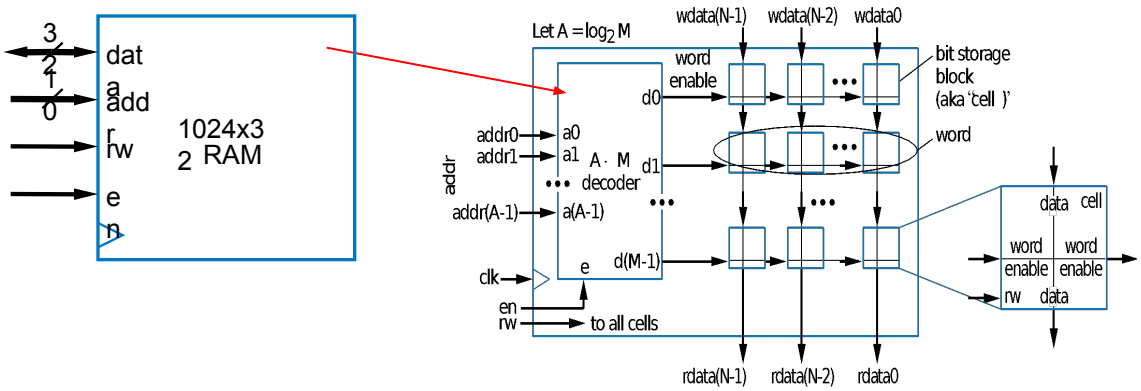
RAM Internal Structure

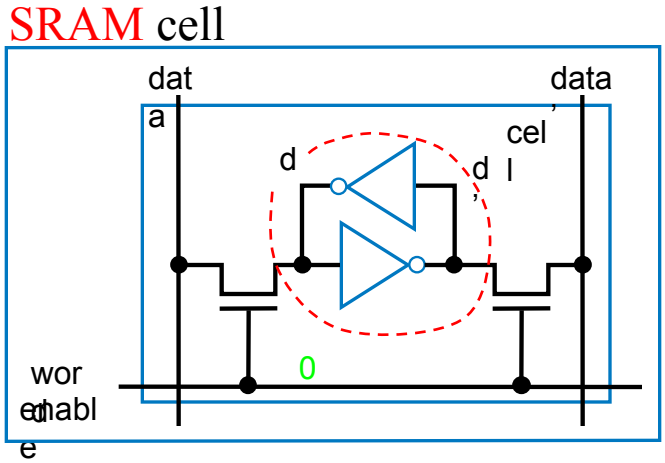


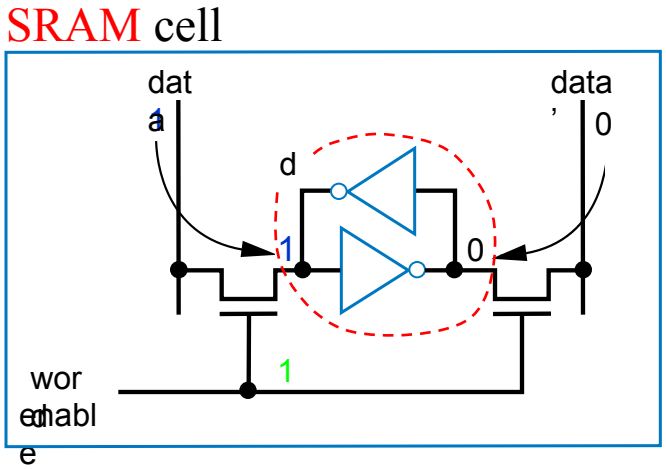
Similar internal structure as register file

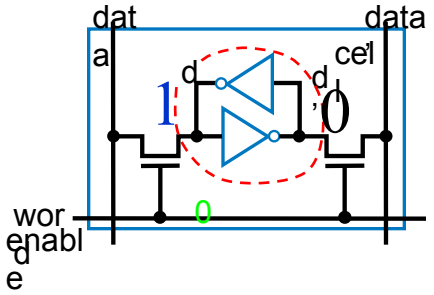
- Decoder enables appropriate word based on address inputs
- rw controls whether cell is written or read
- rd and wr data lines typically combined
- Let's see what's inside each RAM cell

Static RAM (SRAM)



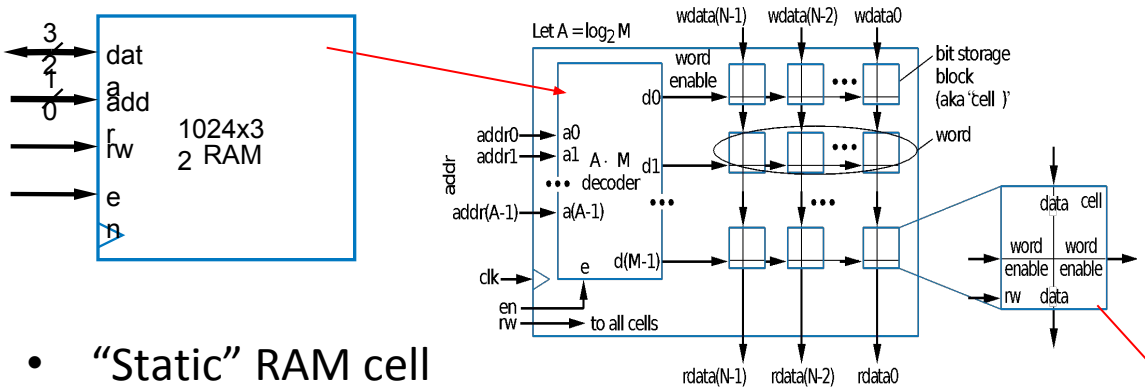






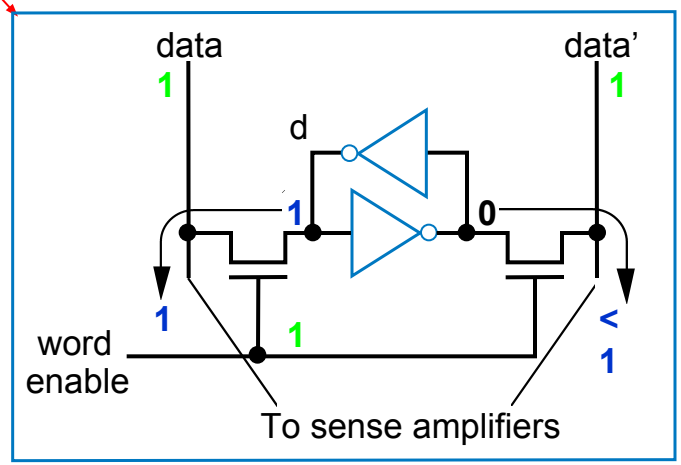
- "Static" RAM cell
  - 6 transistors (recall inverter is 2 transistors)
  - Writing this cell
    - *word enable* input comes from decoder
    - When 0, value *d* loops around inverters
      - That loop is where a bit stays stored
    - When 1, the *data* bit value enters the loop
      - *data* is the bit to be stored in this cell
      - *data'* enters on other side
      - Example shows a "1" being written into cell

Static RAM (SRAM)



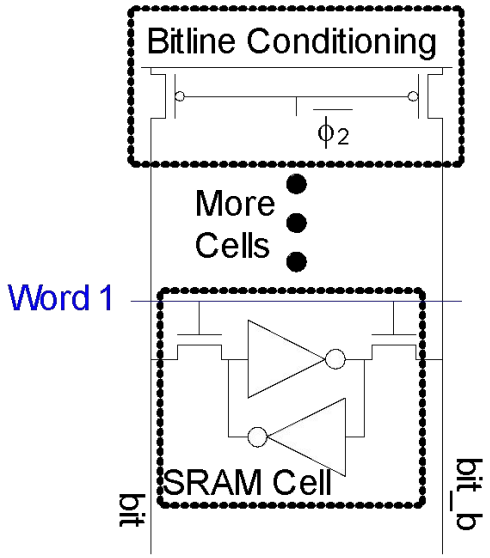
- "Static" RAM cell
  - Reading this cell
    - Somewhat trickier
    - When rw set to read, the RAM logic sets both *data* and *data'* to 1
    - The stored bit d will pull either the left line or the right bit down slightly below 1
    - "Sense amplifiers" detect which side is slightly pulled down
  - The electrical description of SRAM is really beyond our scope – just general idea here, mainly to contrast with *DRAM*...

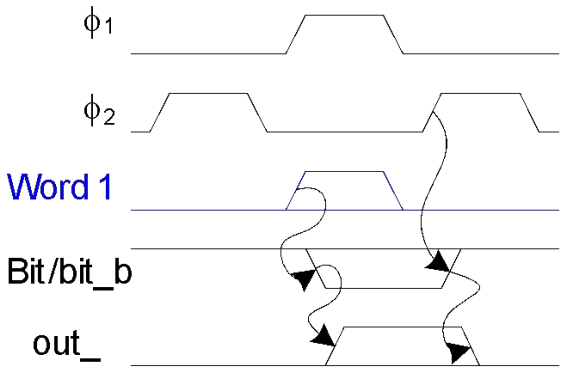
SRAM cell



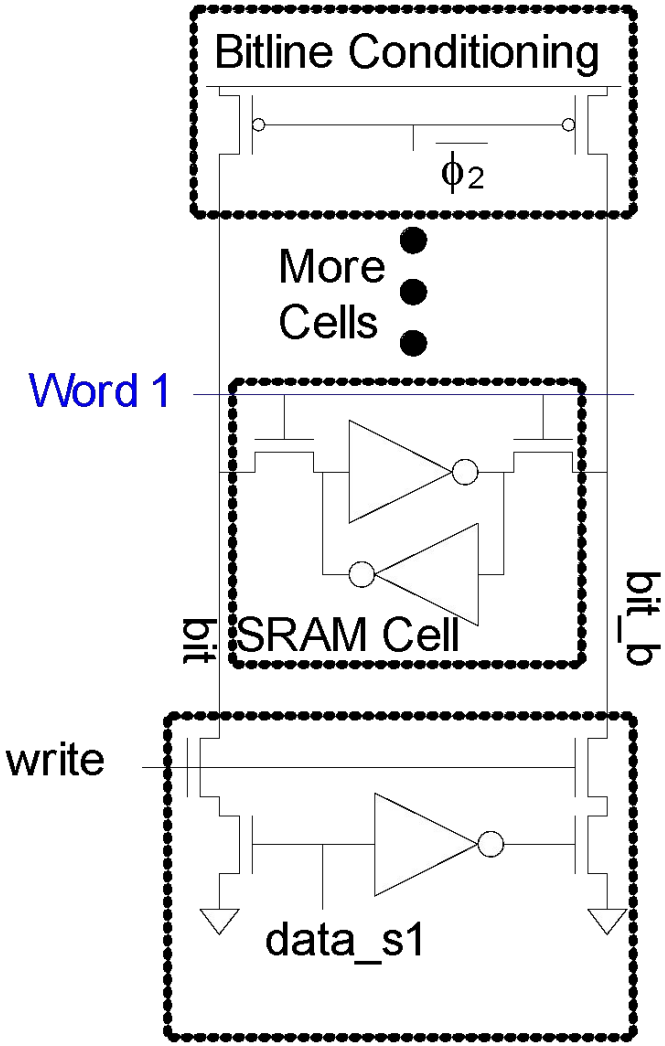
SRAM Column Example

Read

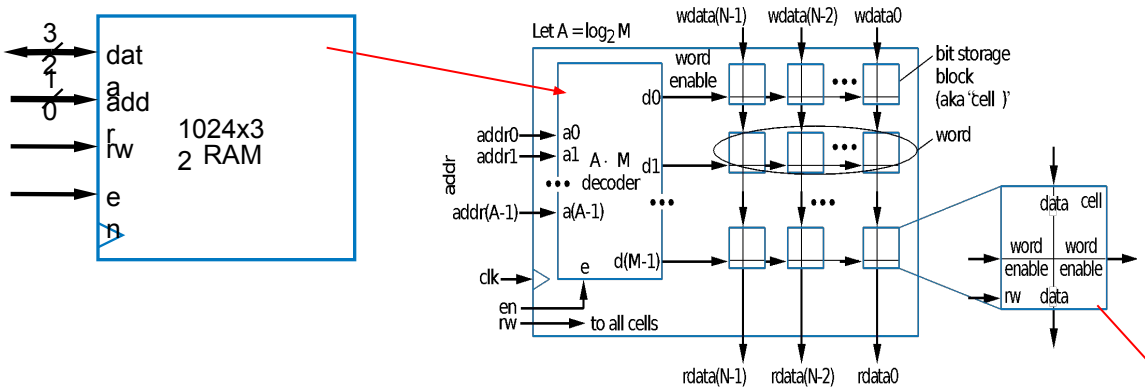


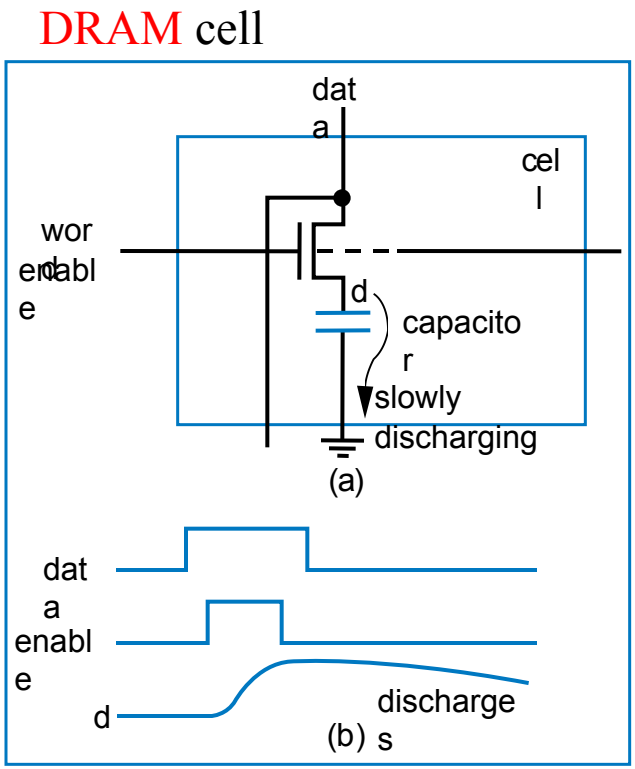


Write



Dynamic RAM (DRAM)

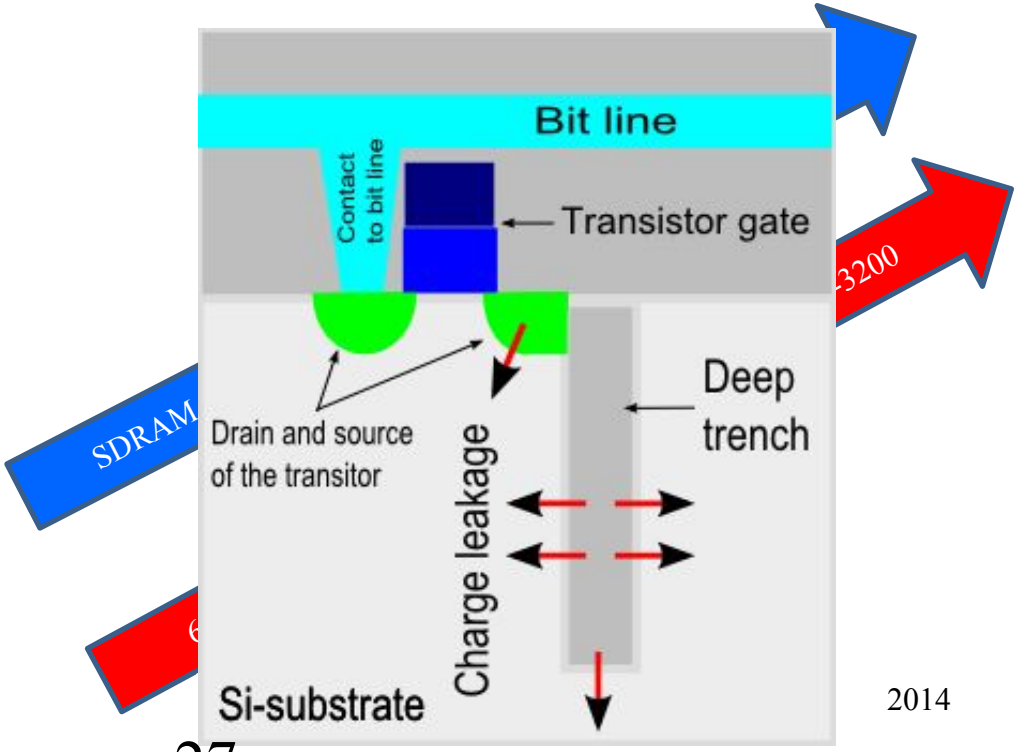


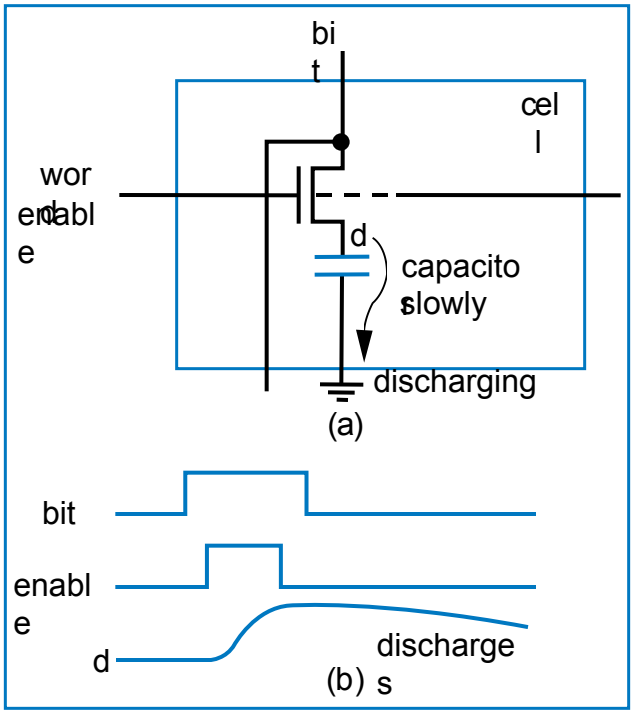


- "Dynamic" RAM cell
  - 1 transistor (rather than 6)
  - Relies on *large* capacitor to store bit
    - Write: Transistor conducts, data voltage level gets stored on top plate of capacitor
    - Read: Just look at value of *d*
    - Problem: Capacitor discharges over time
      - Must "refresh" regularly, by reading *d* and then writing it right back

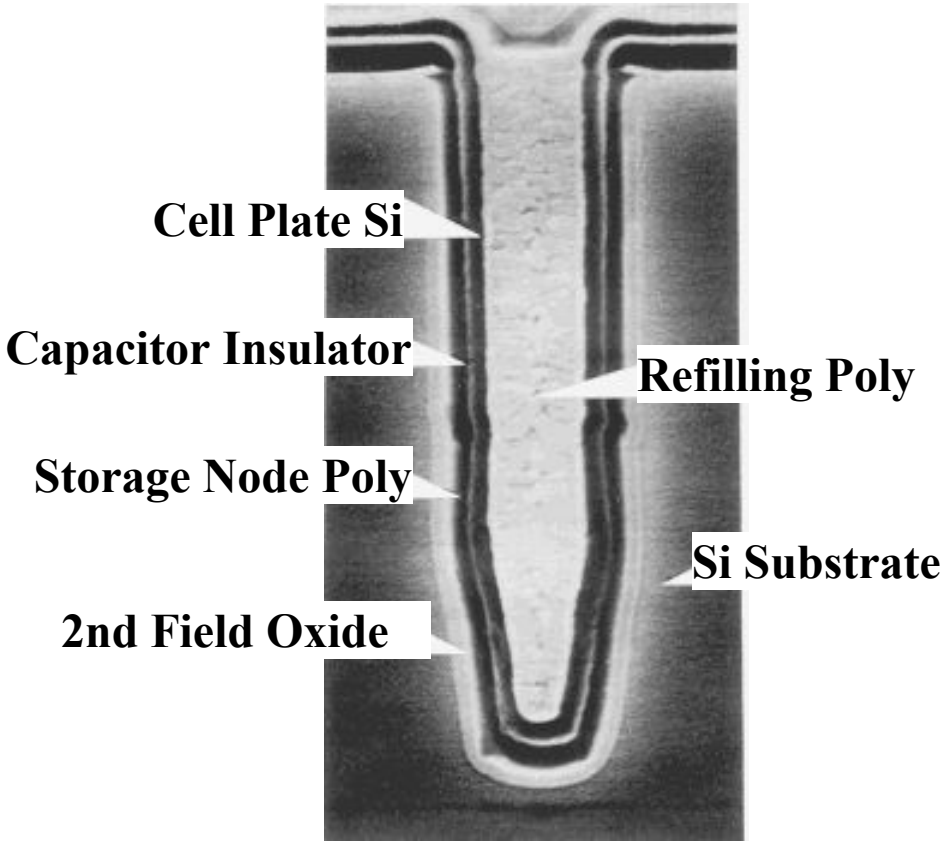
Dynamic RAM (DRAM)

- "Dynamic" RAM cell
  - 1 transistor (rather than 6)
  - Relies on capacitor to store bit

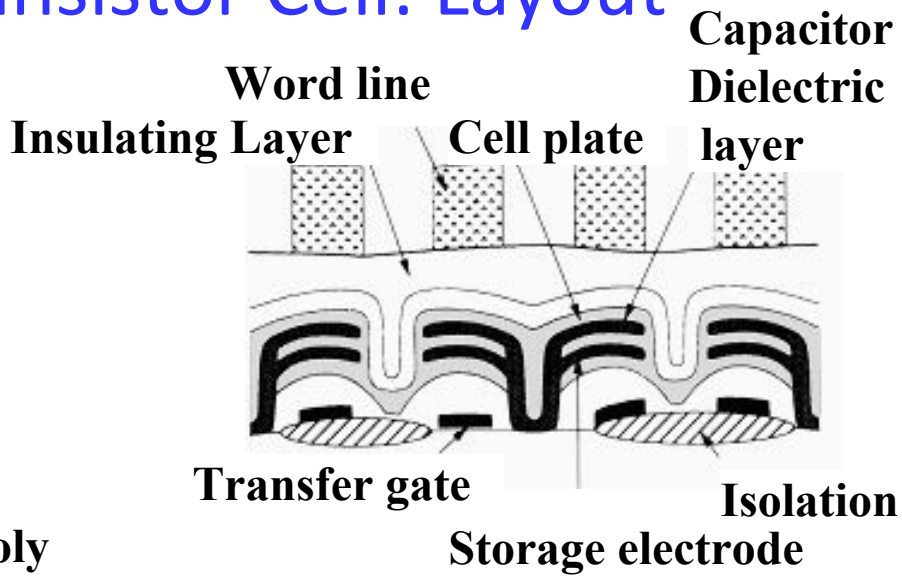


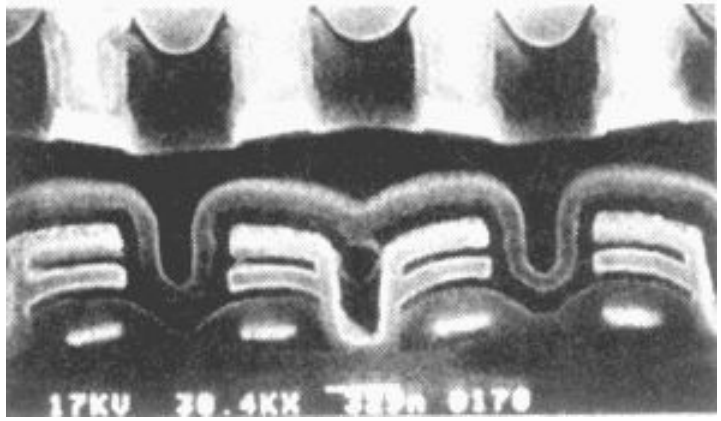


Dynamic RAM 1-Transistor Cell: Layout



Trench Cell

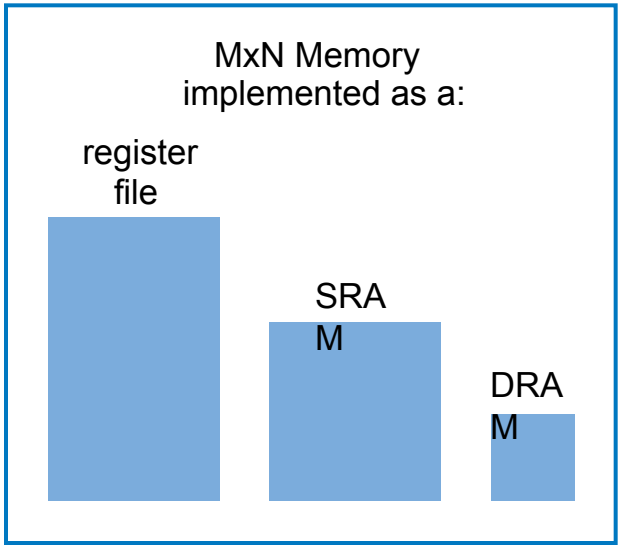




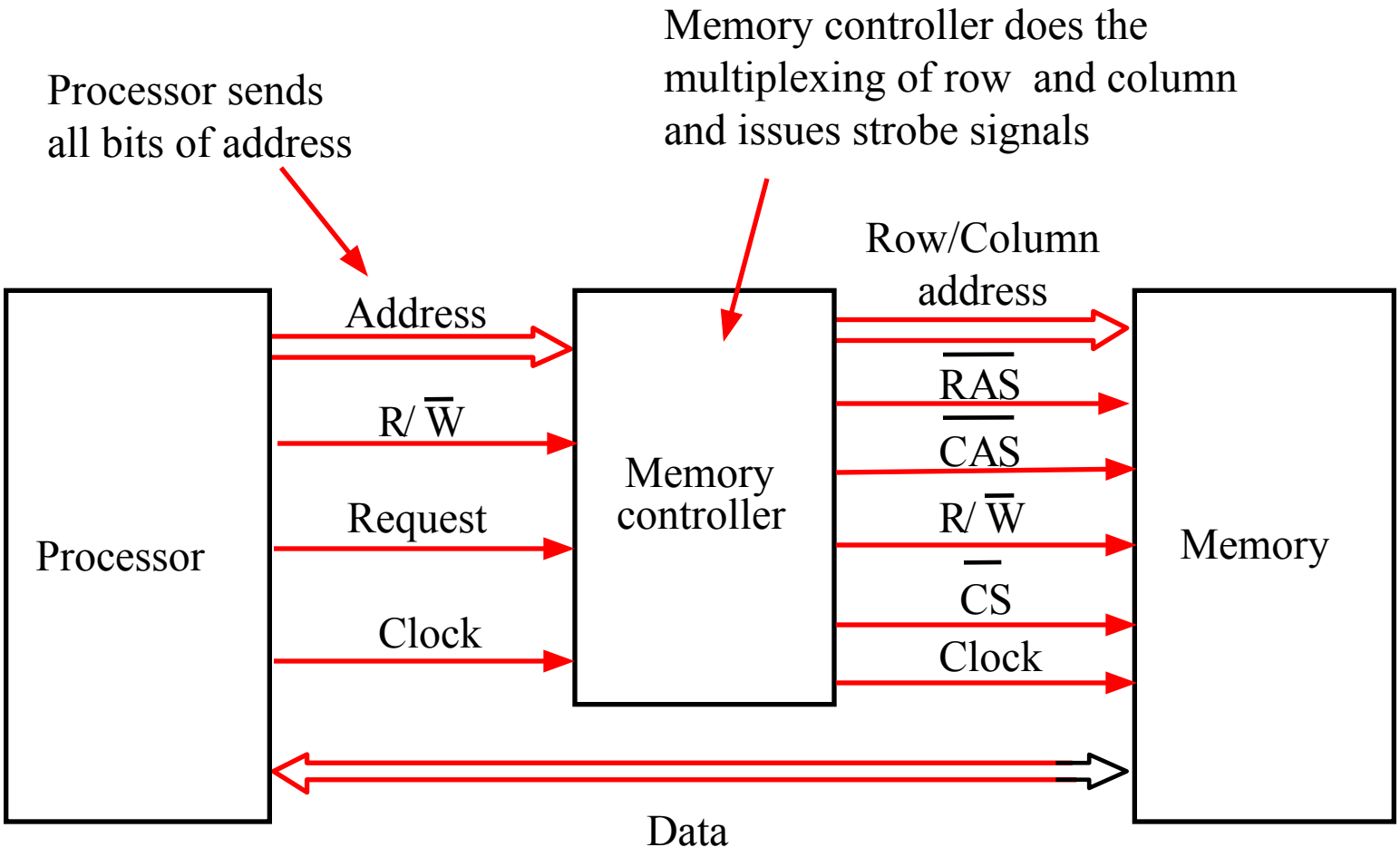
Stacked-capacitor Cell

Comparing Memory

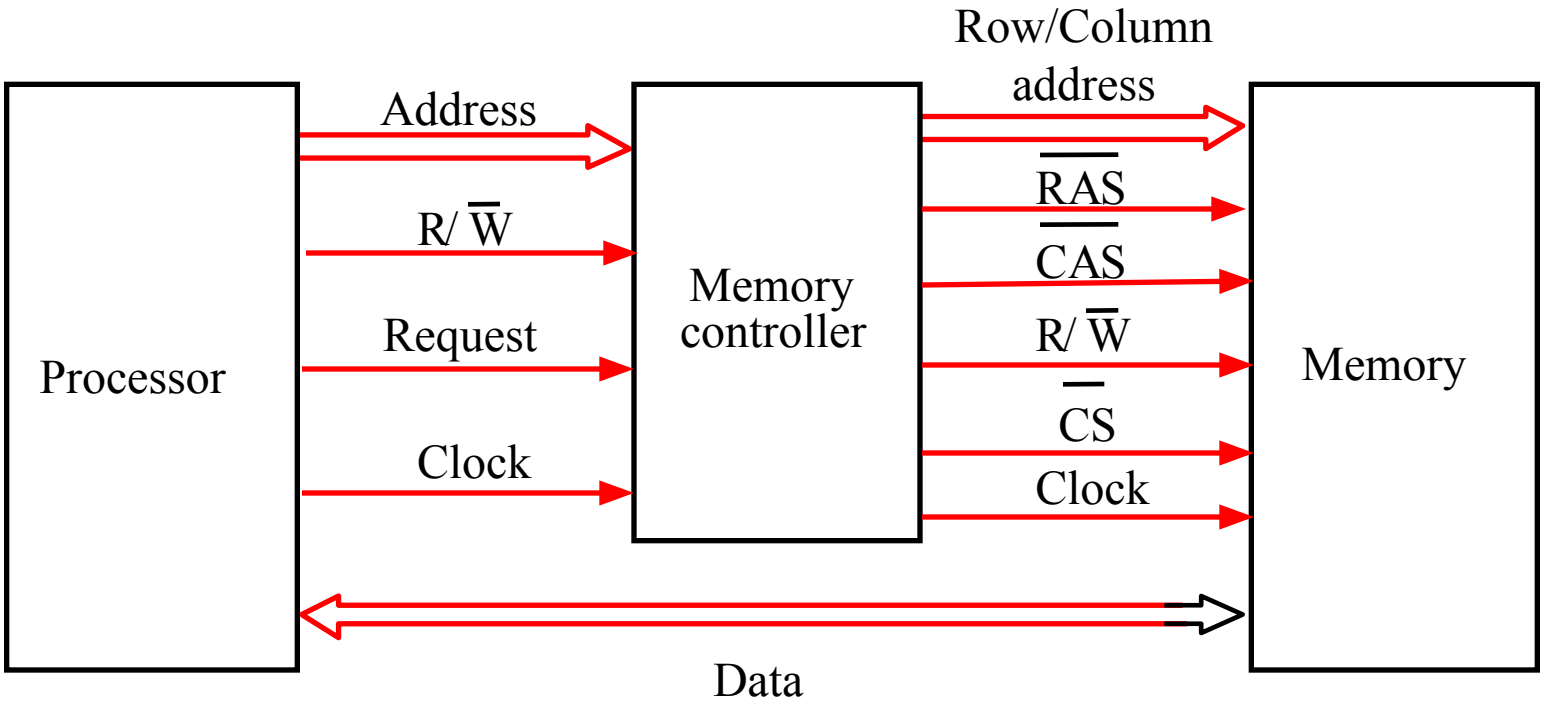
- Register file
  - Fastest
  - But biggest size
- SRAM
  - Fast
  - More compact than register file
- DRAM
  - Slowest (capacitor)
    - And refreshing takes time
  - But very compact (lower cost)
- Use register file for small items, SRAM for large items, and DRAM for huge items
  - Note: DRAM's big capacitor requires a special chip design process, so DRAM is often a separate chip



Size comparison for same number of bits (not to scale)



Use of a memory controller.



Memory controller provides the refresh control if not done on the chip

Refreshing typically once every 64 ms. At a cost of .2ms

Less than .4% overhead

Use of a memory controller.

CAM

- CAM vs. RAM