CSEE 3827: Fundamentals of Computer Systems, Spring 2011

6. Memory Arrays

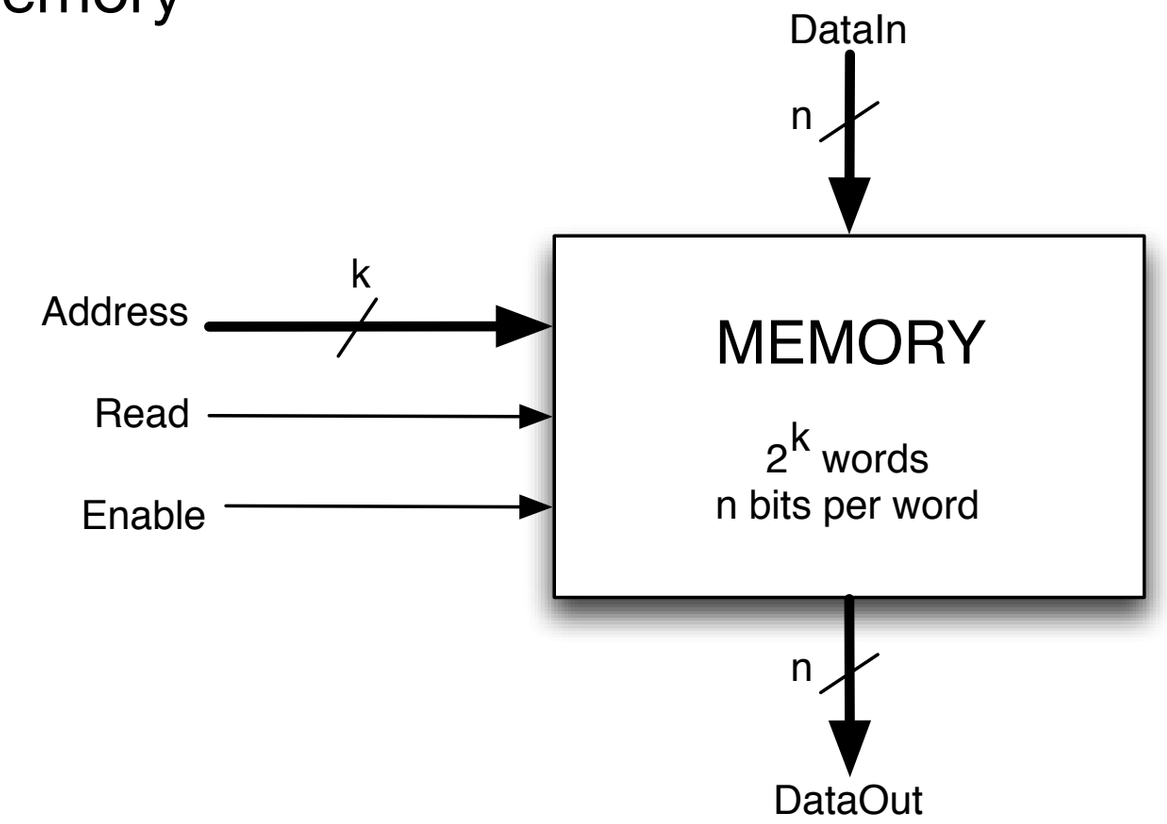
Prof. Martha Kim (martha@cs.columbia.edu)
Web: http://www.cs.columbia.edu/~martha/courses/3827/sp11/

Outline (H&H 5.5-5.6)

- Memory Arrays
  - RAM, ROM, SRAM, DRAM
- Logic Arrays
  - Programmable Logic Arrays (PLAs)
  - Field-Programmable Gate Arrays (FPGAs)

Memory interface

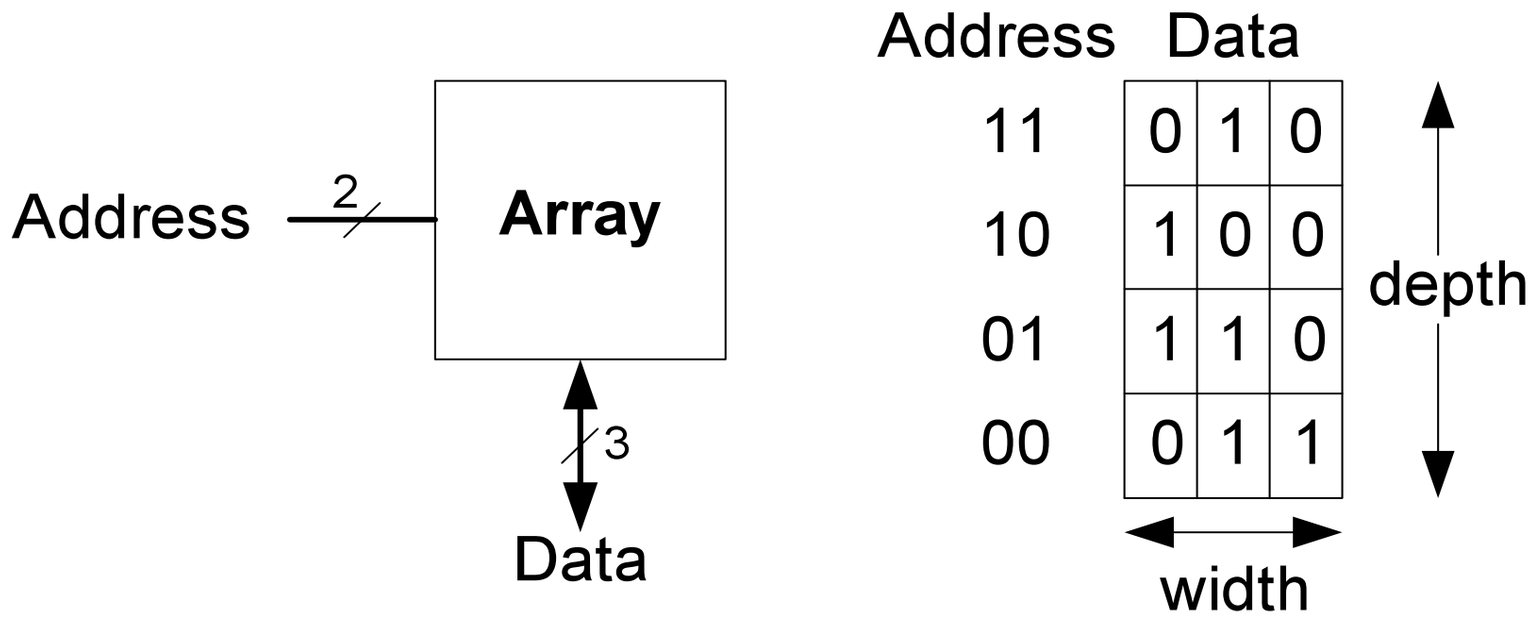
- Stores data in *word* units
- A *word* is several bytes (16-, 32-, or 64-bit words are typical)
- *write* operations store data to memory
- *read* operations retrieve data from memory



An n-bit value can be read from or written to each k-bit address

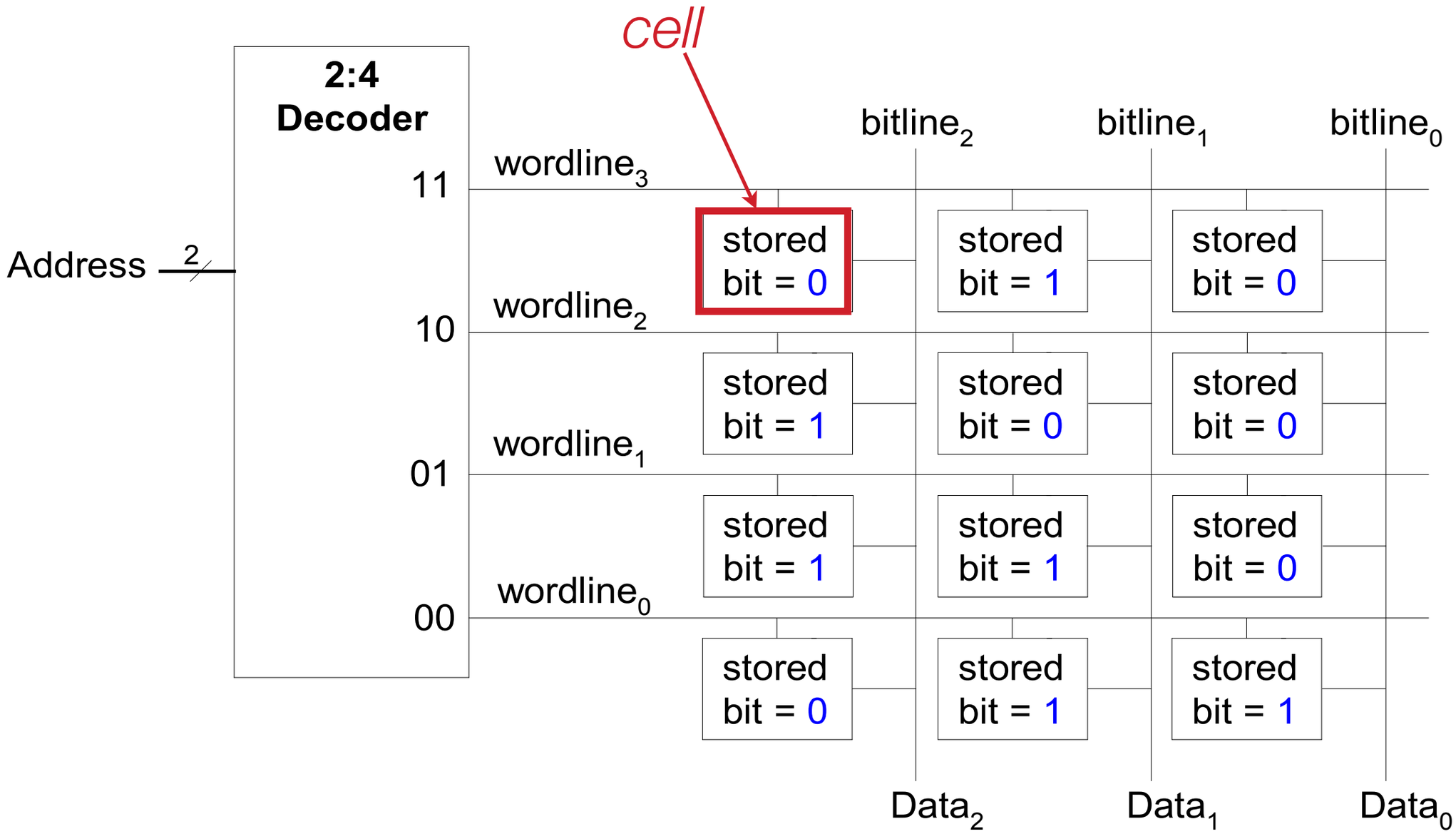
Memory Array: Example

- 22 × 3-bit array
- Number of words: 4
- Word size: 3-bits
- For example, the 3-bit word stored at address 10 is 100



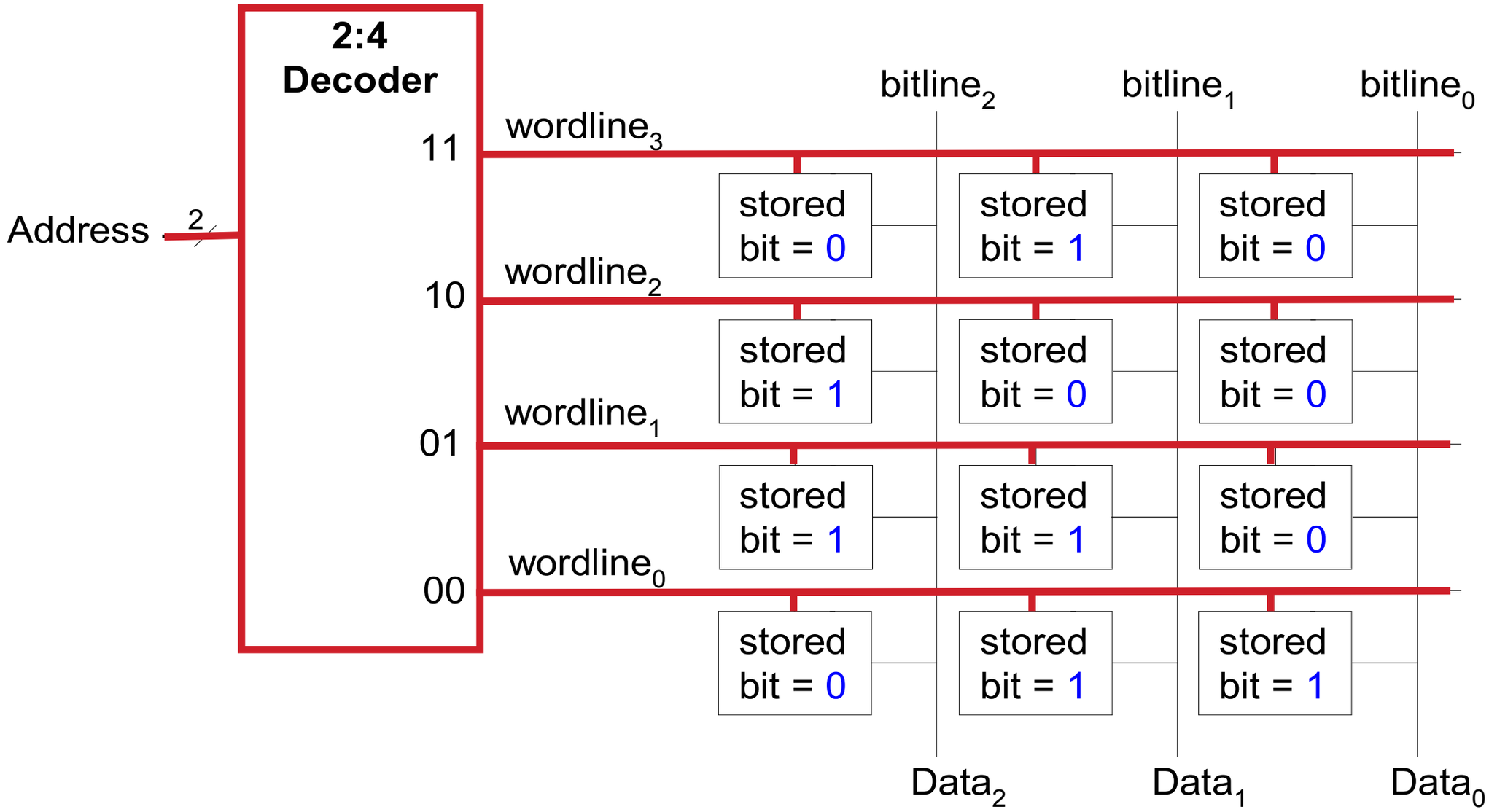
Memory array architecture (1)

Memory is a 2D array of bits. Each bit stored in a *cell*.



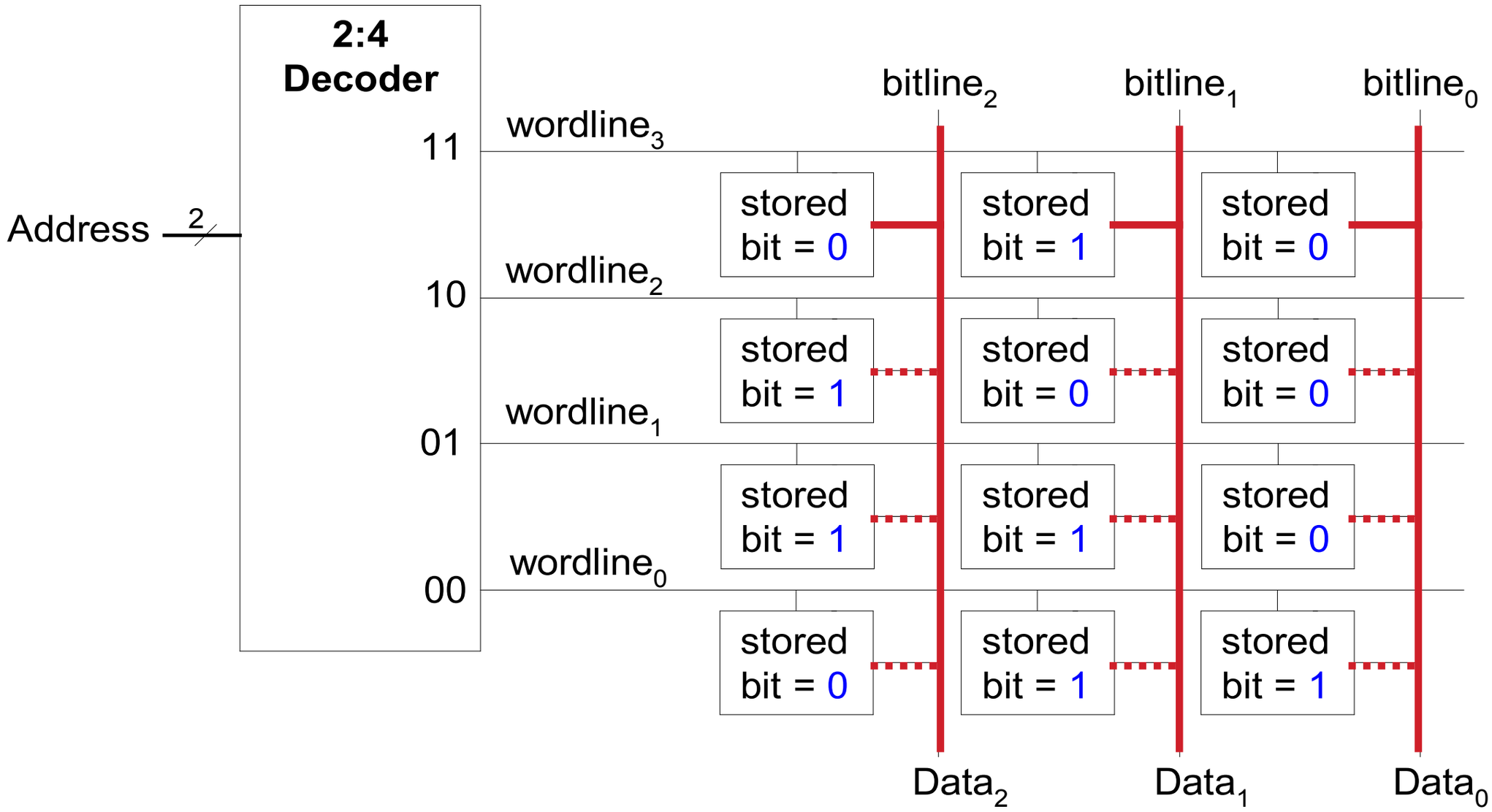
Memory array architecture (2)

Address is decoded into set of *wordlines*.
Wordlines select row to be read/written.
Only one wordline=1 at a time.



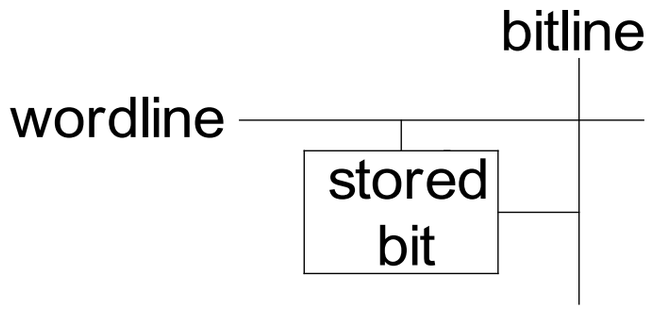
Memory array architecture (3)

When reading, contents of word written to *bitlines.*



Memory cell abstraction

Cell is base element of memory that stores a single bit



wordline	stored bit	bitline
0	x	Z
1	0	0
1	1	1

Implemented with a tristate buffer. Value "Z" does not drive output wire to either a 0 or 1.

Implementation of cell varies with type of memory.

Types of memory

Random access memory (RAM)

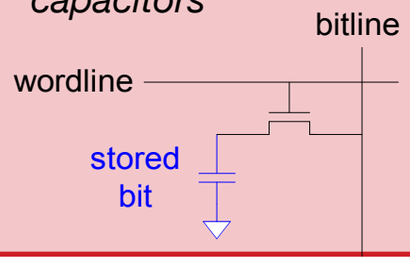
Volatile (data lost on power off)

Fast reads and writes

Historically called RAM because equal time to read/write all addresses (in contrast to serial-access devices such as a hard disk or tape). Somewhat misleading as ROM also can have uniform access time.

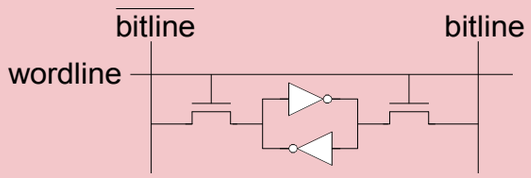
Dynamic RAM (DRAM)

Cell stores data w. capacitors



Static RAM (SRAM)

Cell stores data w. cross-coupled inverters



Flip-flop

Register

Read-only memory (ROM)

Non-volatile (retains data when powered off)

Fast reads, writing is impossible or slow (again, misleading name)

Historically called ROMs because written by permanently blowing fuses (so rewriting was impossible). Modern ROMs, such as flash memory in iPod are rewritable, just slowly.

ROM

Mask-programmed (at chip fab)

PROM

Fuse- or antifuse-programmed

FLASH

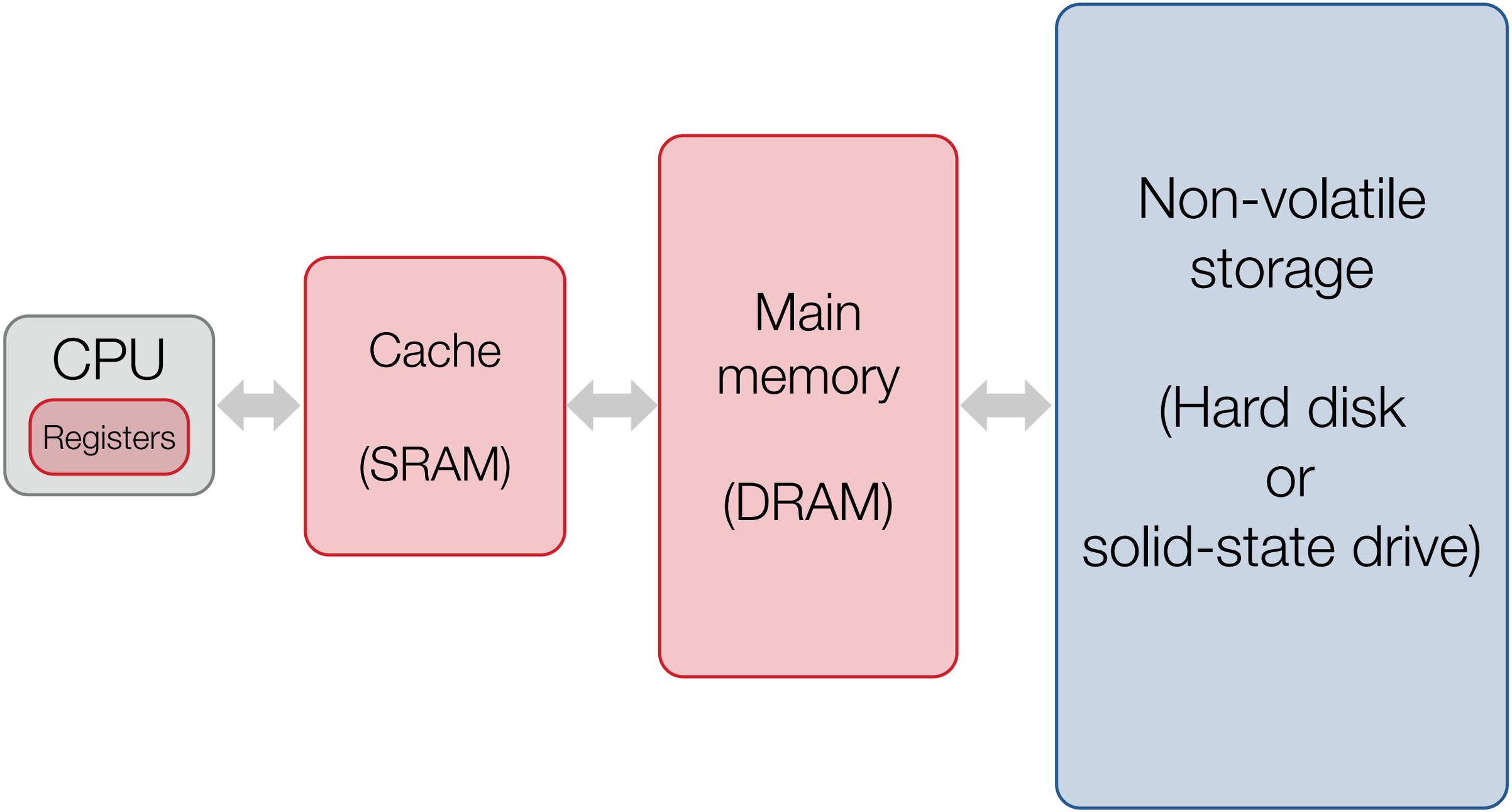
Electrically erasable floating gate with multiple erasure and programming modes

Hard Disk

Volatile storage (RAM) comparisons

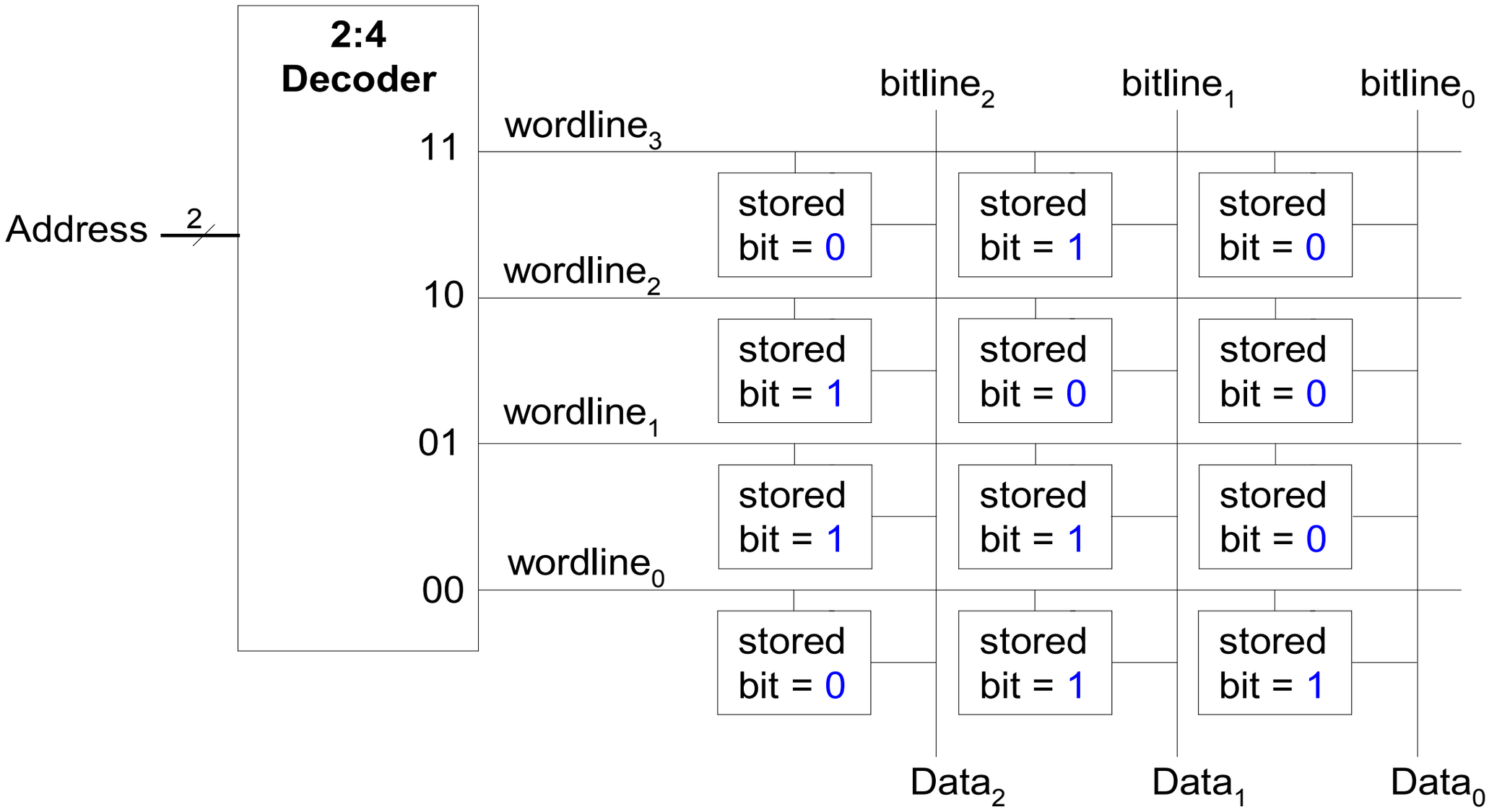
	Flip-flop	SRAM	DRAM
Transistors / bit	~20	6	1
Density	Low	Medium	High
Access time	Fast	Medium	Slow
Destructive read?	No	No	Yes (refresh required)
Power consumption	High	Medium	Low

Storage hierarchy



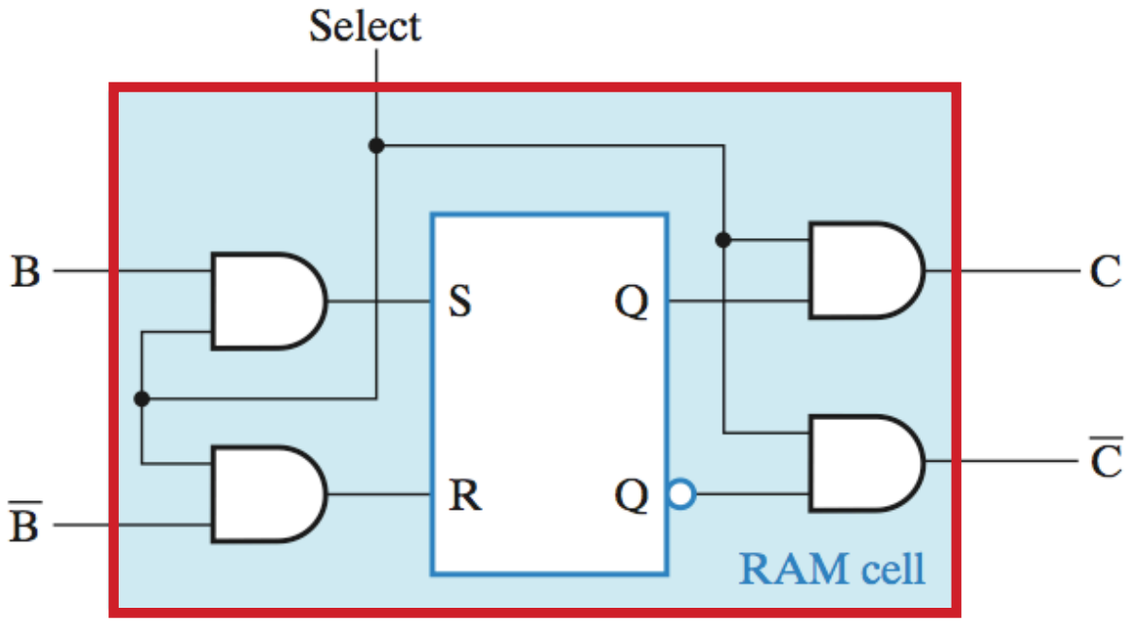
Fast access to small amount of data ⟷ *Slow access to large amount of data*

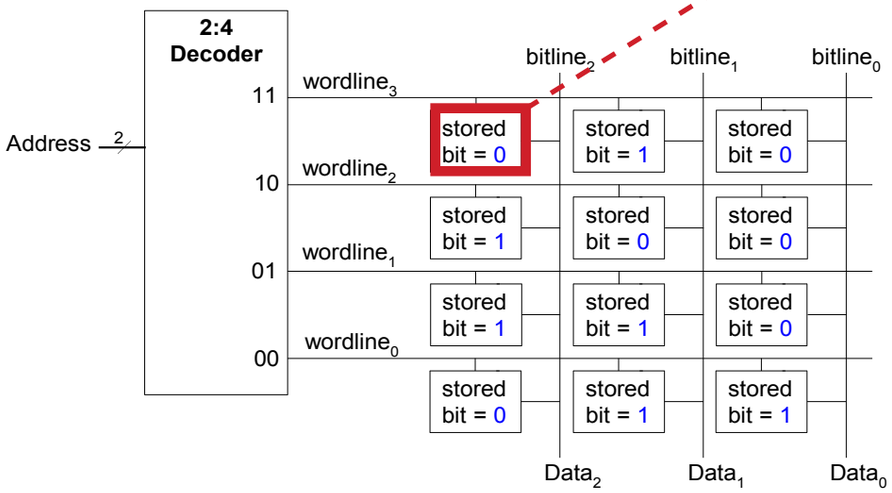
Bottom-up examination of SRAM circuits



Bottom-up examination of SRAM circuits (2)

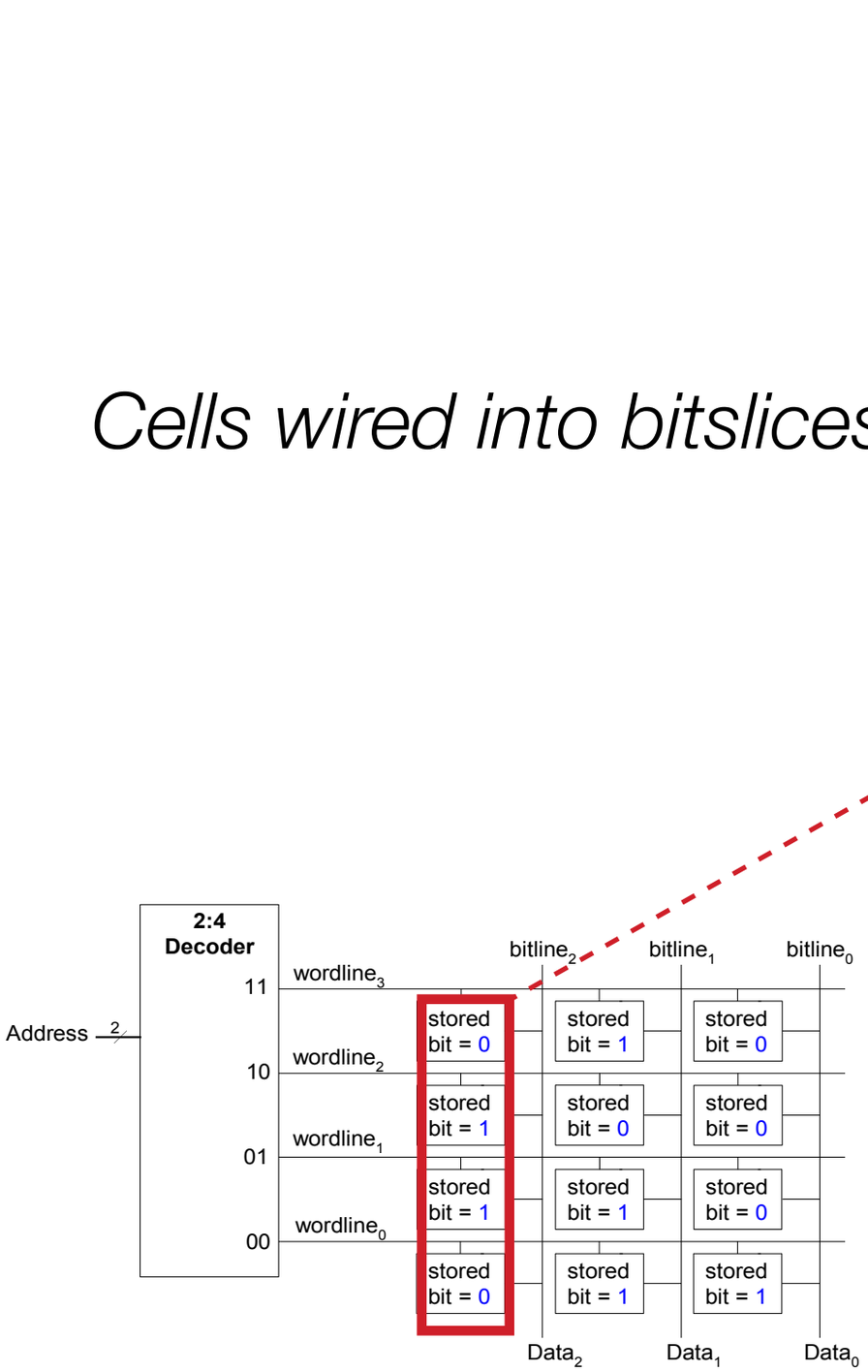
Bits stored in cells
(modeled by an SR latch)

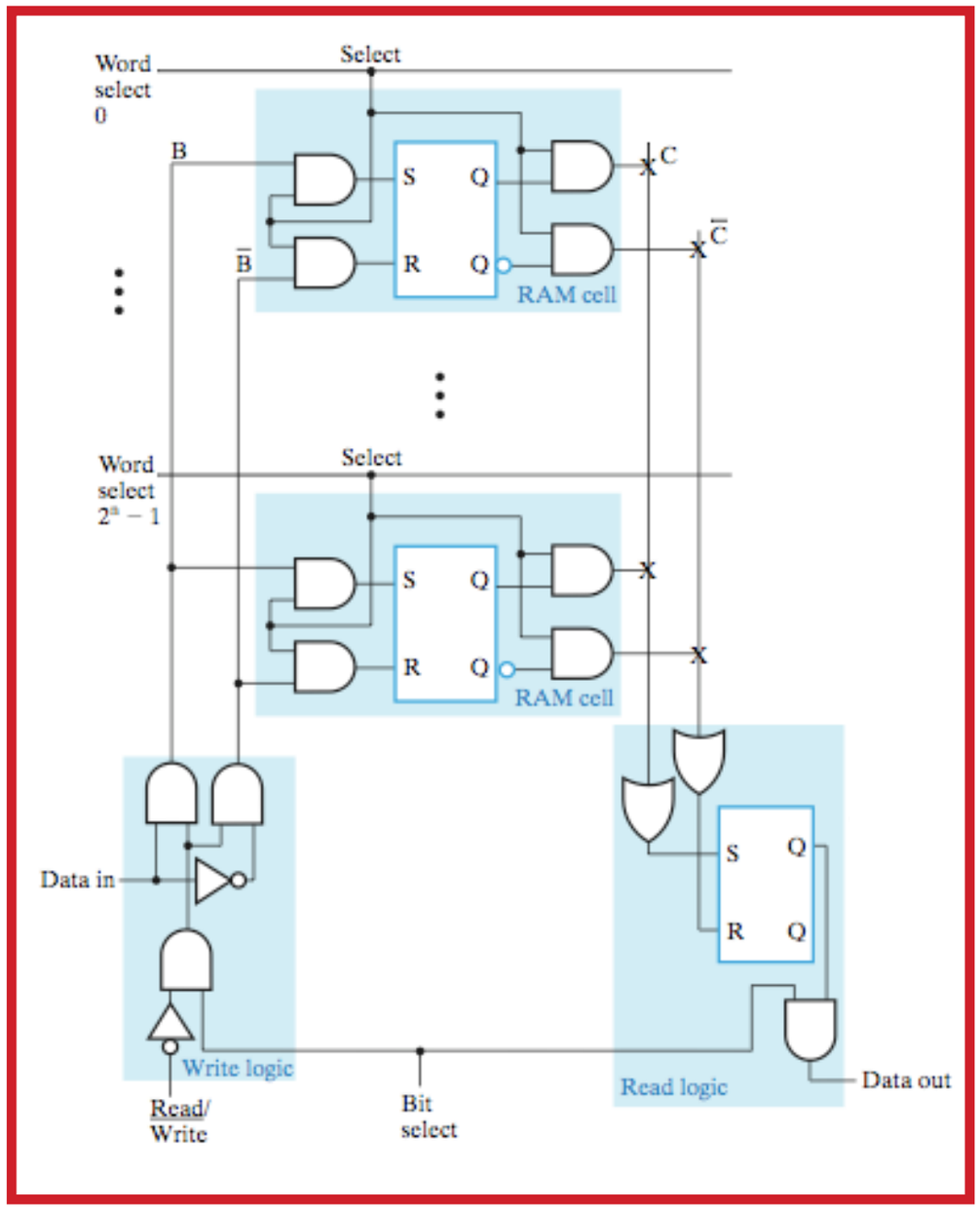




Bottom-up examination of SRAM circuits (3)

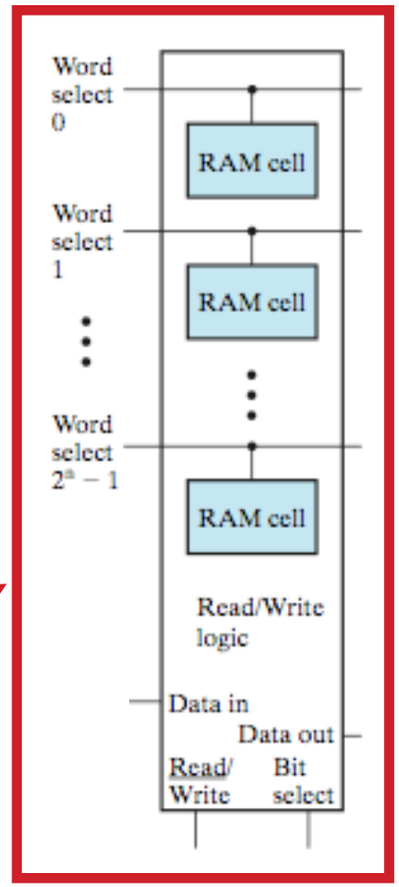
Cells wired into bitslices.

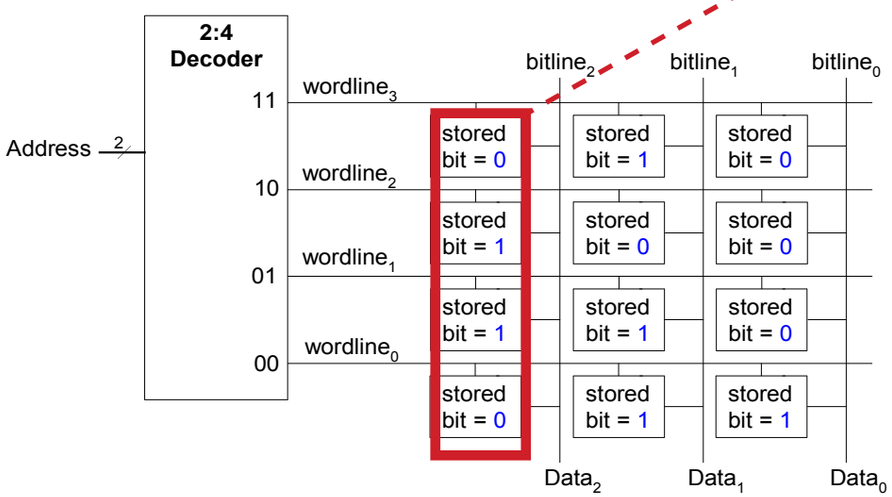




Bottom-up examination of SRAM circuits (4)

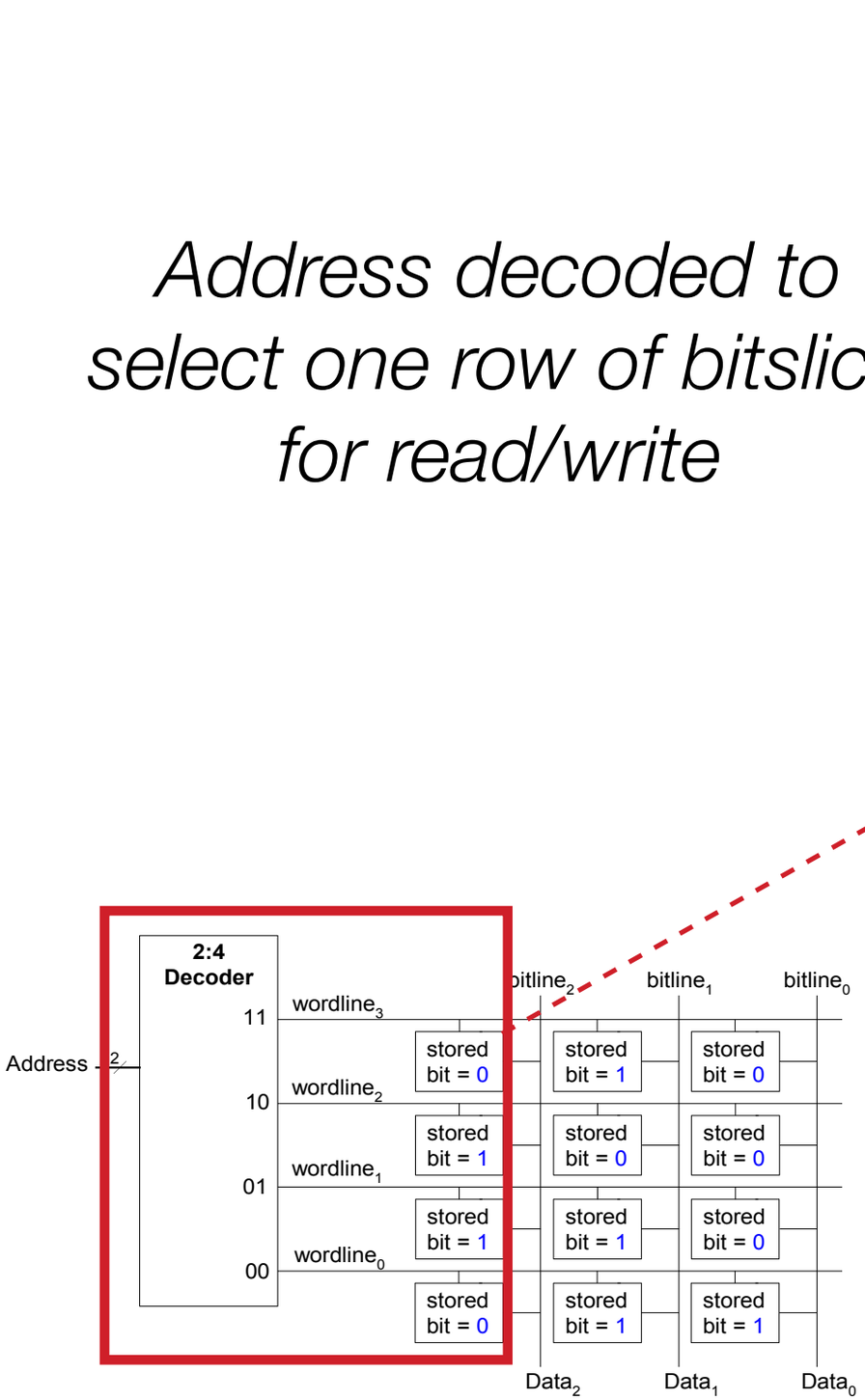
Block diagram of a bitslice

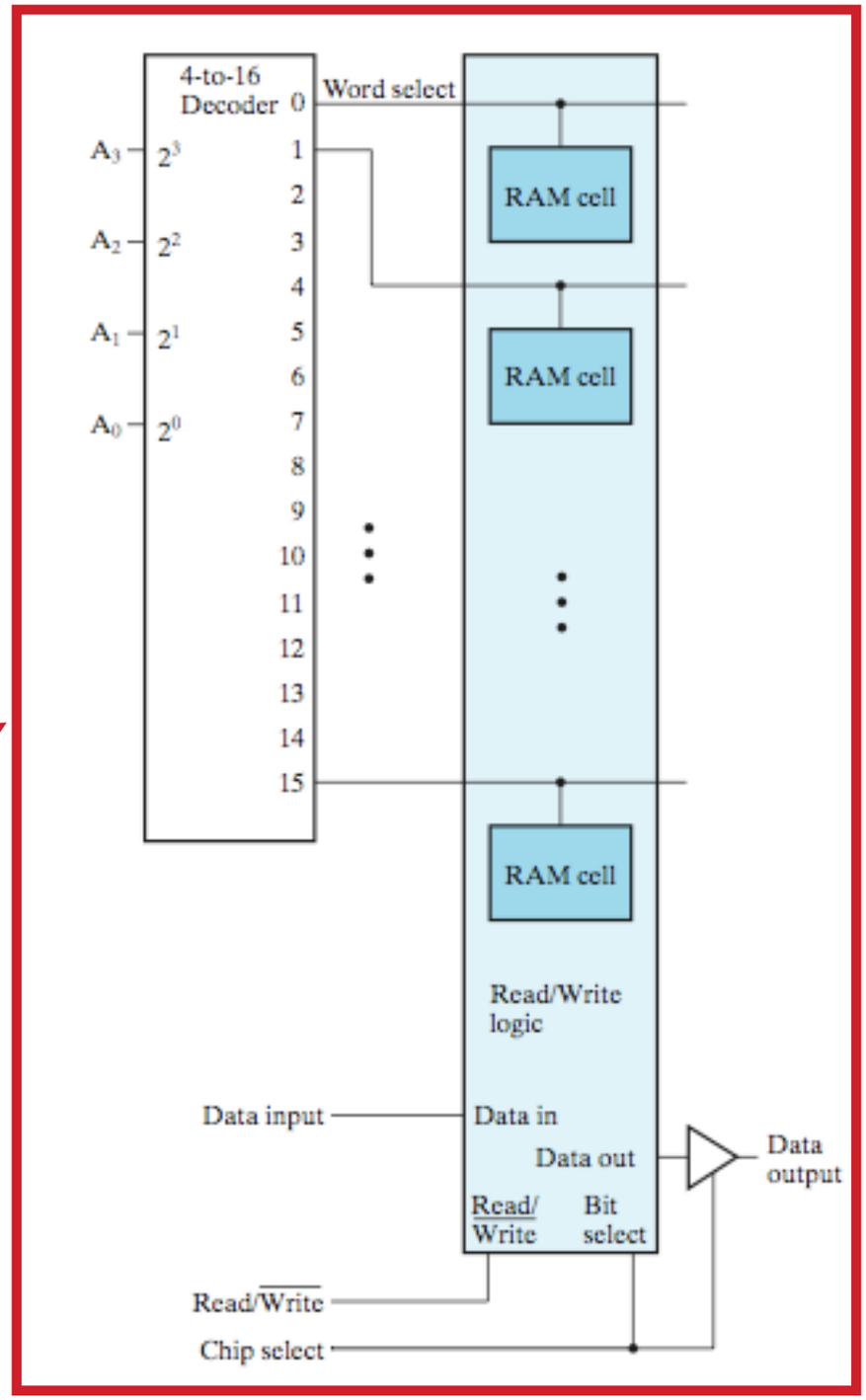




Bottom-up examination of SRAM circuits (5)

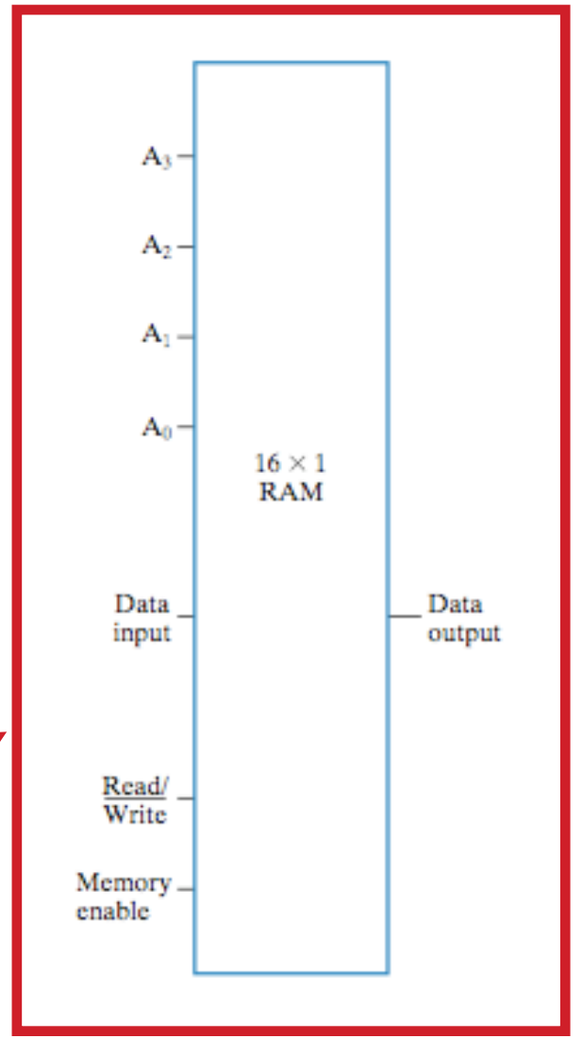
Address decoded to select one row of bitslice for read/write

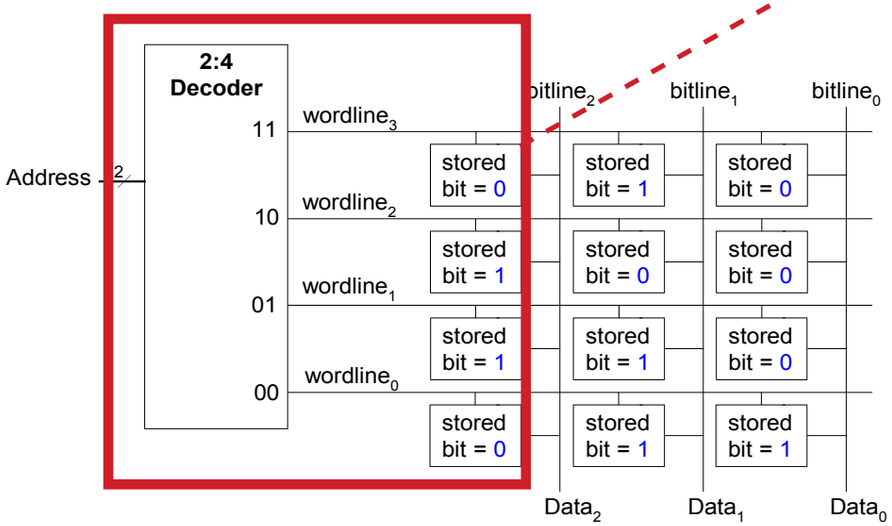




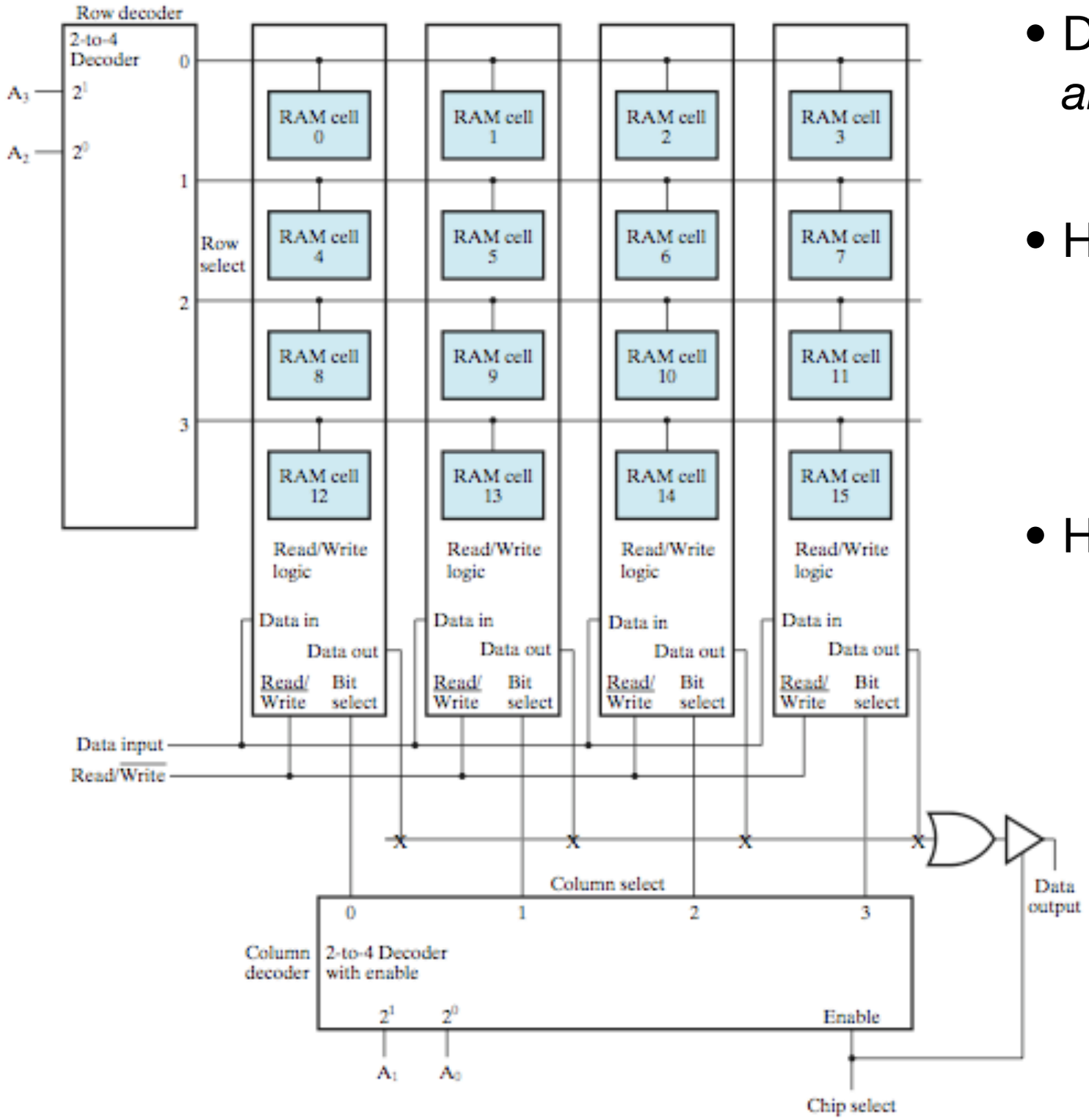
Bottom-up examination of SRAM circuits (6)

To increase word size, add bitslices.



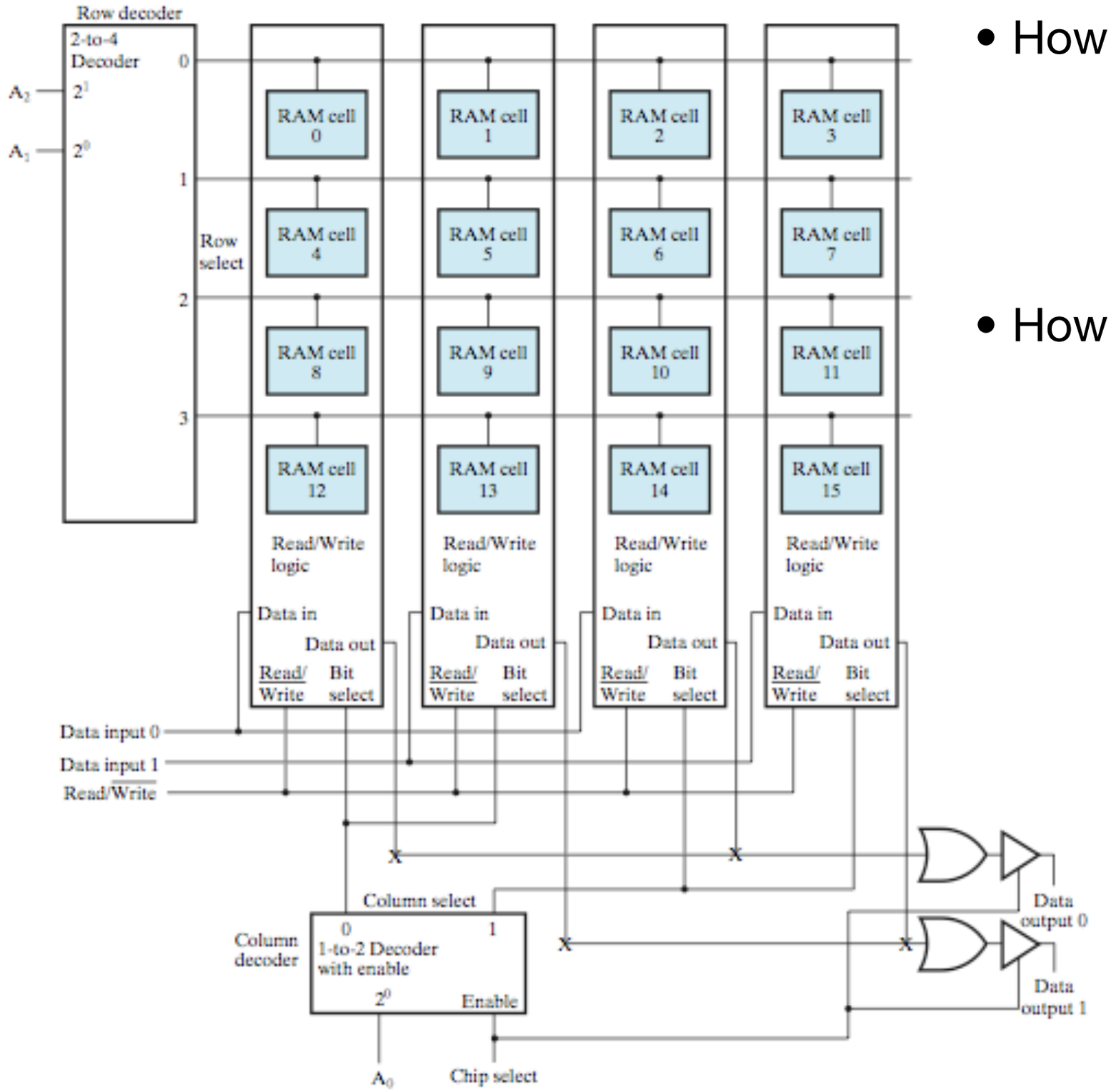


Coincident cell selection



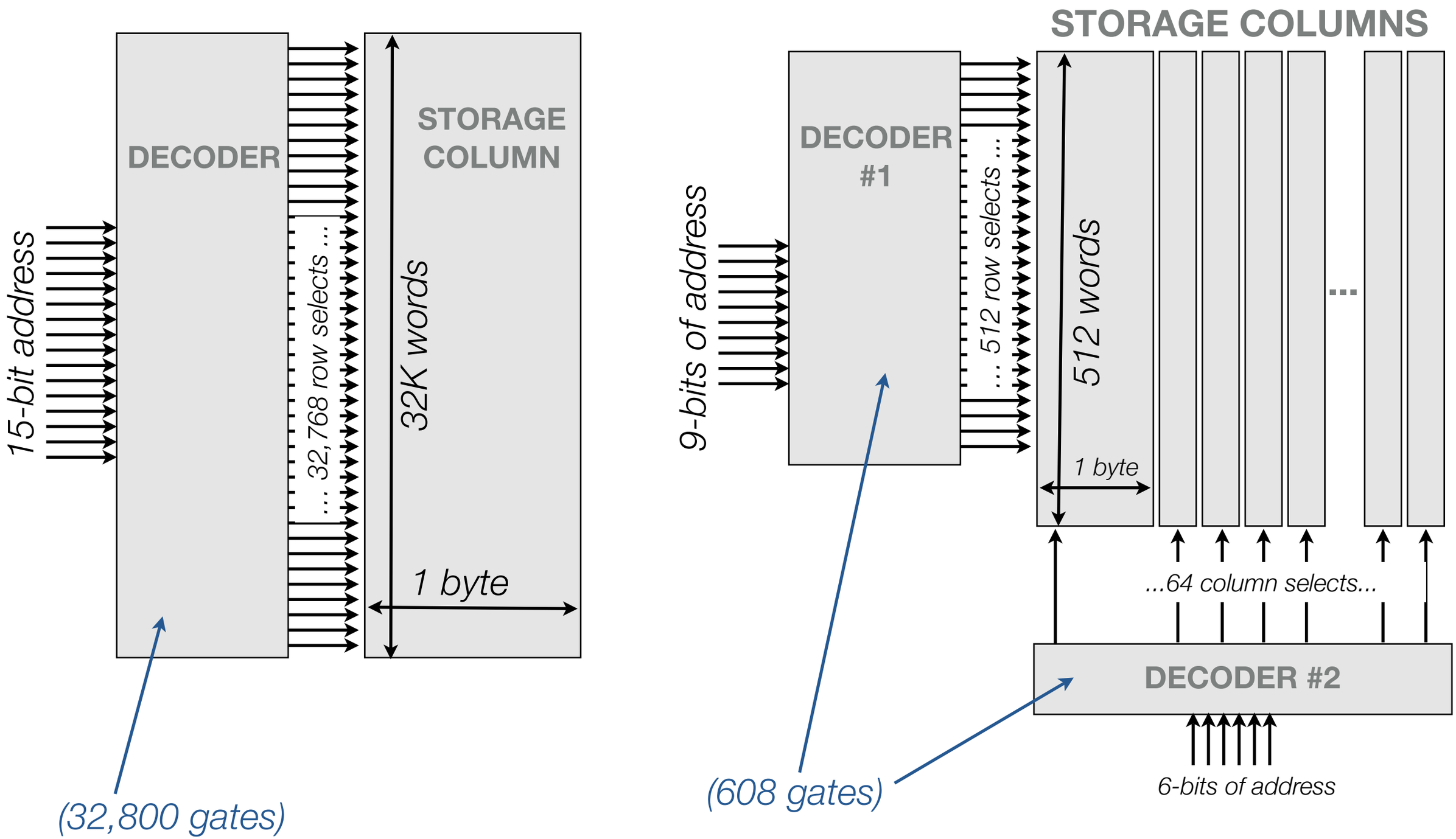
- Decode address into *both* row *and* column select signals
- How many words in this RAM?
- How many bits per word?

Coincident cell selection w. larger words

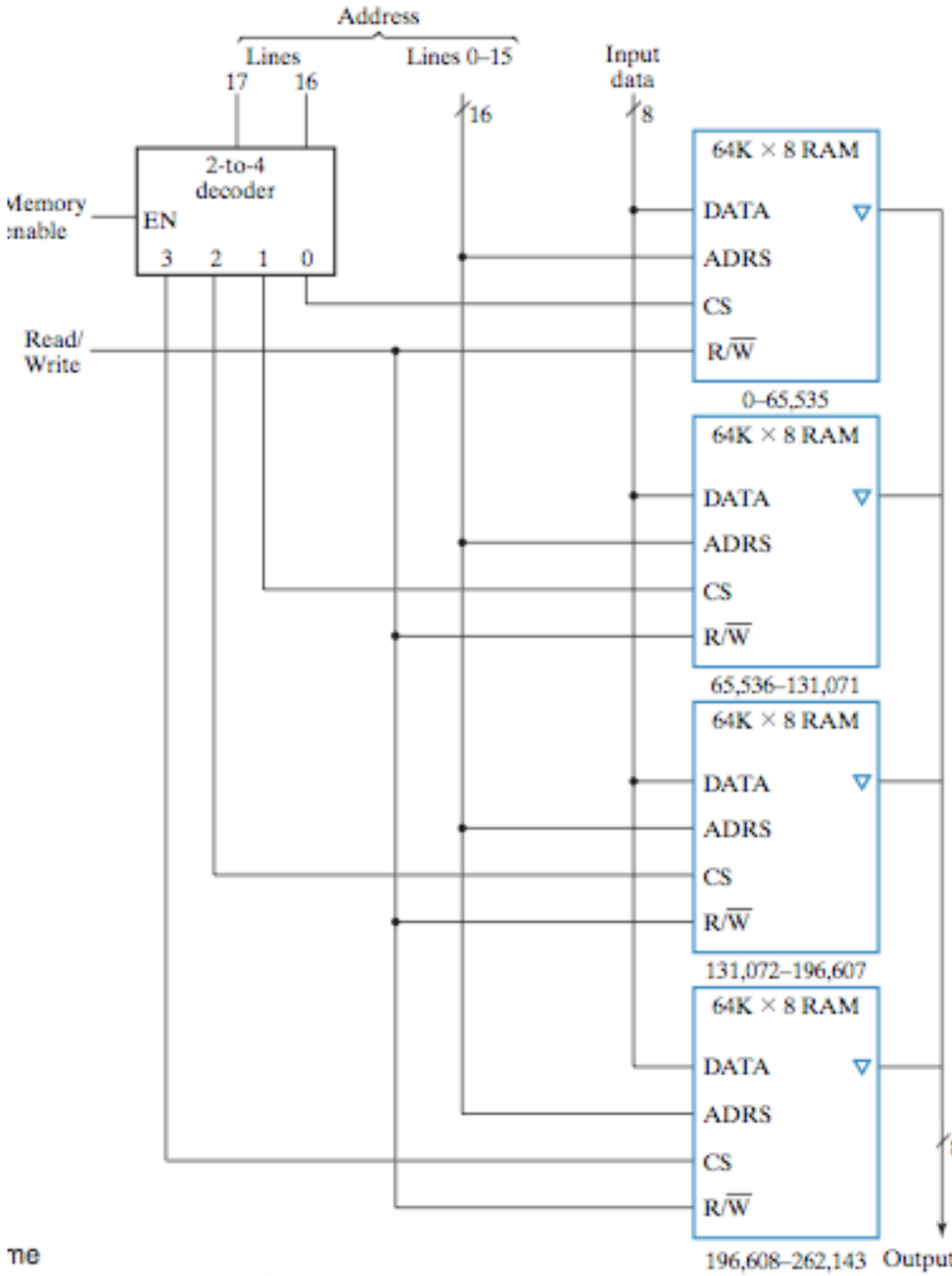


- How many words in this RAM?
- How many bits per word?

Coincident cell selection saves decode logic

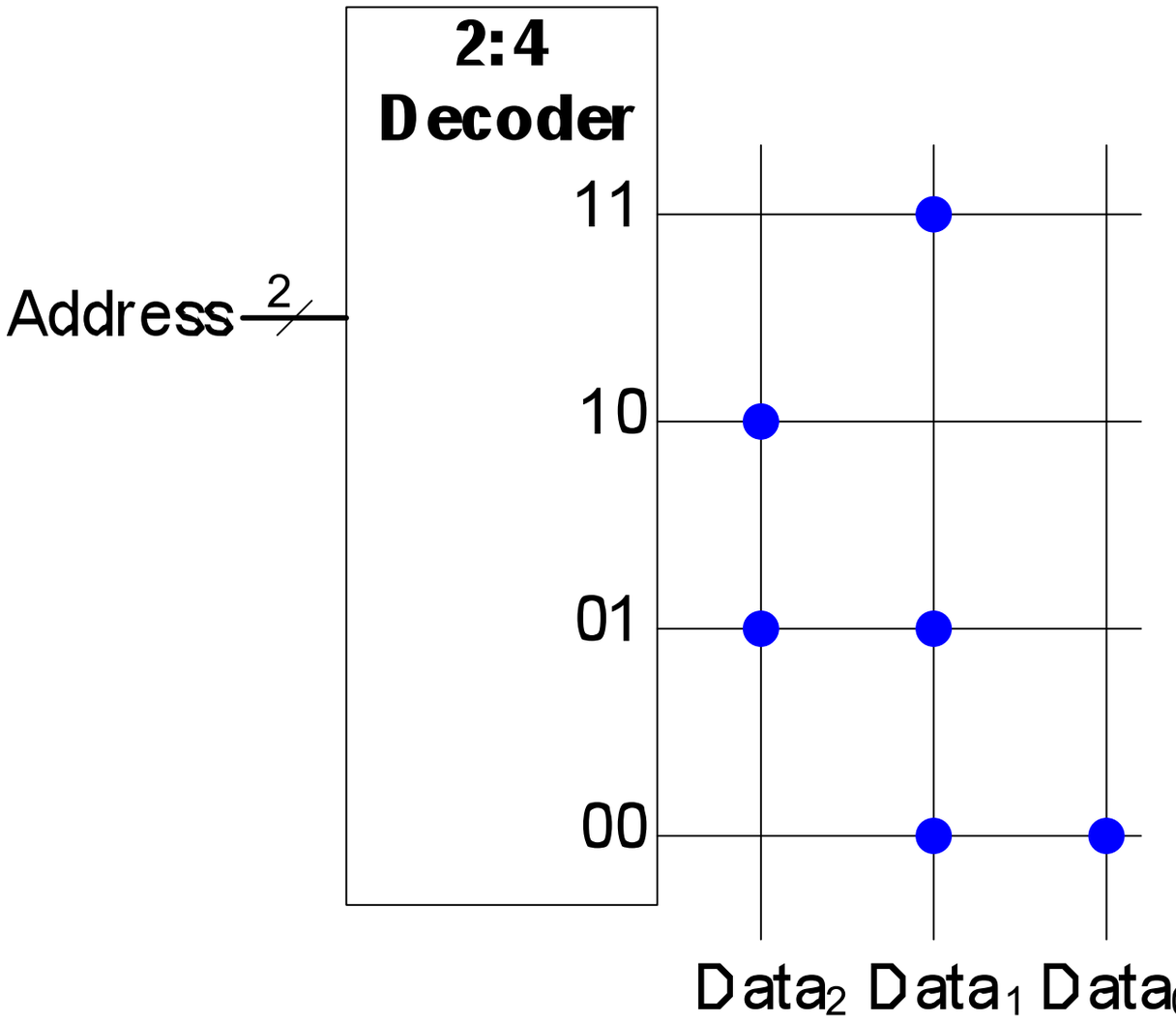


Multi-chip memories



- If you need a larger memory than any available chip
- Wire multiple RAM chips together to work in concert as one large memory

ROMs: Dot Notation



Address	Data		
11	0	1	0
10	1	0	0
01	1	1	0
00	0	1	1

depth

width

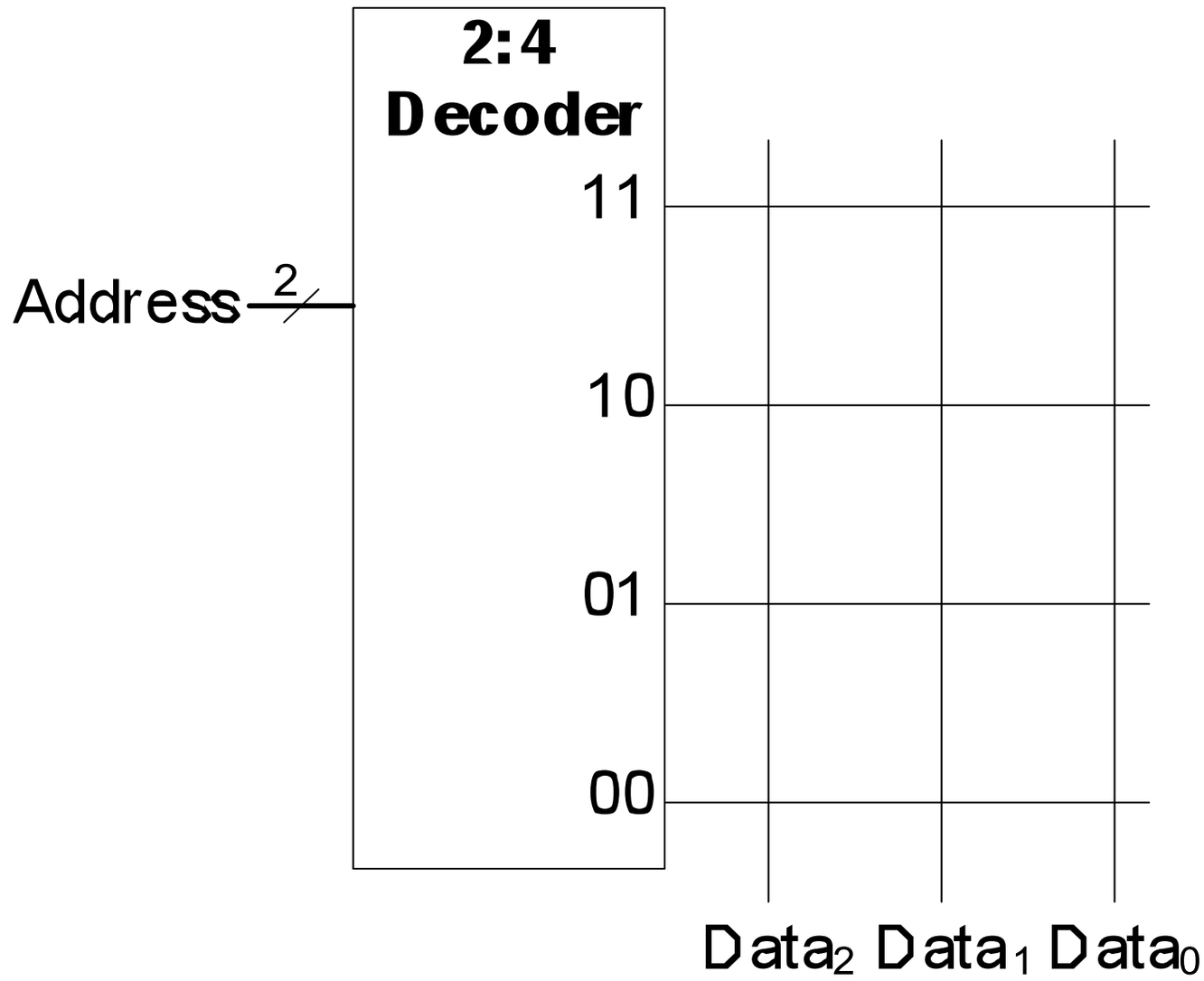
$Data_2 = A_1 \oplus A_0$

$Data_1 = A_1 + A_0$

$Data_0 = A_1 A_0$

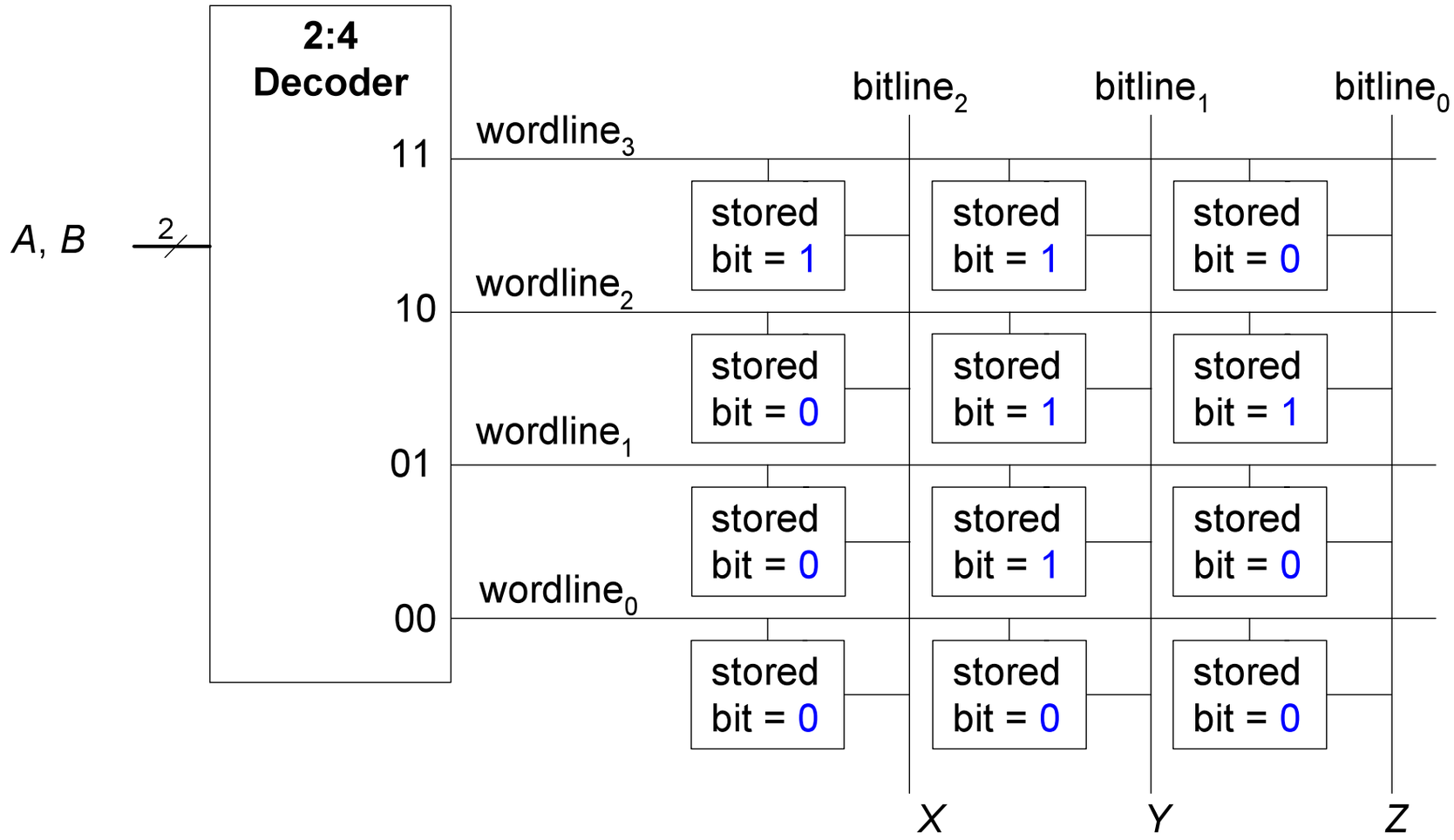
Example: Logic with ROMs

- Implement the following logic functions using a 2^2 × 3-bit ROM:
  - X = AB
  - Y = A + B
  - Z = AB



Logic with Any Memory Array

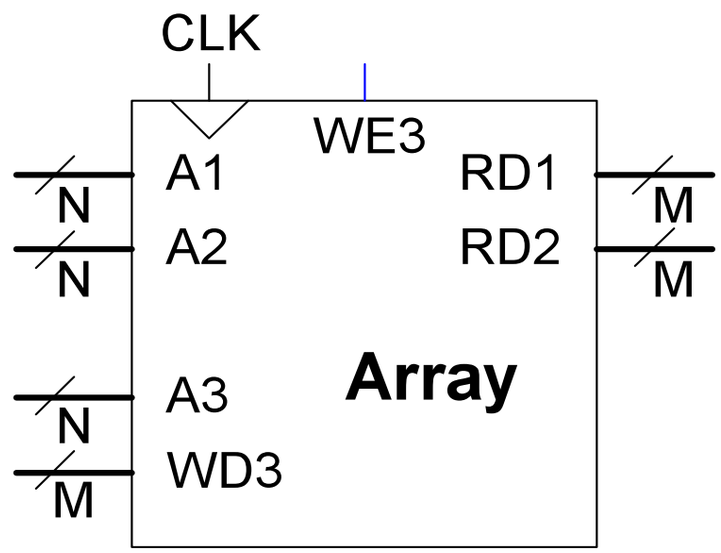
- X = AB
- Y = A + B
- Z = AB



Called lookup tables (LUTs): look up output at each input combination (address)

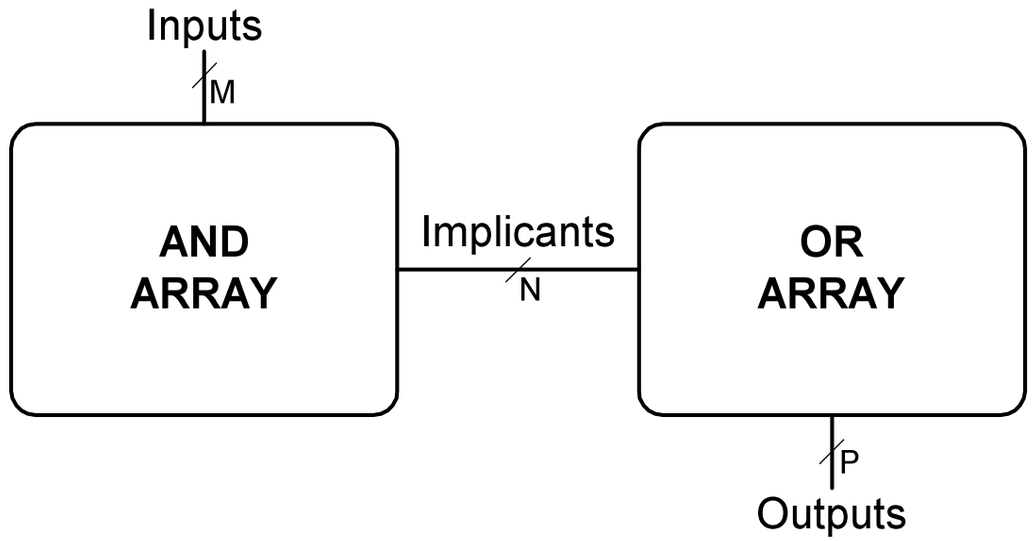
Multi-ported Memories

- Port: address/data pair
- 3-ported memory
  - 2 read ports (A1/RD1, A2/RD2)
  - 1 write port (A3/WD3, WE3 enables writing)
- Small multi-ported memories are called register files

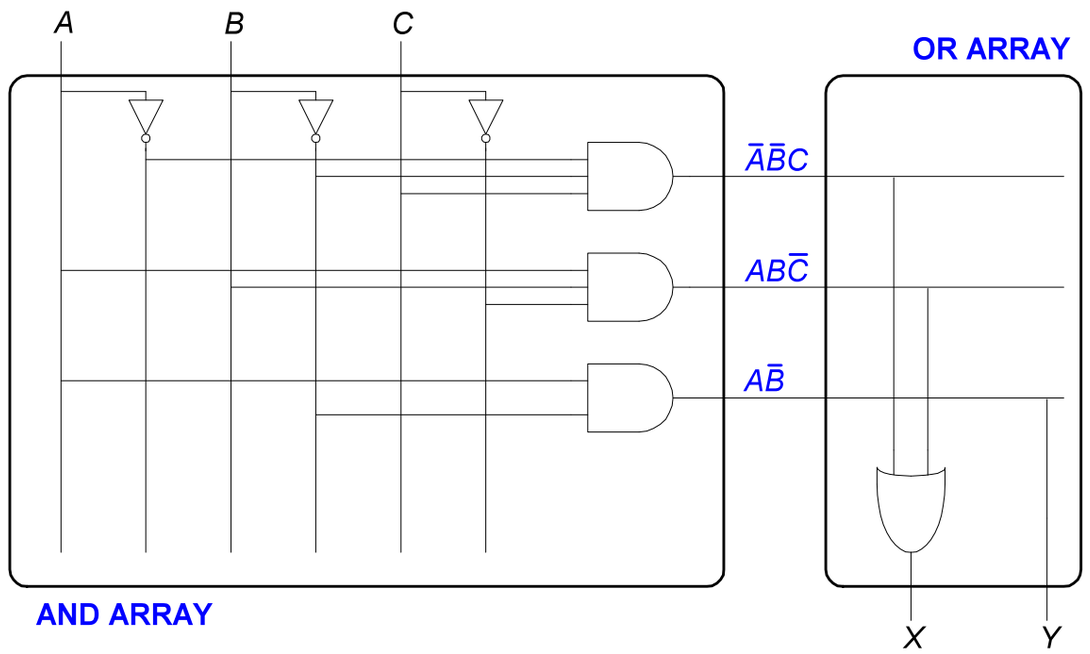


Logic Arrays: PLAs

- Programmable logic arrays (PLAs)
  - AND array followed by OR array
  - Perform combinational logic only
  - Fixed internal connections

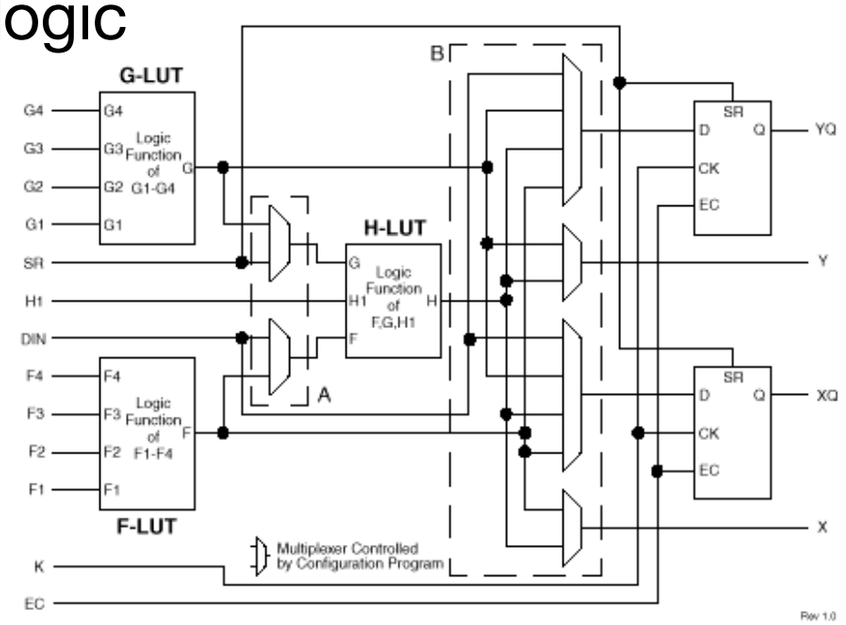


- Example
  - X = `A`BC + AB`C
  - Y = A`B



Logic Arrays: FPGAs

- FPGAs are composed of:
  - **CLBs (Configurable logic blocks)**: perform logic, are composed of:
    - LUTs (lookup tables): perform combinational logic
    - Flip-flops: perform sequential functions
    - Multiplexers: connect LUTs and flip-flops



- **IOBs (Input/output buffers)**: interface with outside world
- **Programmable interconnection**: connect CLBs and IOBs
- Some FPGAs include other building blocks such as multipliers and RAMs