

CSEE 3827: Fundamentals of Computer Systems

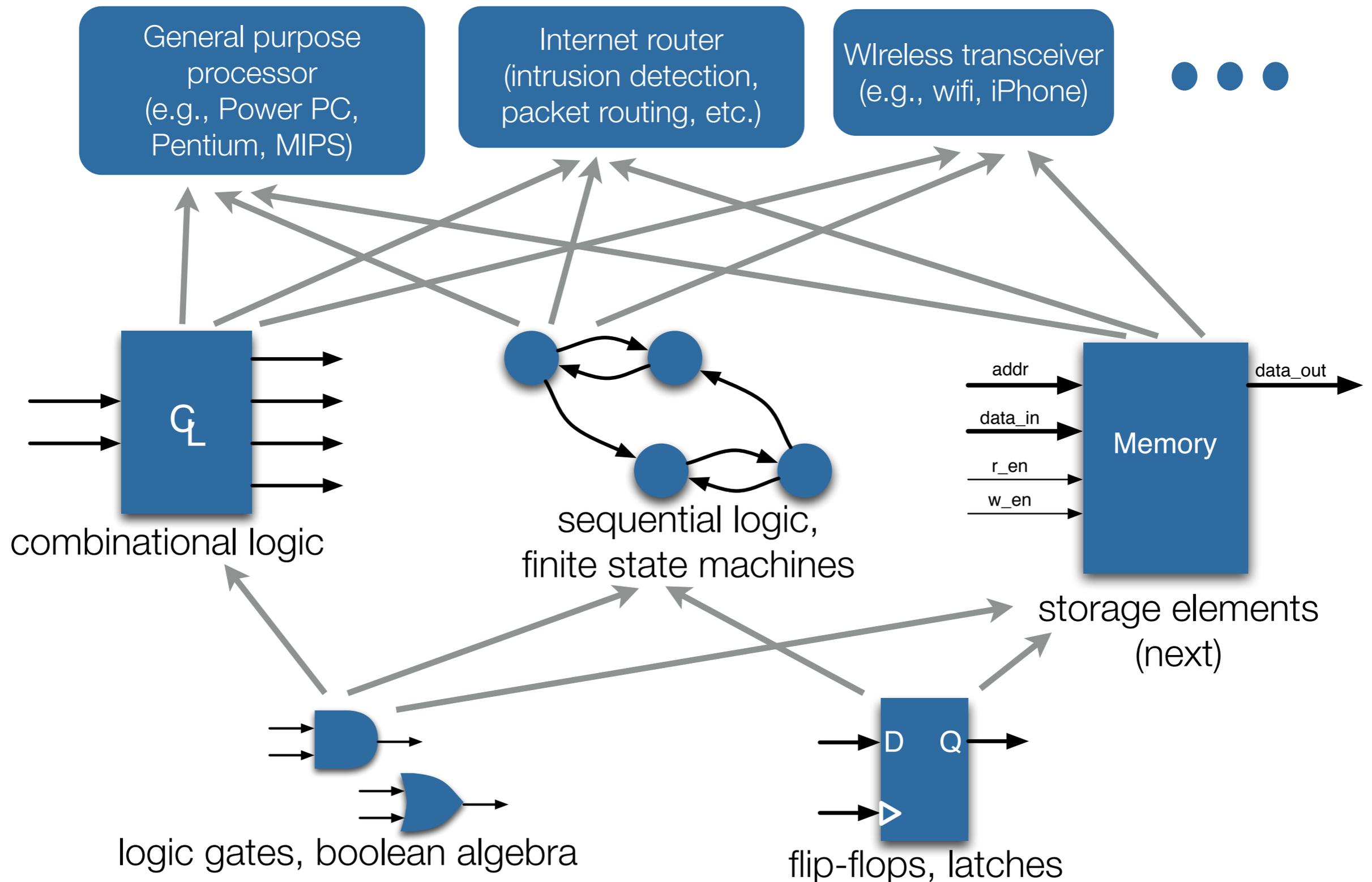
Lecture 13 & 14

March 25 & 30, 2009

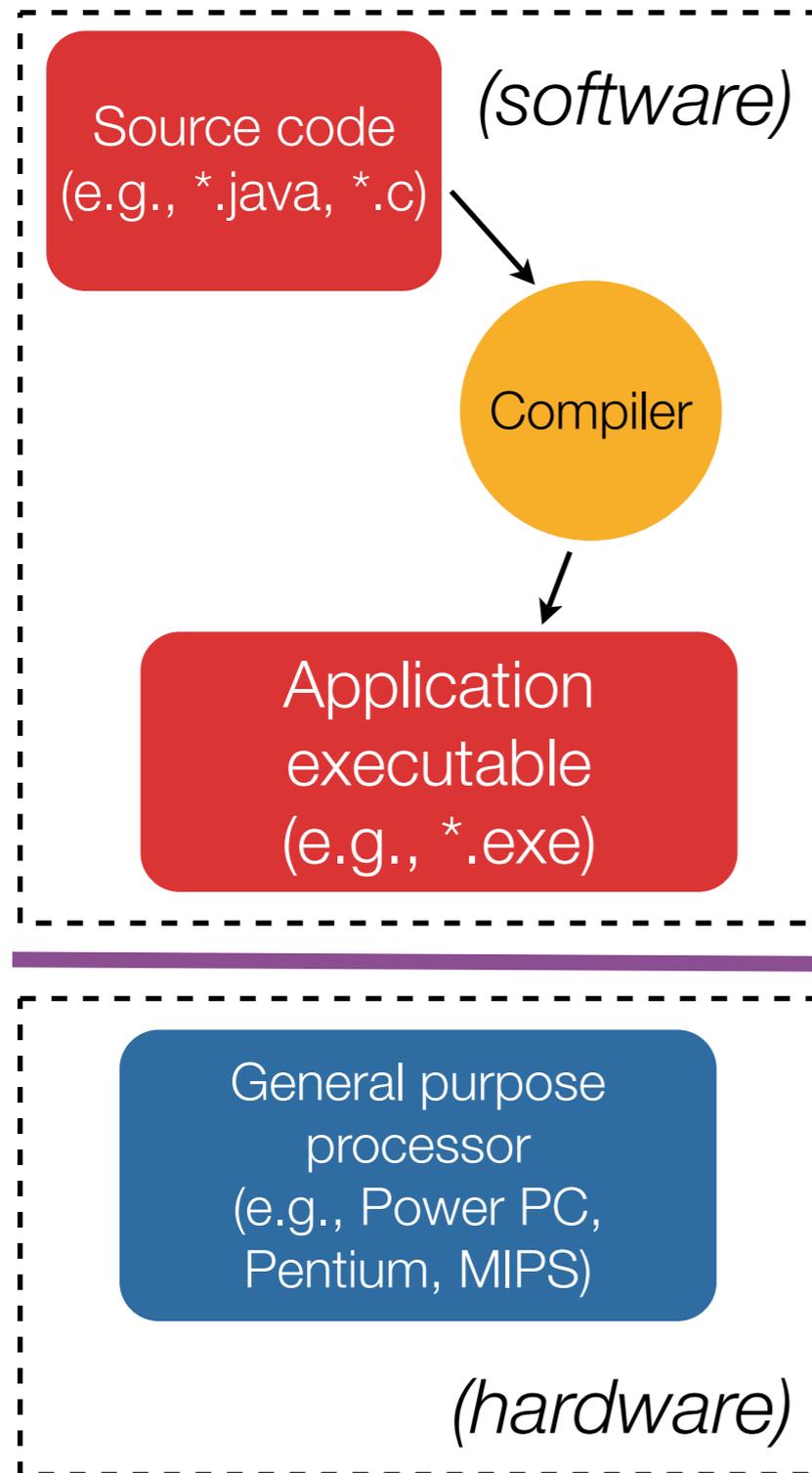
Martha Kim

martha@cs.columbia.edu

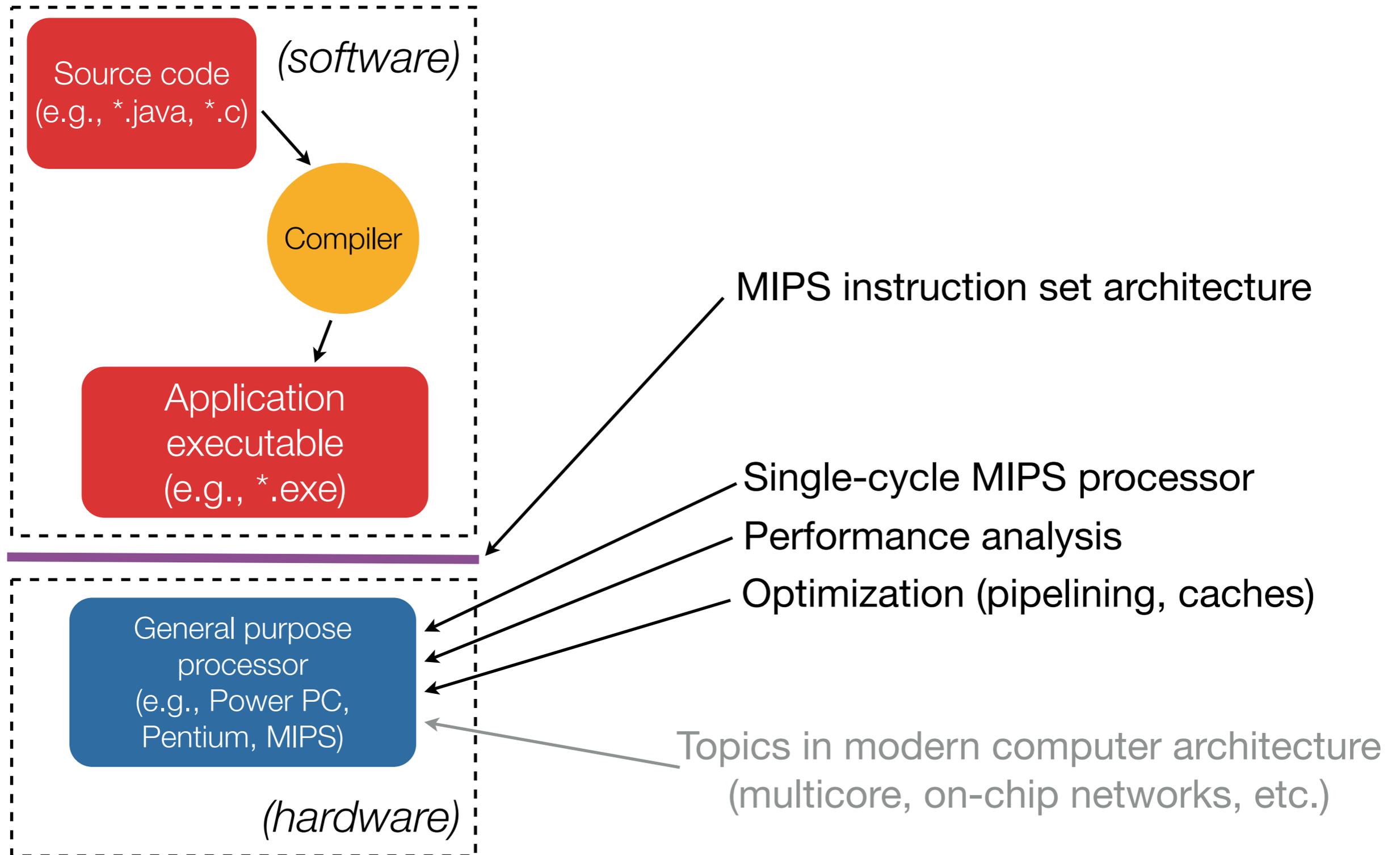
The big picture



The rest of the picture



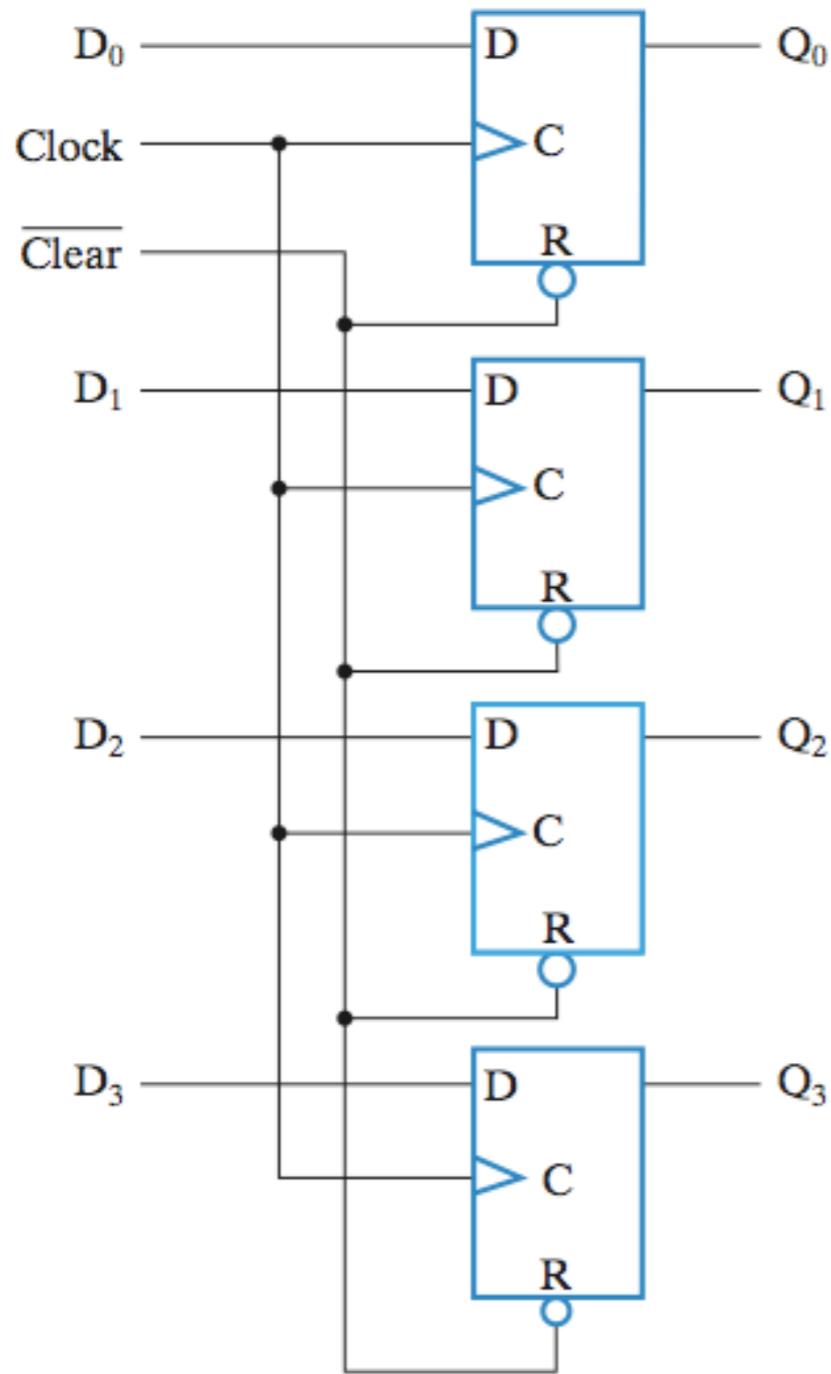
... and the rest of the semester



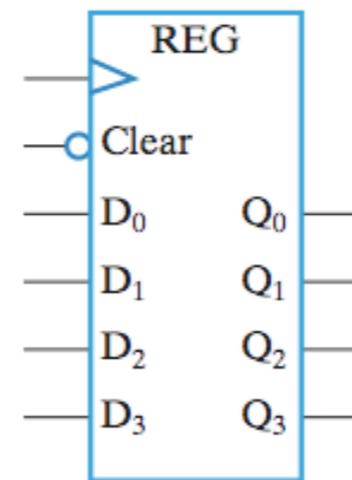
Registers (M&K 7.1)

Registers

A flip-flop can store 1 bit. A register is a set of n flip flops that stores n bits.



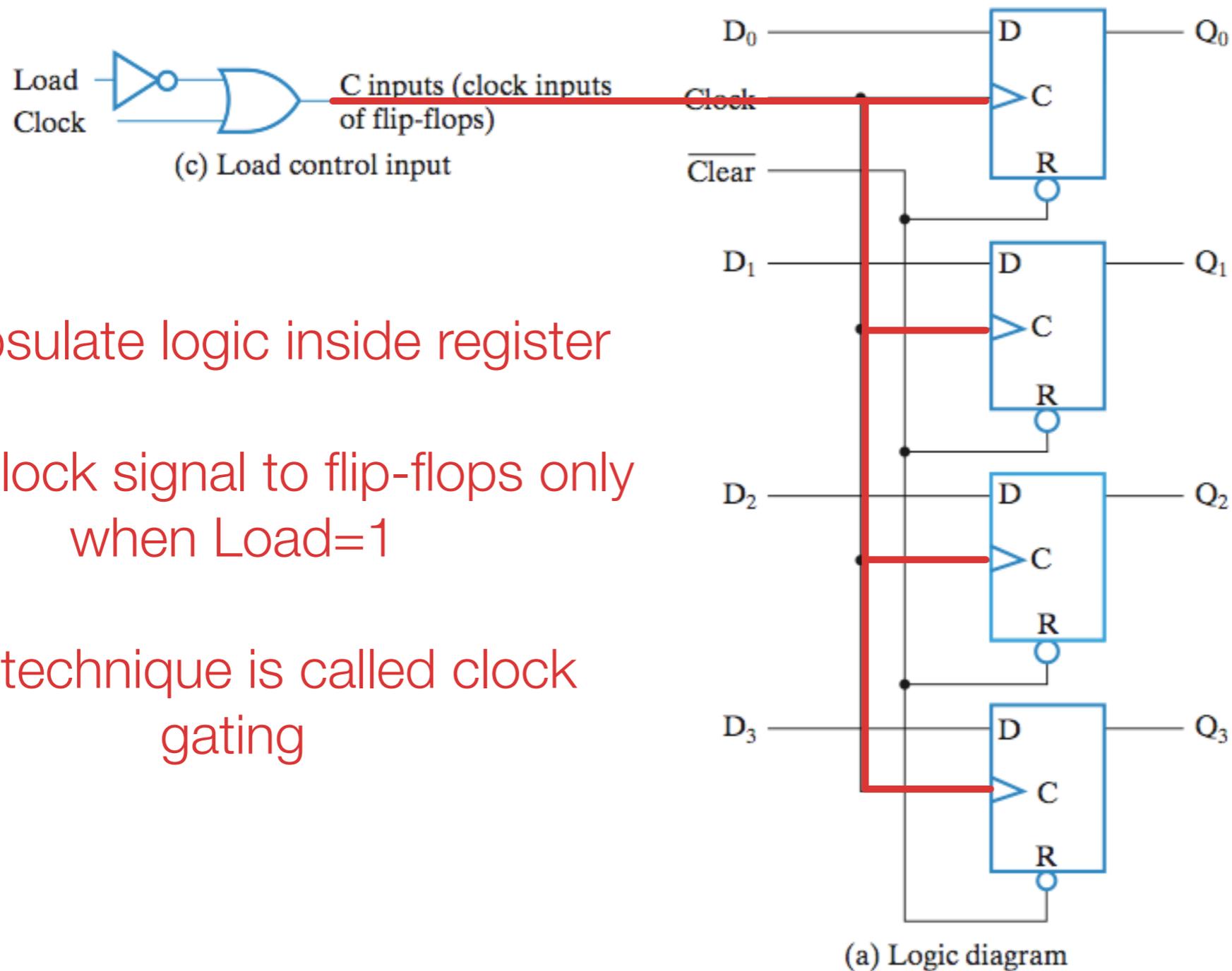
(a) Logic diagram



(b) Symbol

© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

Register w. load control input (v1)



Encapsulate logic inside register

Send clock signal to flip-flops only when Load=1

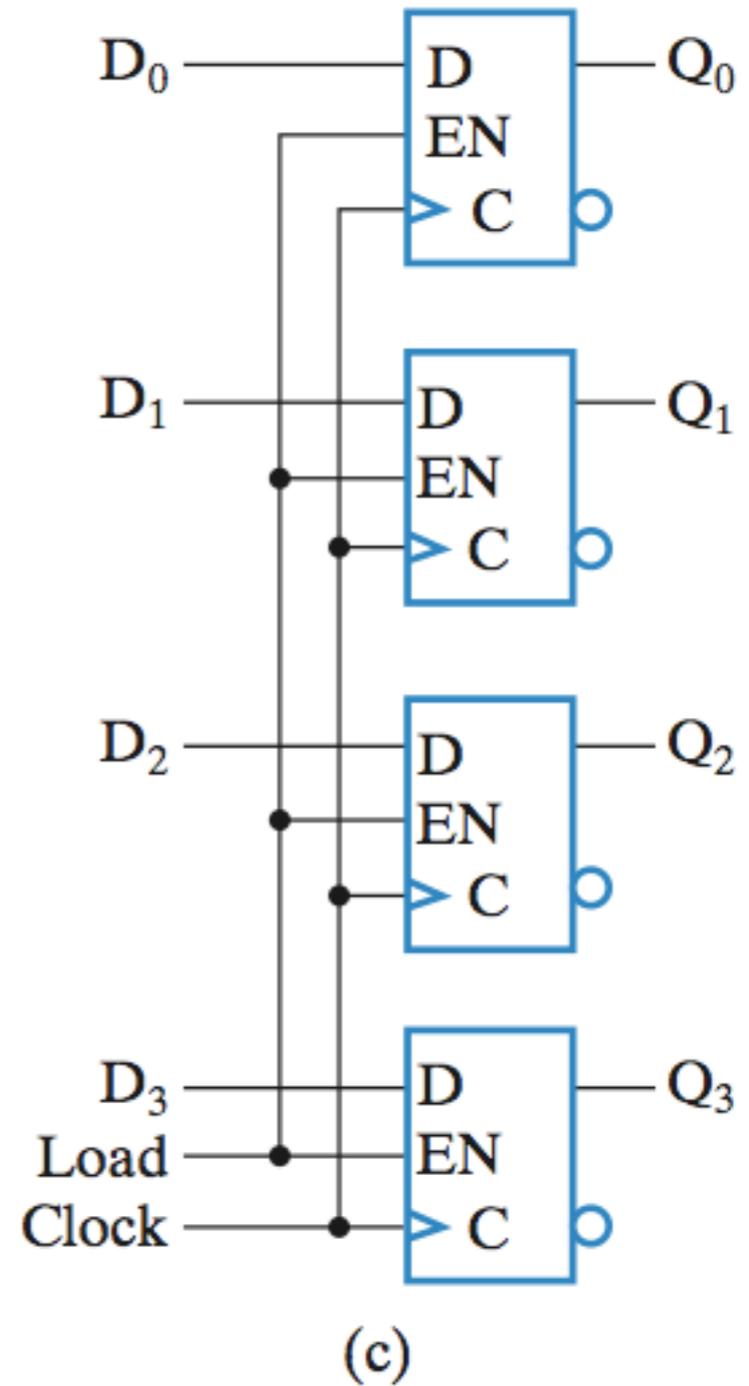
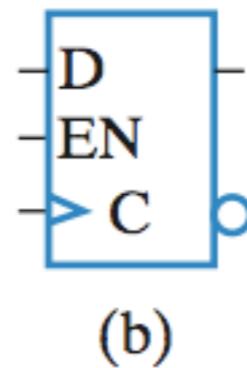
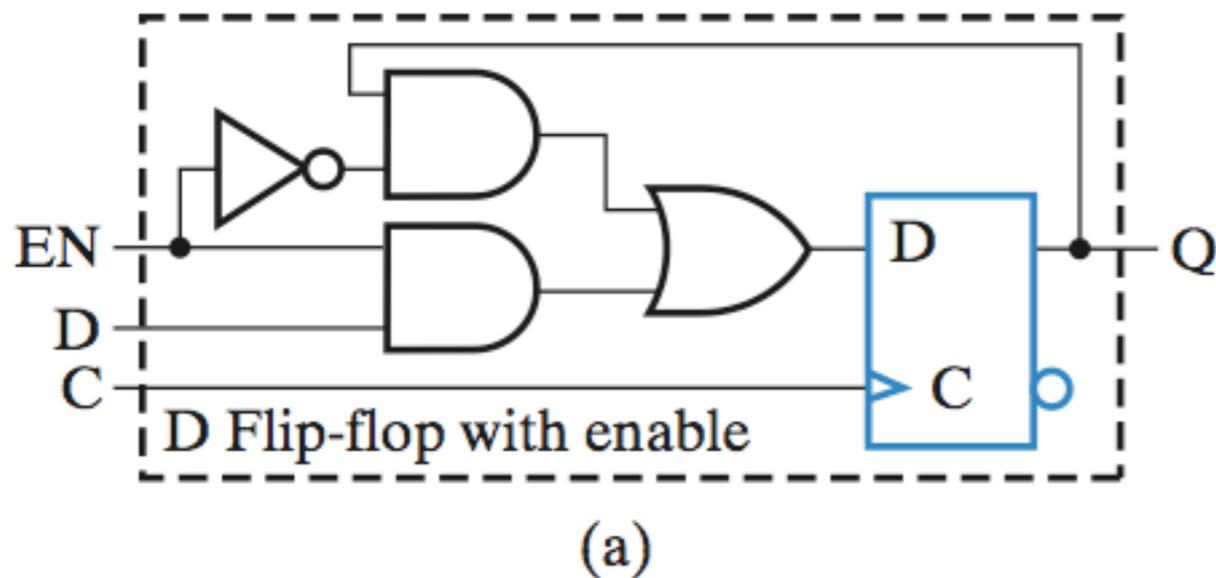
This technique is called clock gating

Register w. load control input (v2)

Encapsulate logic inside flip-flop

EN signal selects between current value of register (Q) or new value (D)

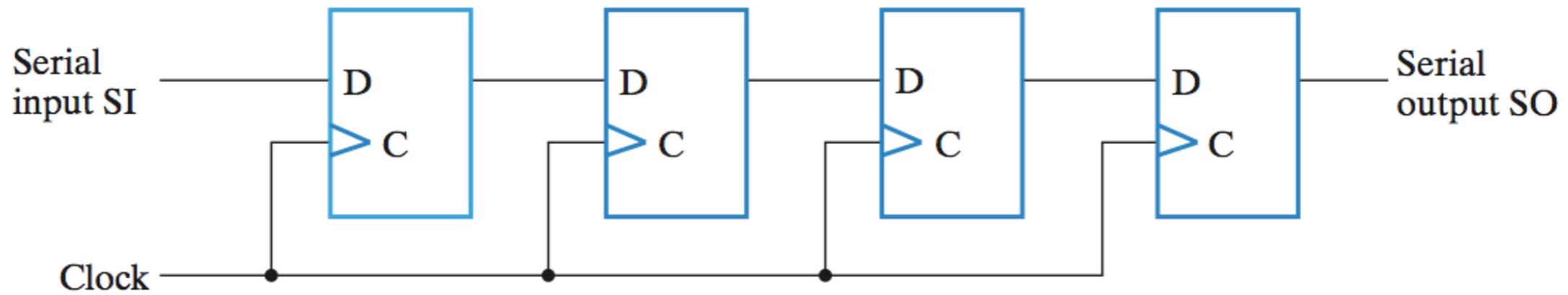
Preferable to v1 as it leaves clock signal unadulterated.



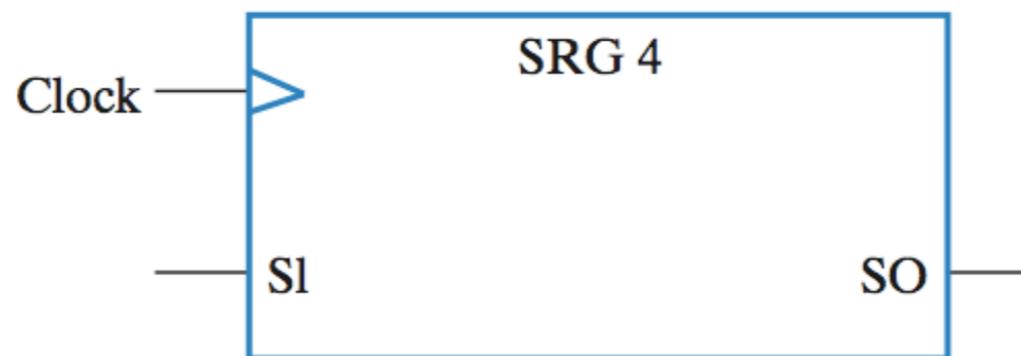
© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

Shift register

A register capable of shifting bits laterally in one or both directions



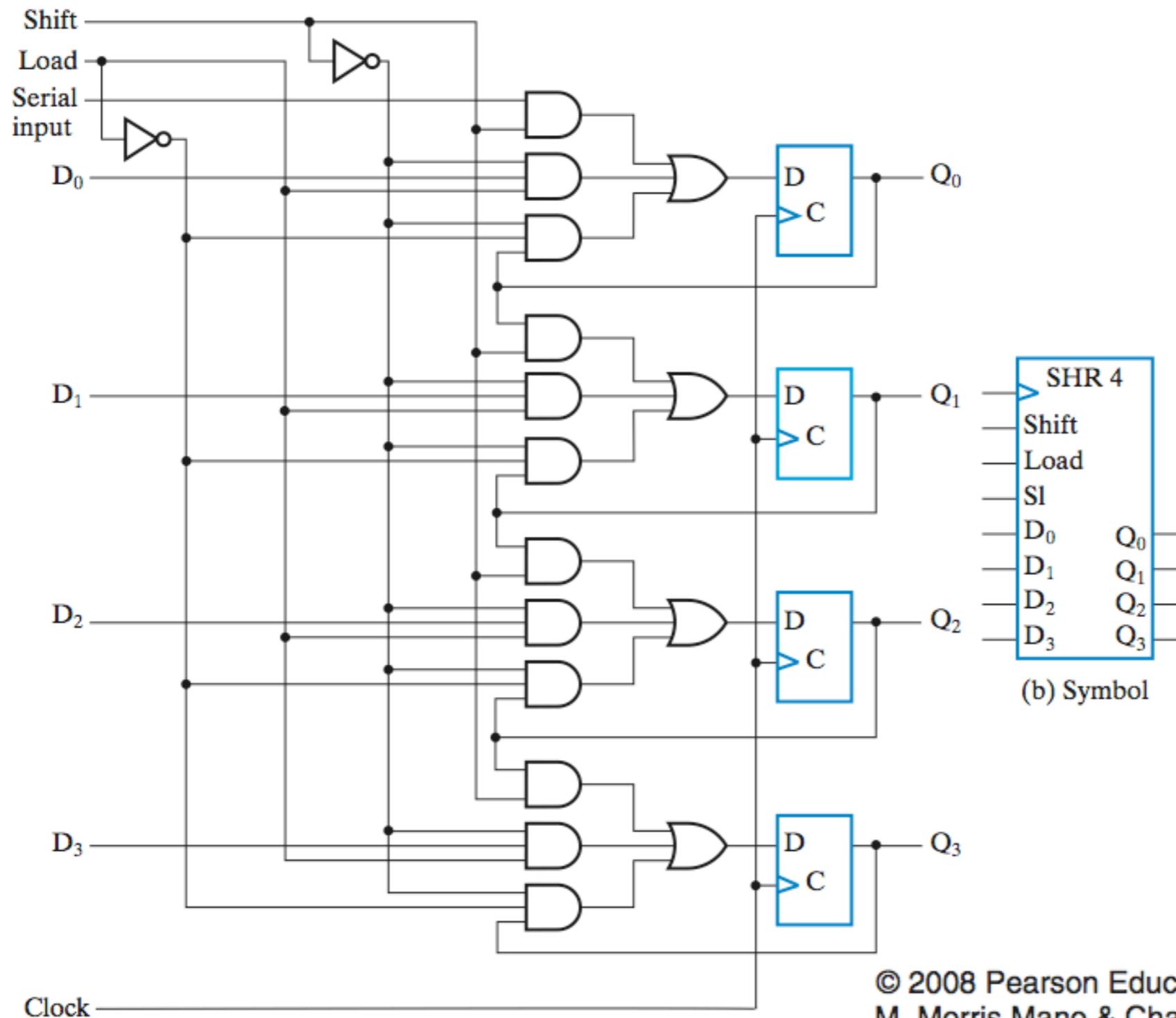
(a) Logic diagram



(b) Symbol

© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

Shift register w. parallel load

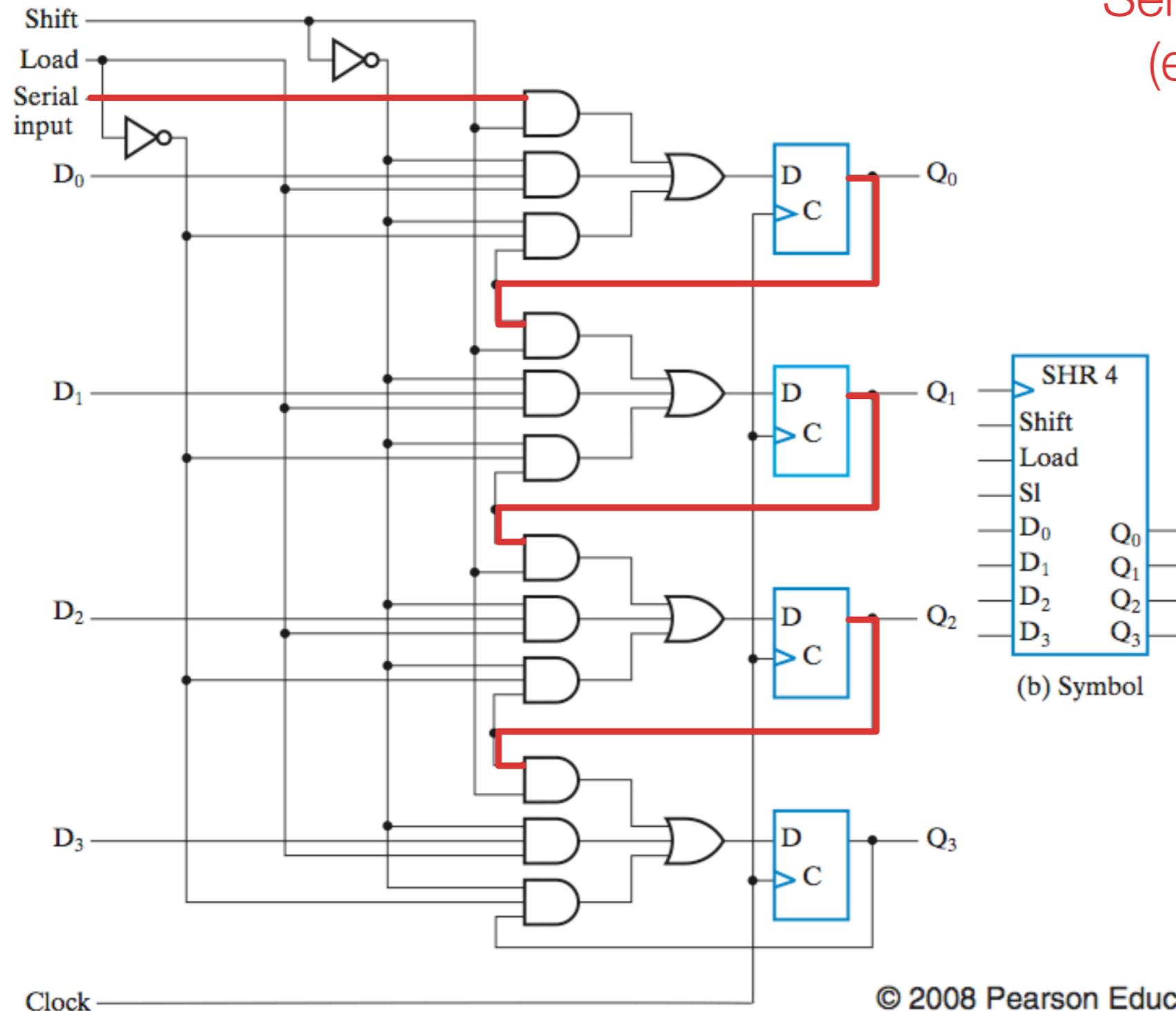


Three modes:
Serial input
Parallel input
No input

© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

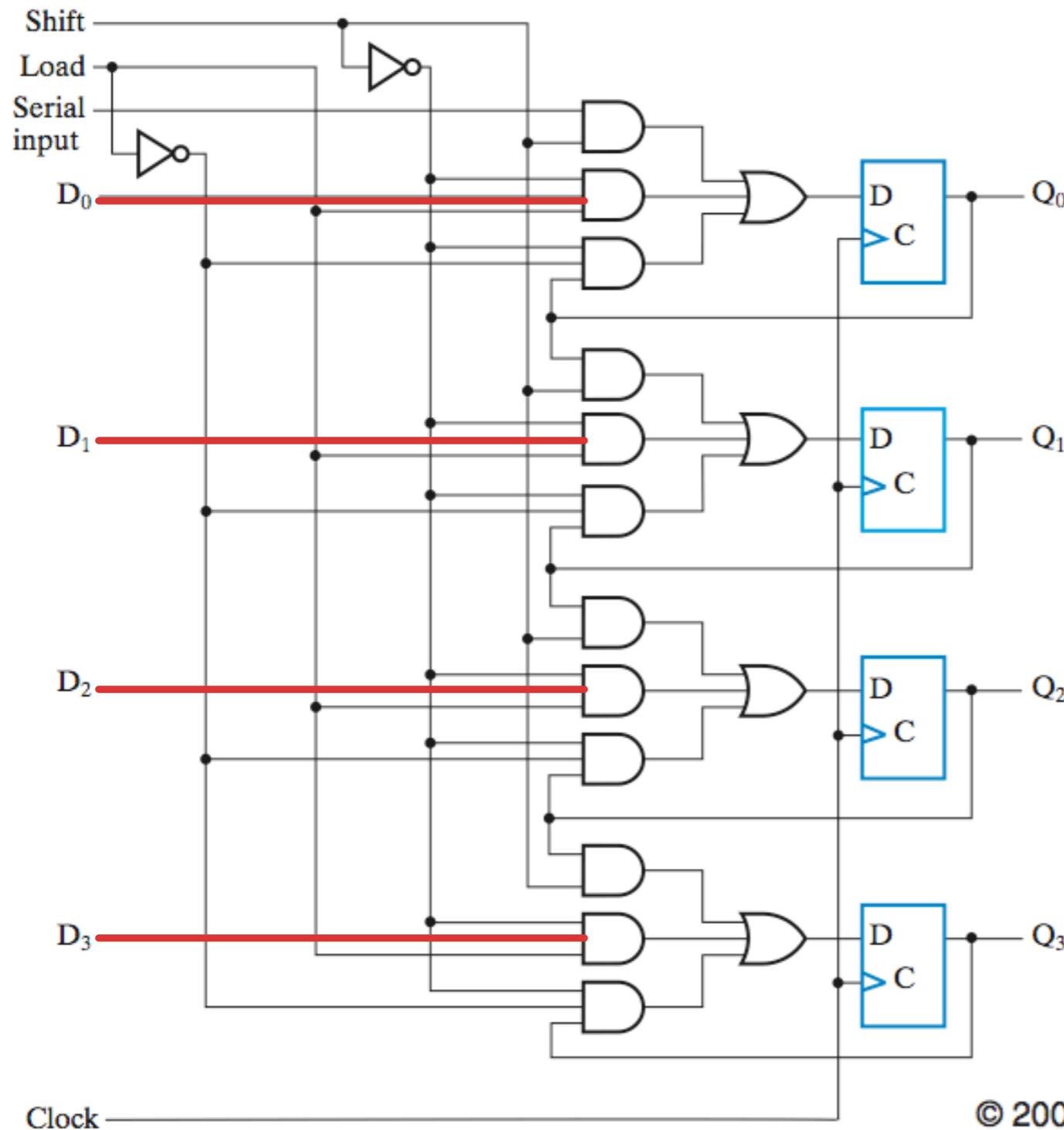
Shift register w. parallel load

Serial input operation
(enabled by Shift)

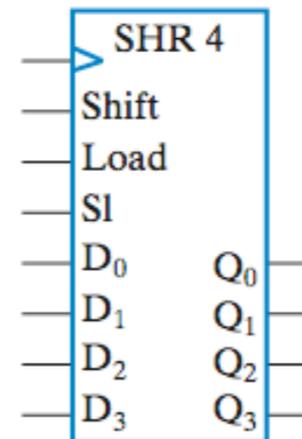


© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

Shift register w. parallel load



Parallel load operation
(enabled by Load AND $\overline{\text{Shift}}$)

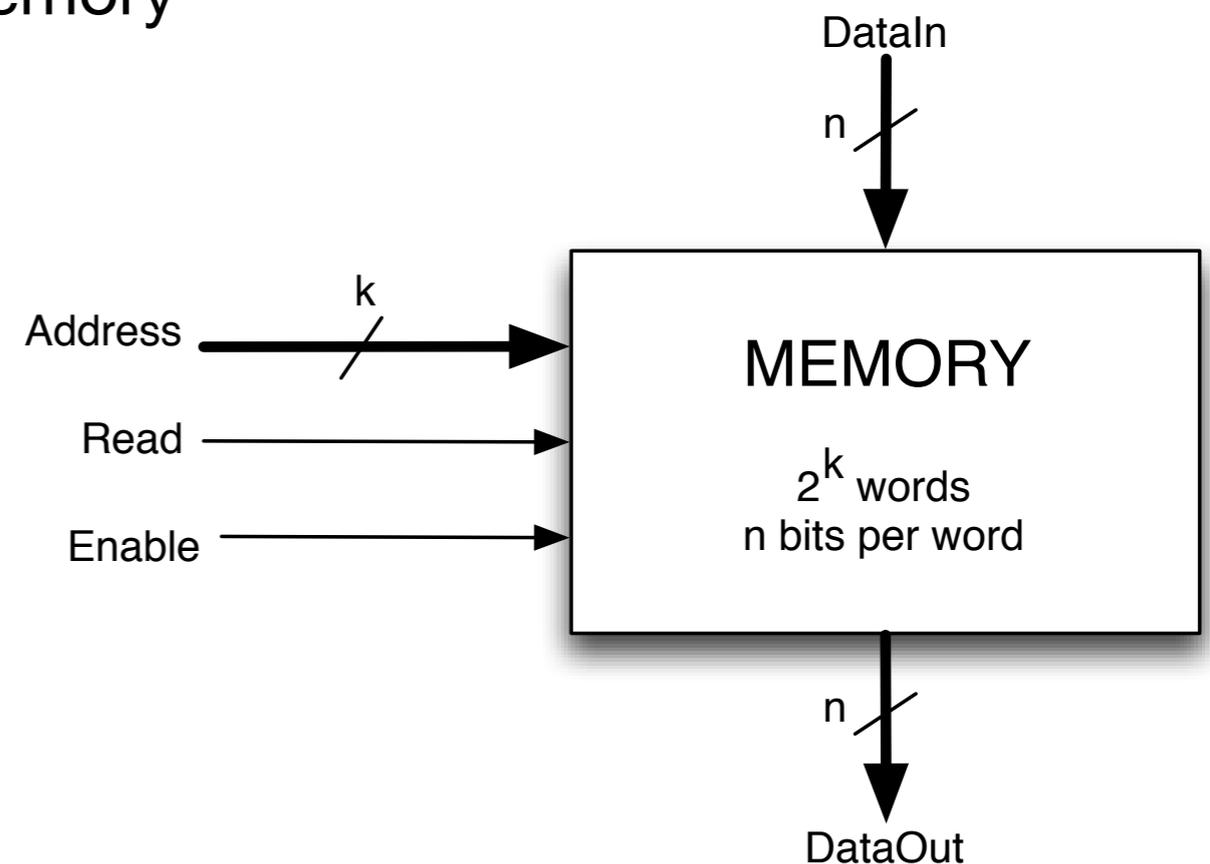


© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

Memory Arrays (M&K 8.1-8.7)

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

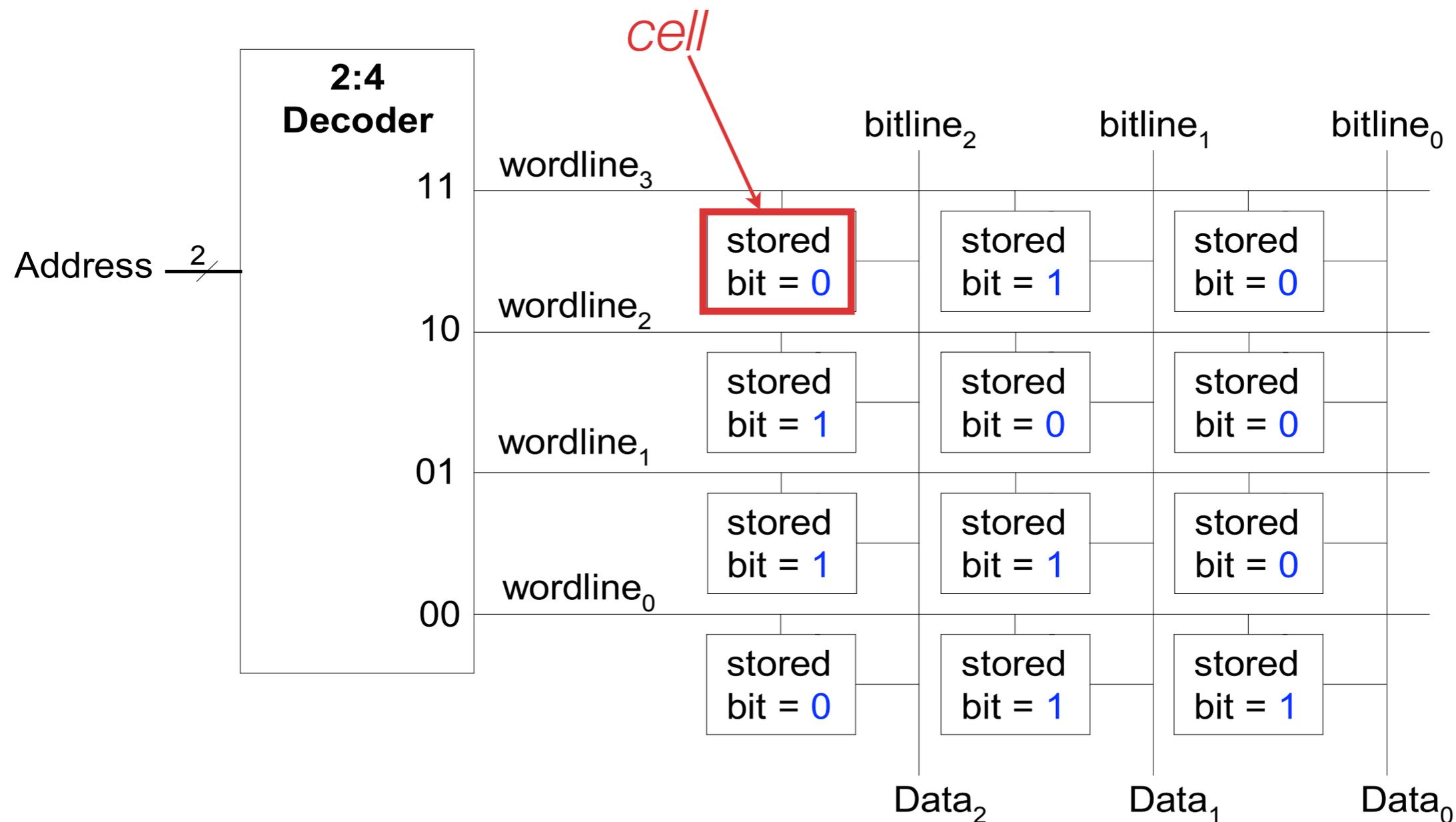
Conceptual view of memory

<u>Memory address</u>		<u>Memory contents</u>
<u>Binary</u>	<u>Decimal</u>	
0000000000	0	10110101 01011100
0000000001	1	10101011 10001001
0000000010	2	00001101 01000110
	.	.
	.	.
	.	.
	.	.
	.	.
1111111101	1021	10011101 00010101
1111111110	1022	00001101 00011110
1111111111	1023	11011110 00100100

© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

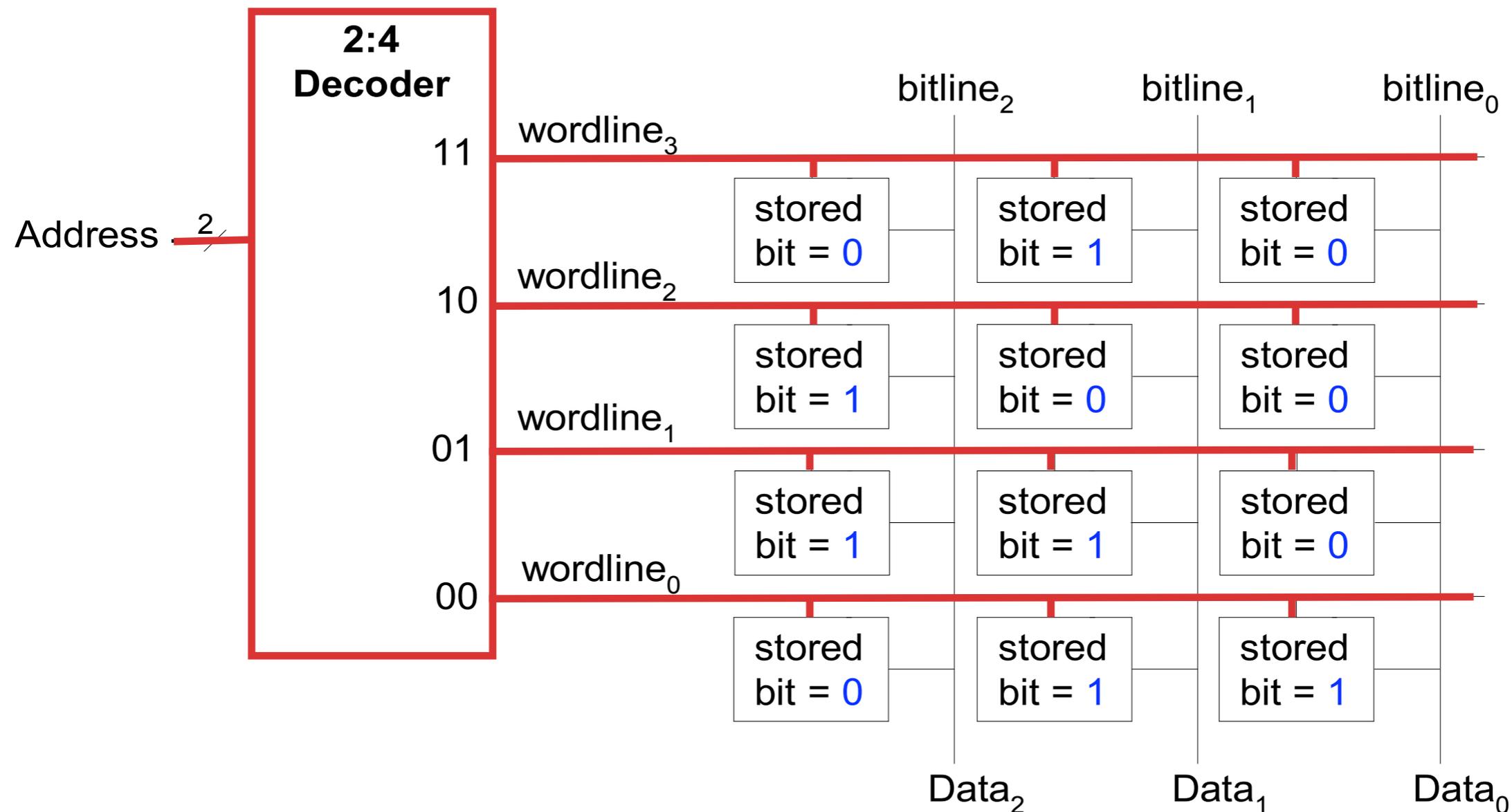
Memory array architecture (1)

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



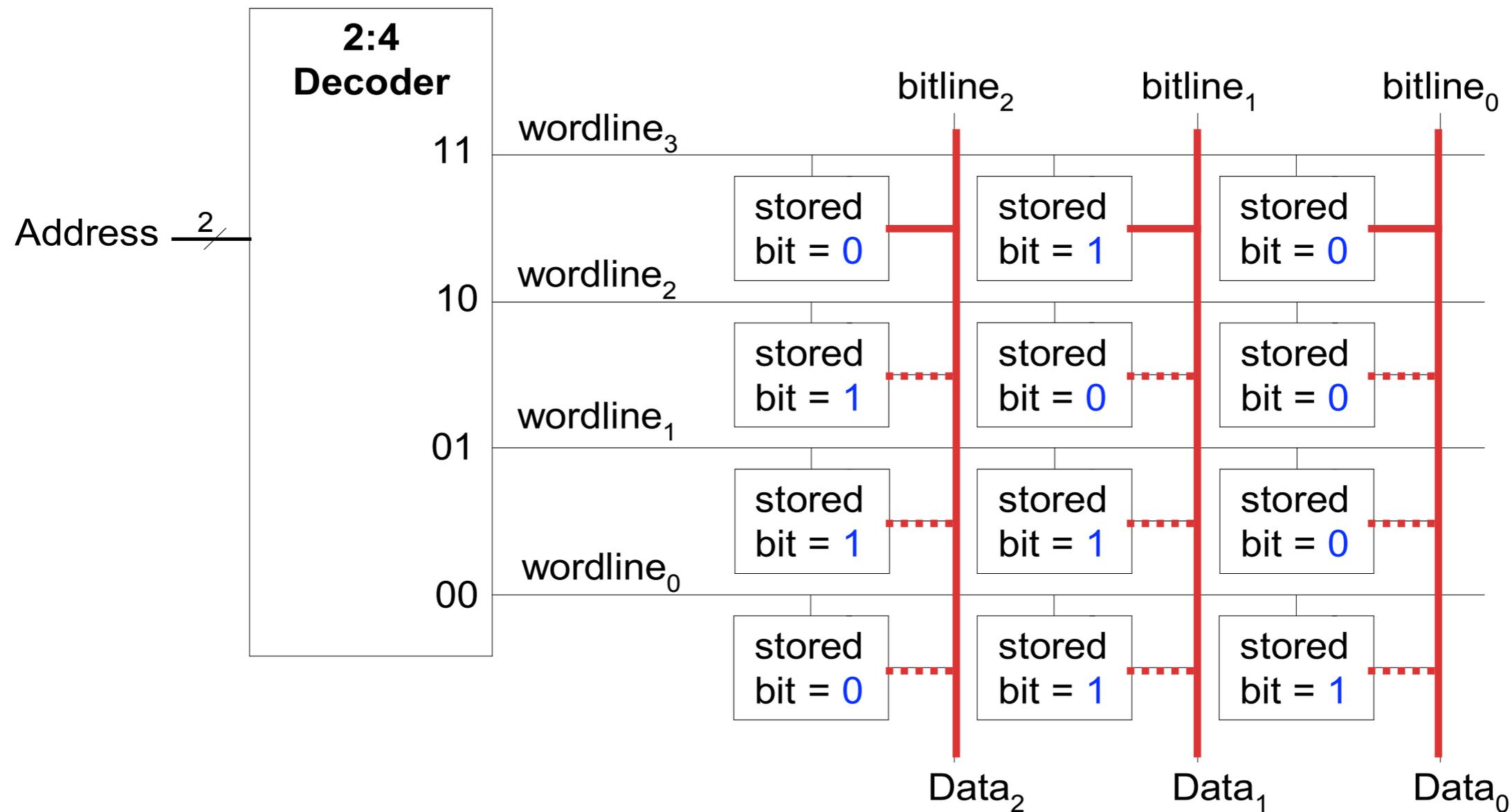
Memory array architecture (2)

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



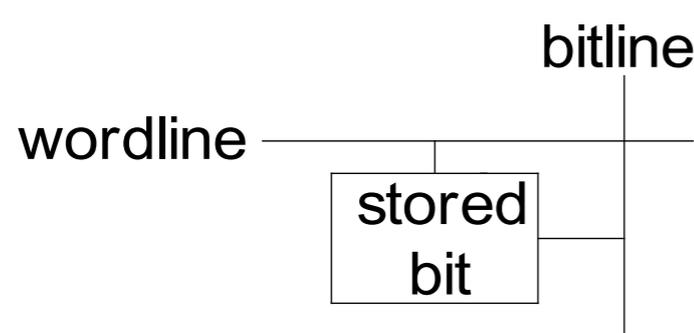
Memory array architecture (3)

When writing, 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)

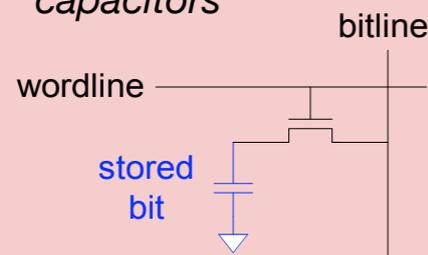
Volatile (no storage when power off)

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.

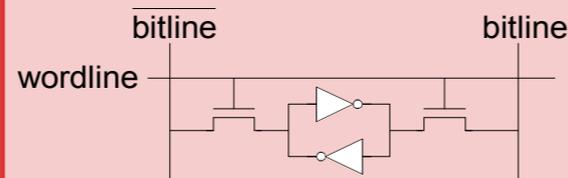
Dynamic RAM (DRAM)

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

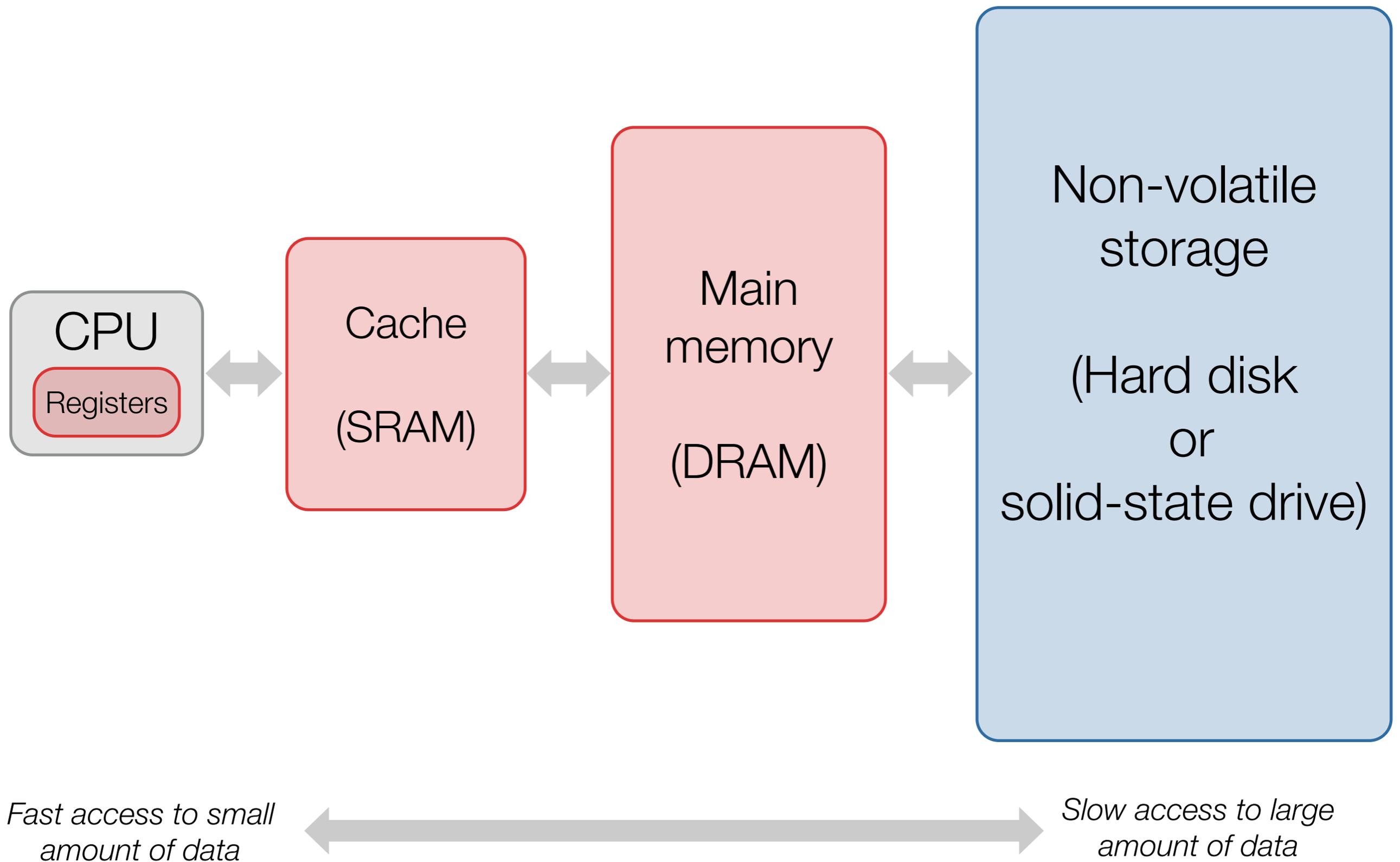
Electrically erasable floating gate with multiple erasure and programming modes

Hard Disk

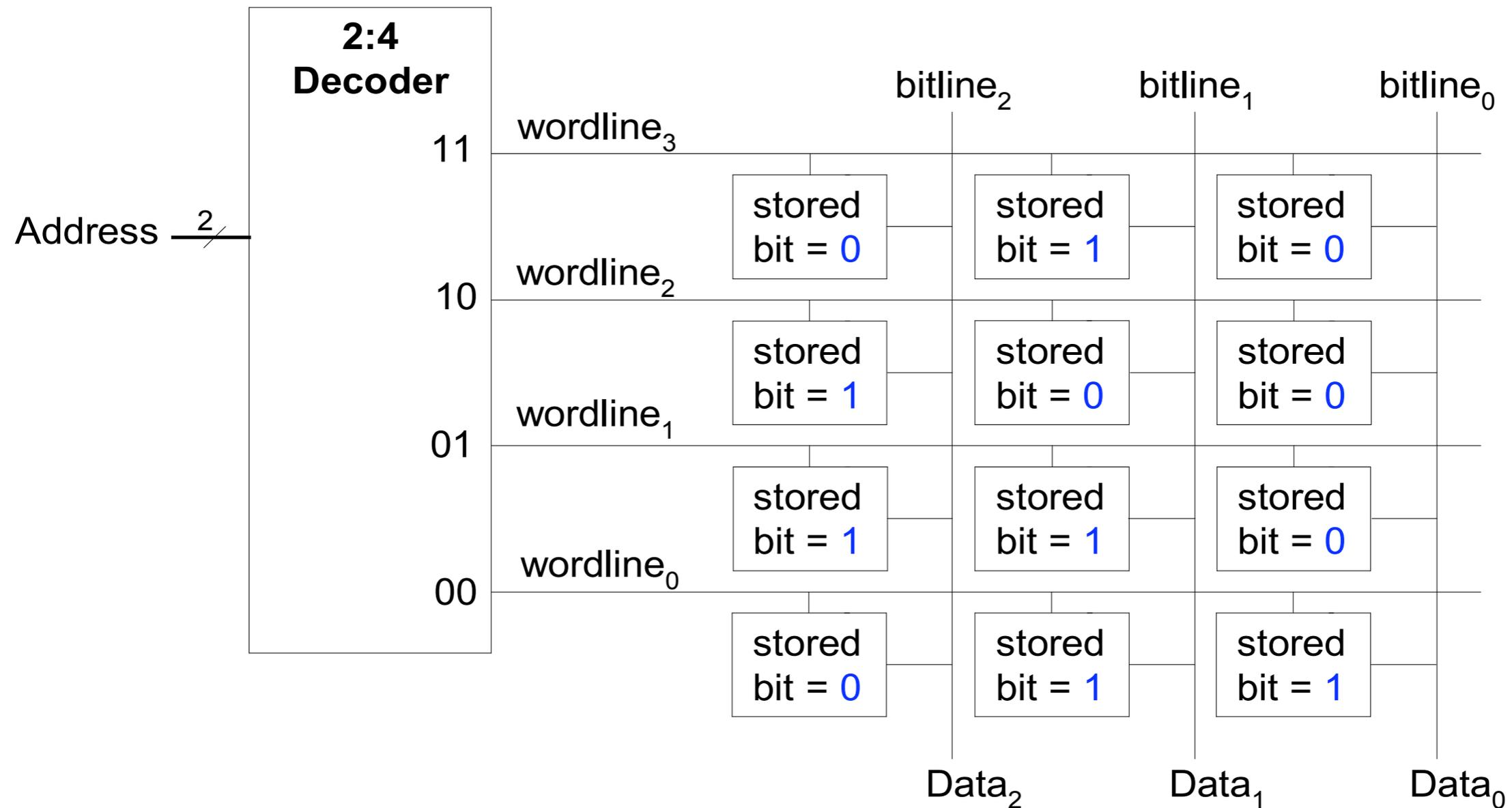
Volatile storage (RAM) comparisons

	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

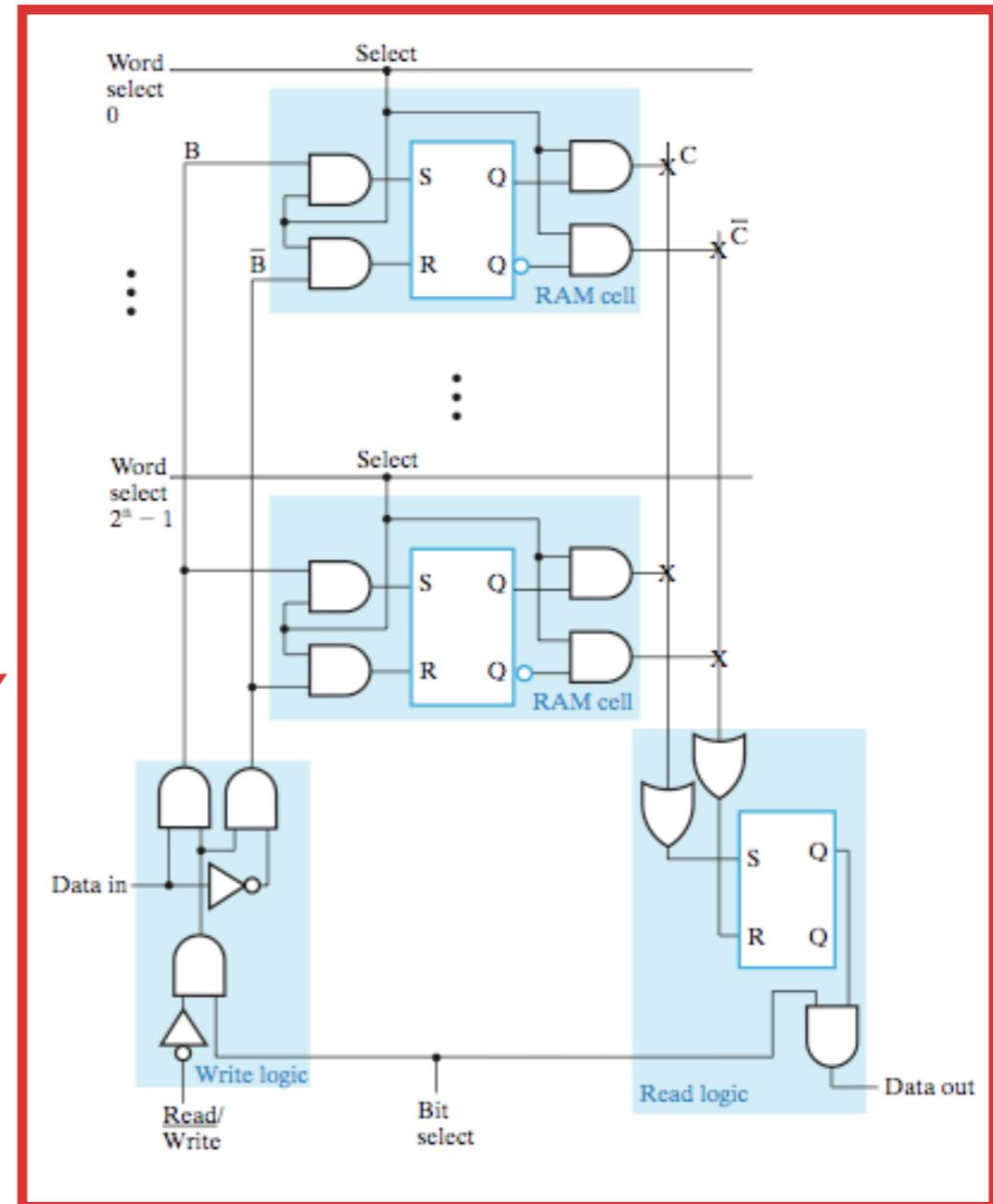
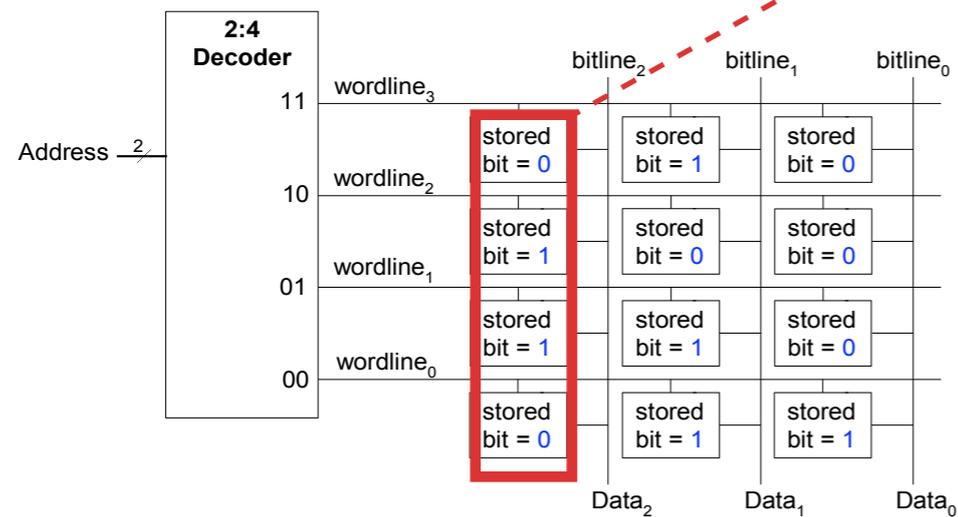


Bottom-up examination of SRAM circuits



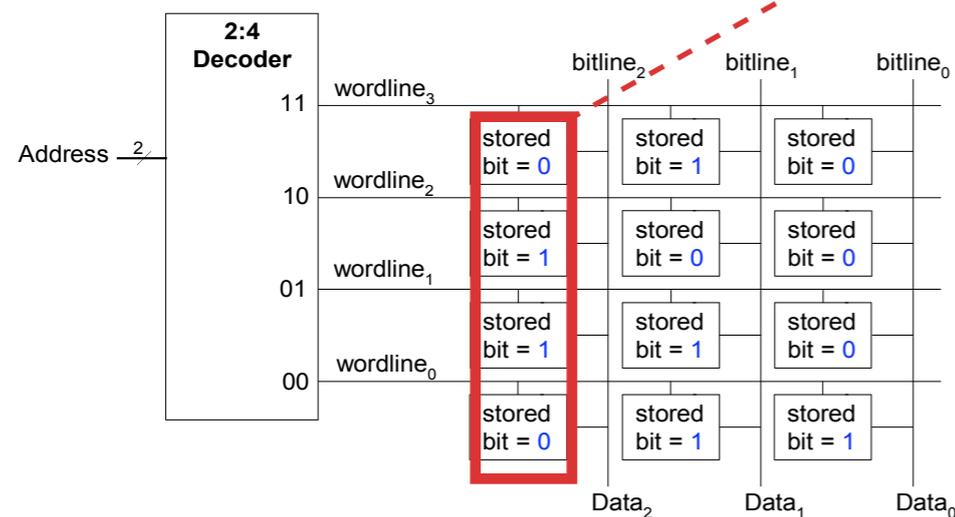
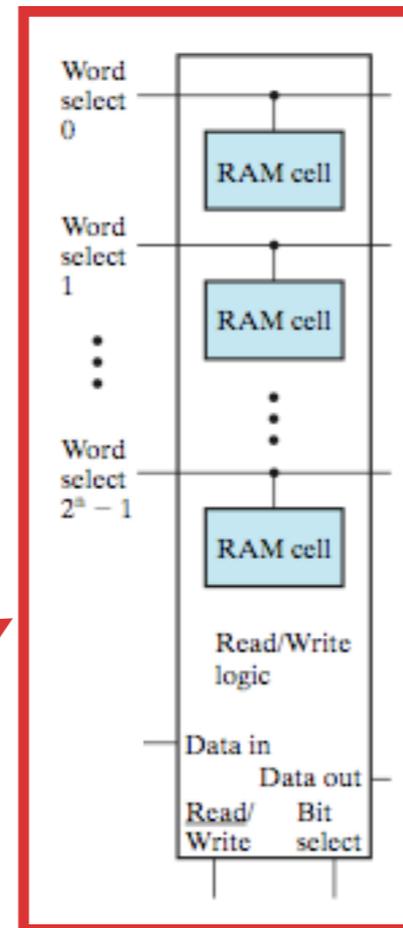
Bottom-up examination of SRAM circuits (3)

Cells wired into bitslices.



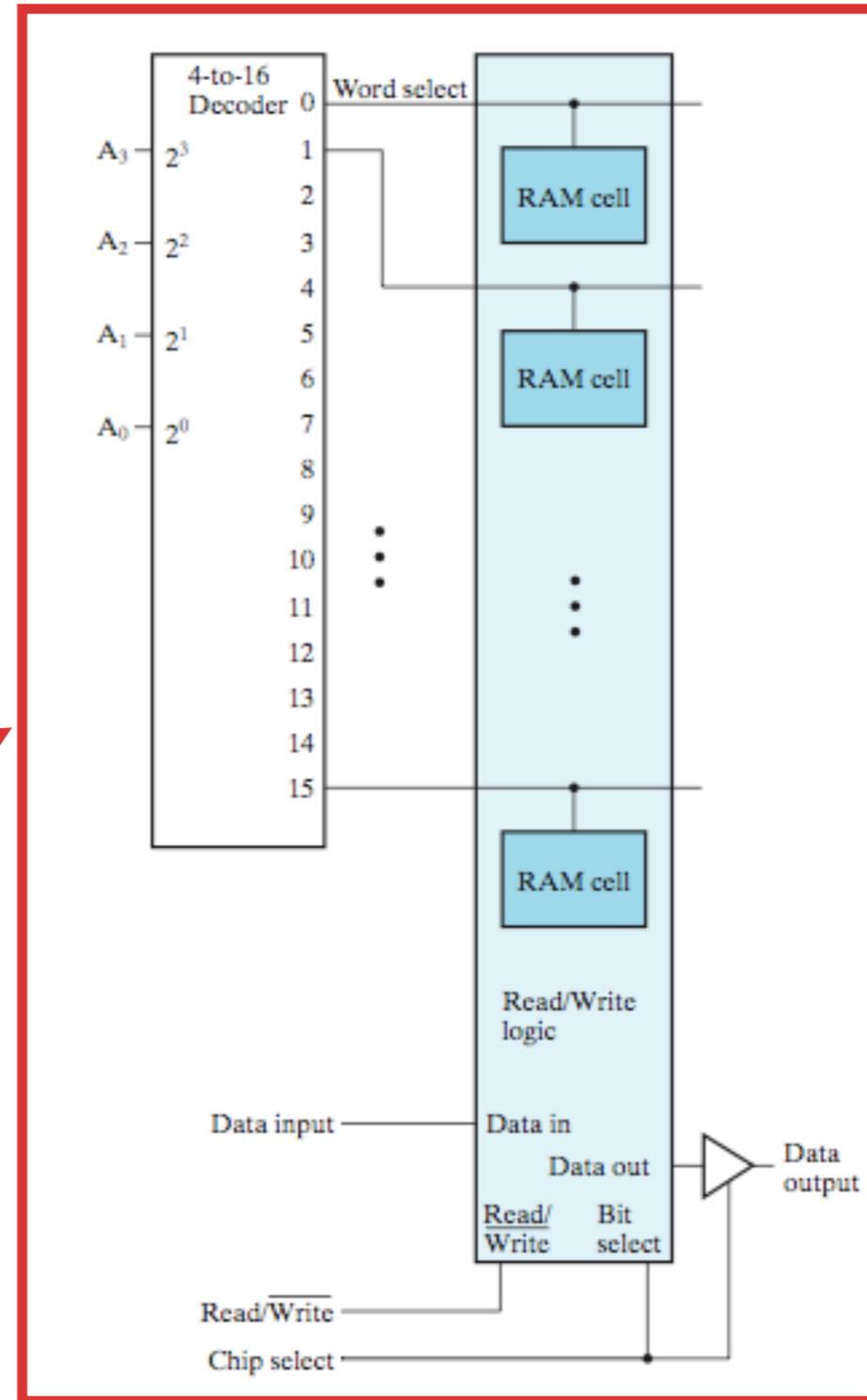
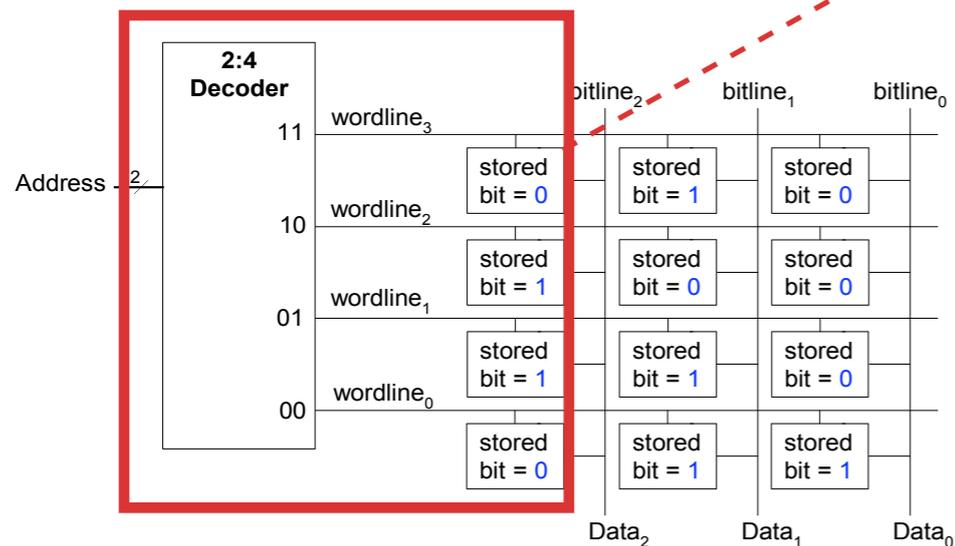
Bottom-up examination of SRAM circuits (4)

Block diagram of a bitslice



Bottom-up examination of SRAM circuits (5)

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

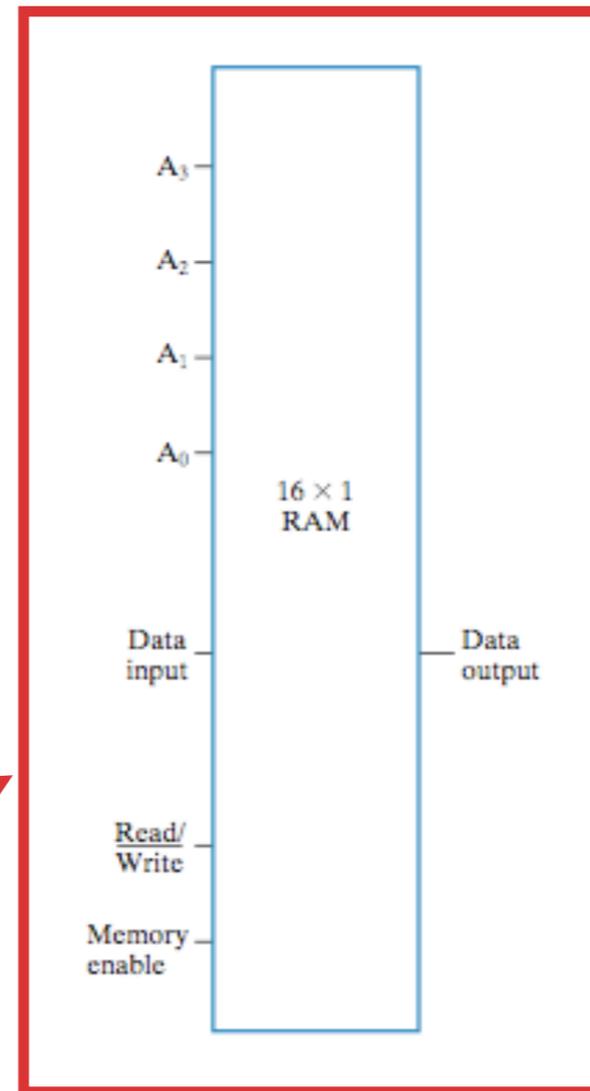
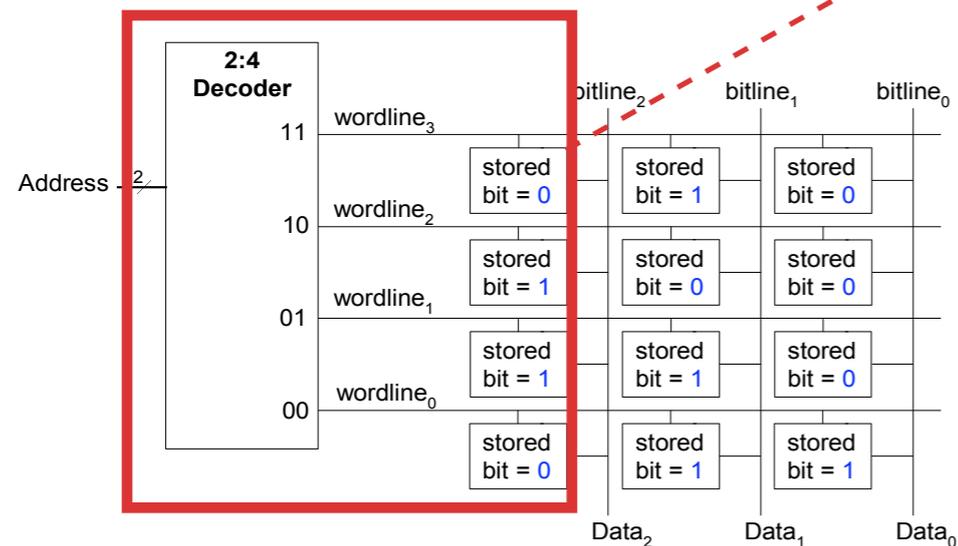


© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

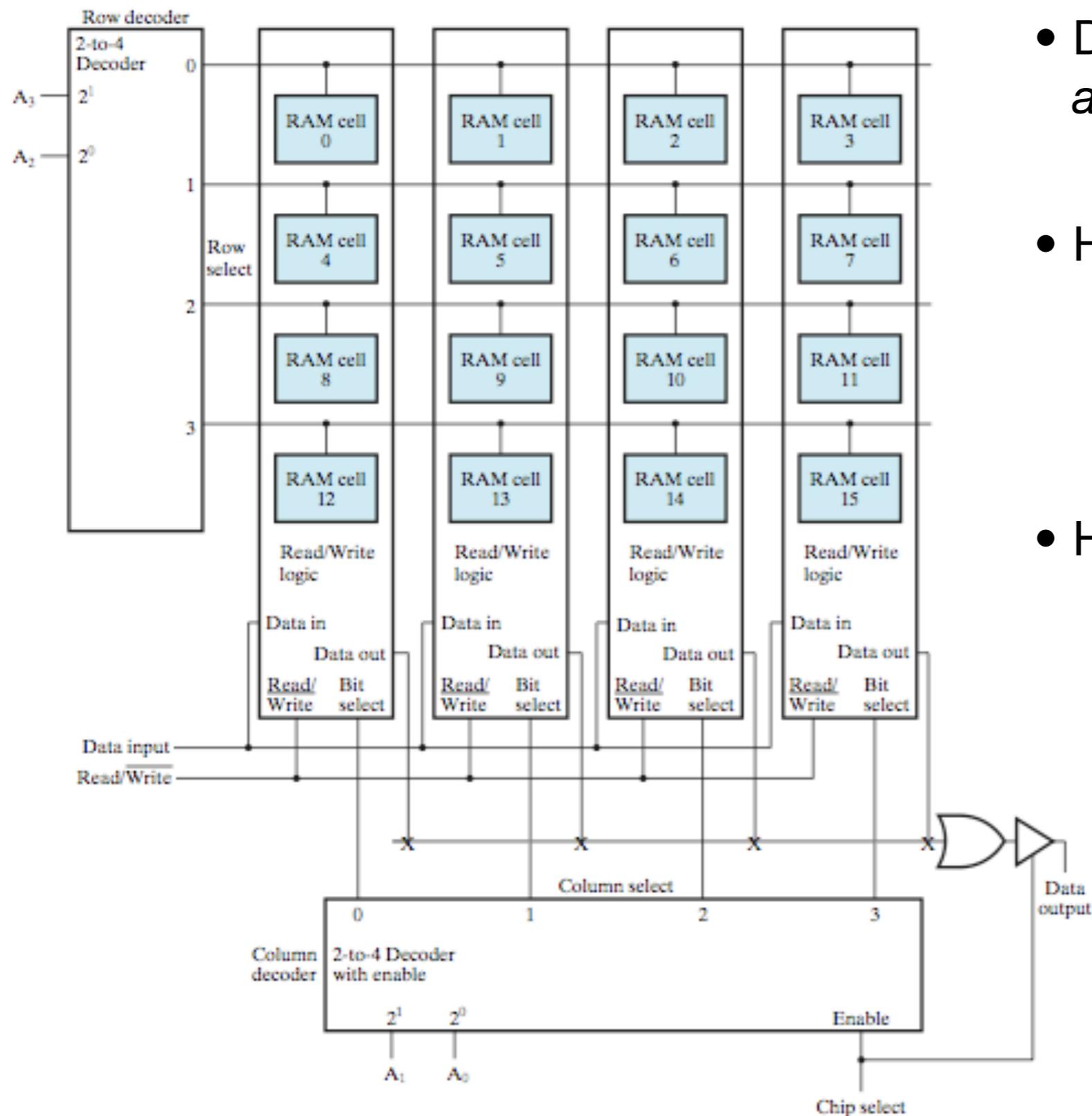
Copyright © 2007 Elsevier

Bottom-up examination of SRAM circuits (6)

*To increase word size,
add bitslices.*

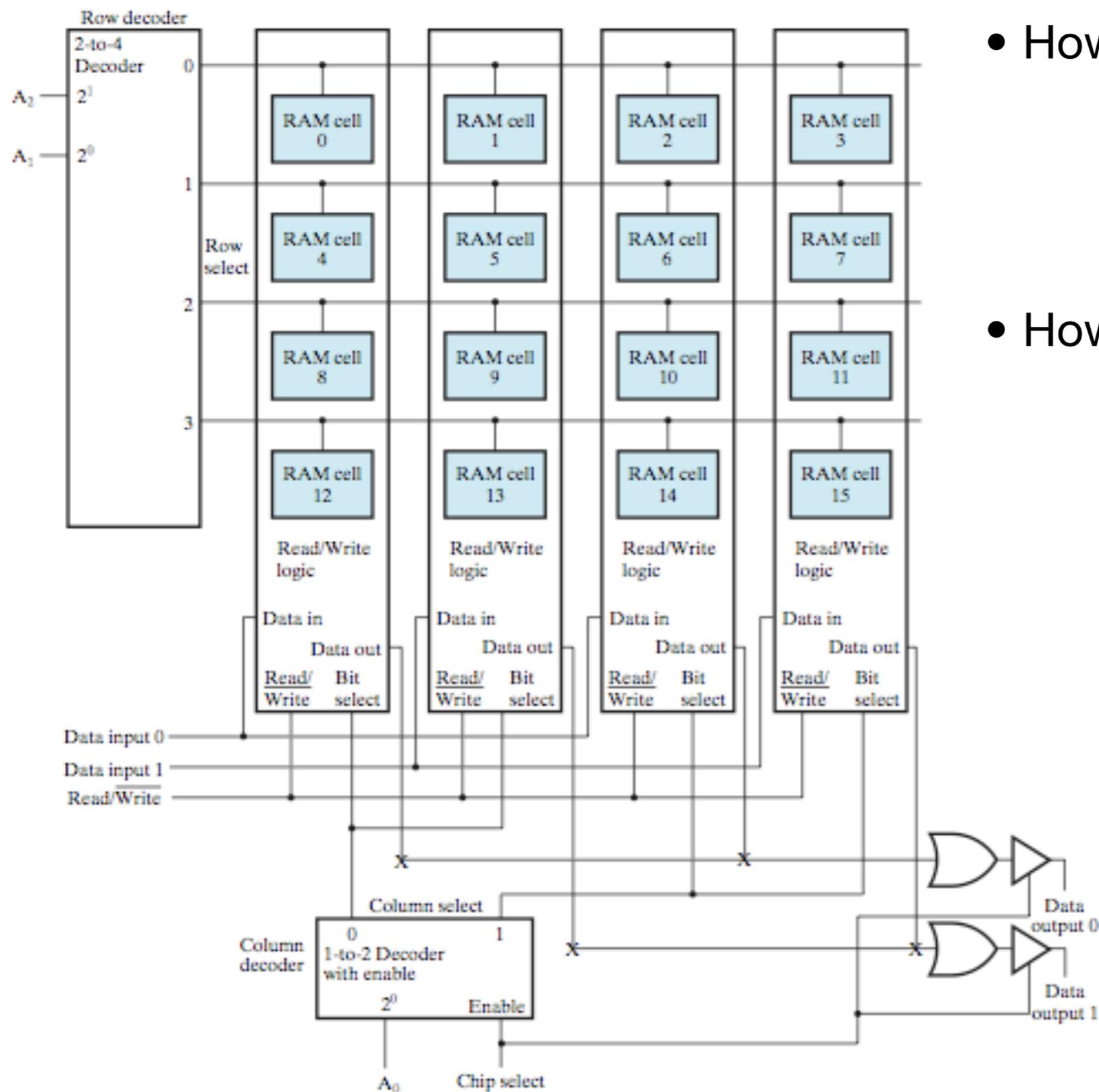


Coincident cell selection



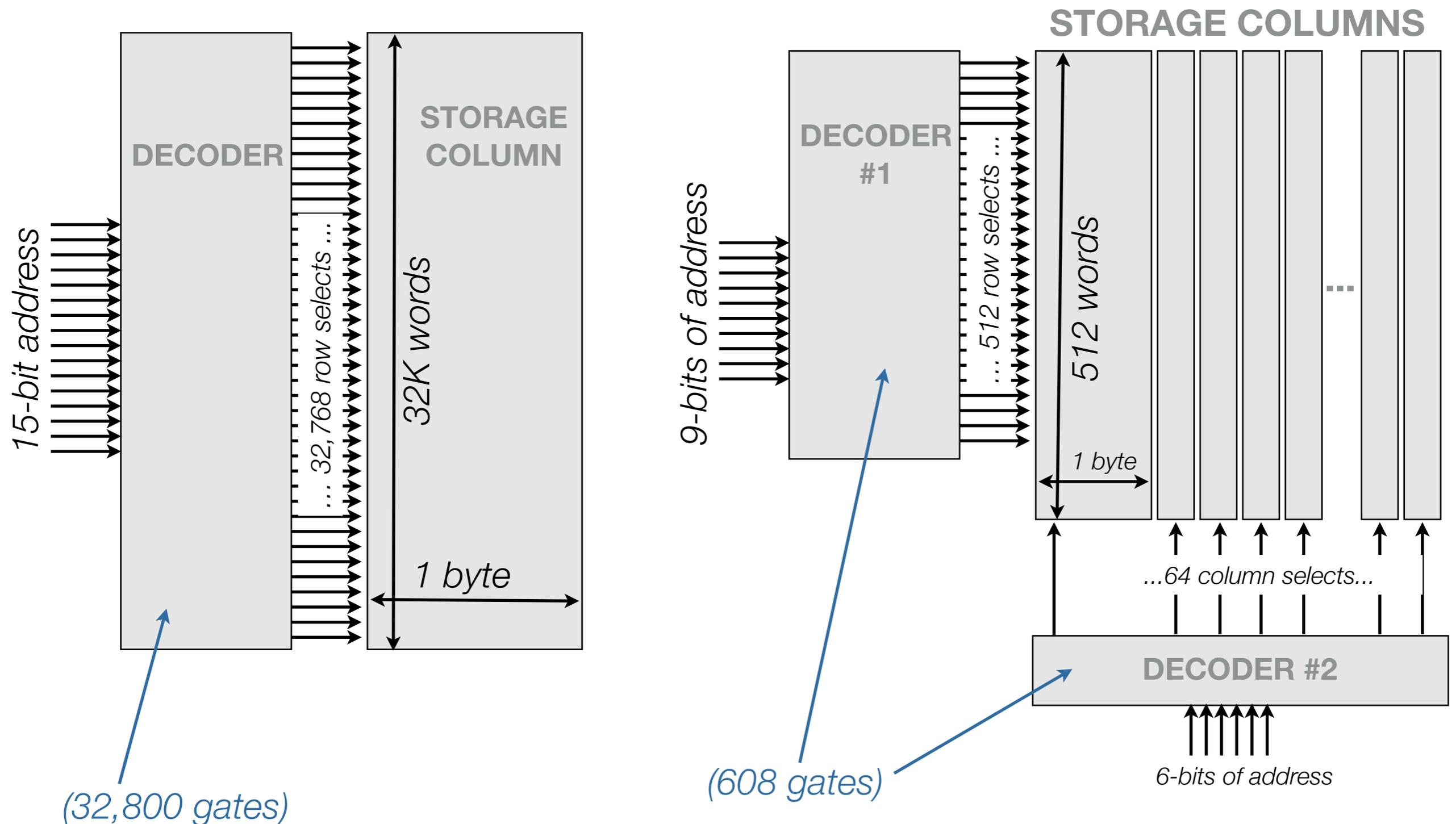
- Decode address into *both* row *and* column select signals
- How many words in this RAM?
- How many bits per word?

Coincident cell selection w. larger words

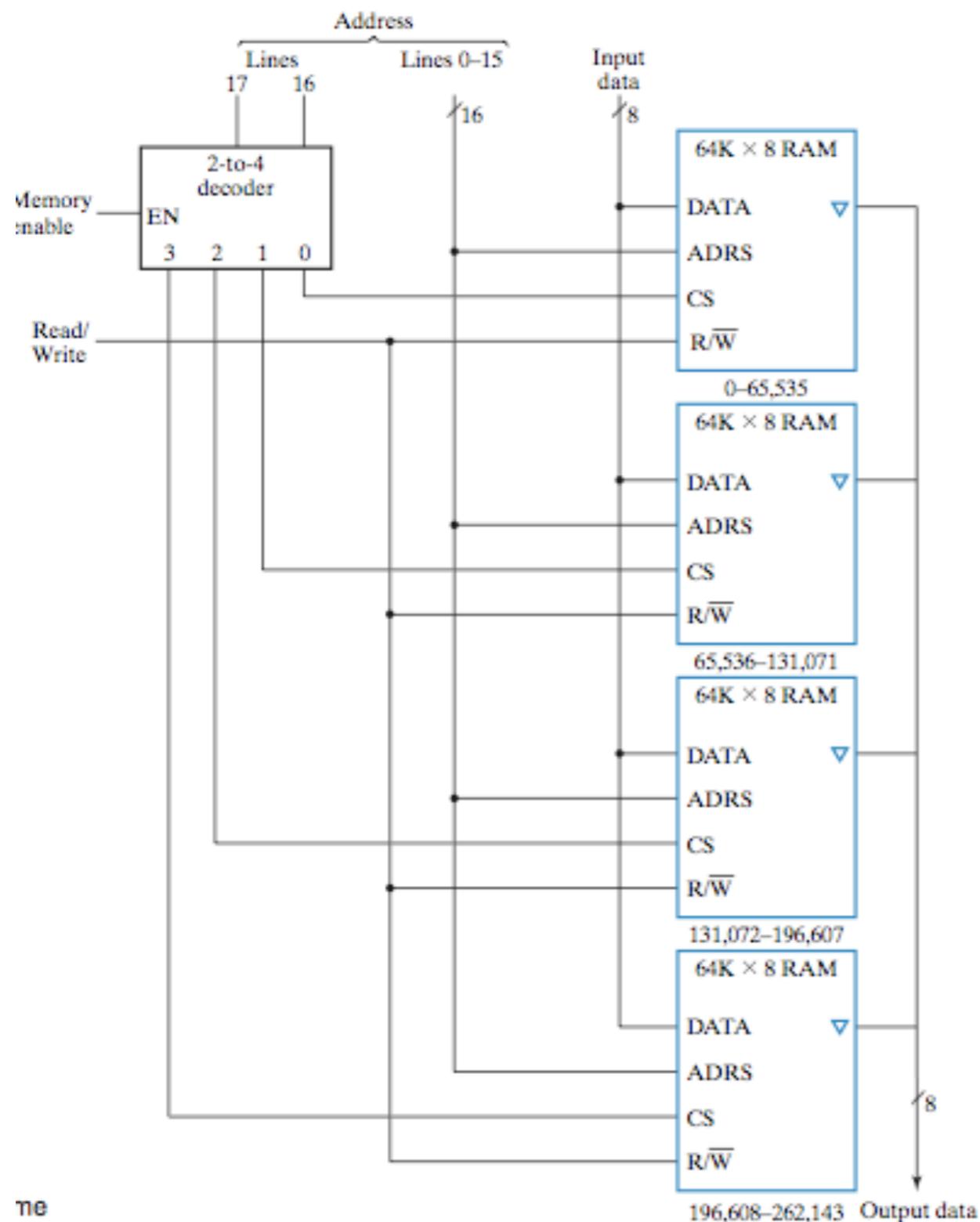


- How many words in this RAM?
- How many bits per word?

Coincident cell selection saves decode logic



Multi-chip memories



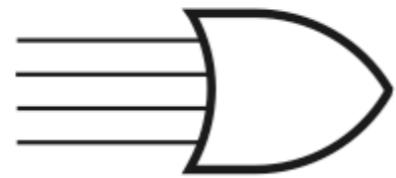
- If you need a larger memory than any available chip
- Wire multiple RAM chips together to work in concert as one large memory

© 2008 Pearson Education, Inc.
M. Morris Mano & Charles R. Kime
LOGIC AND COMPUTER DESIGN FUNDAMENTALS, 4e

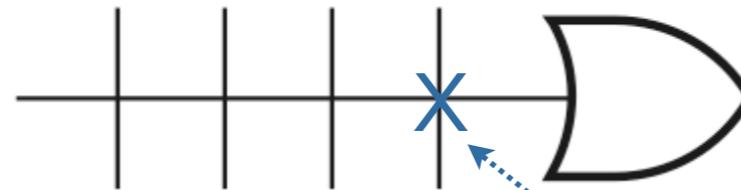
Programmable Logic Devices (M&K 6.8)

Programmable Logic Devices

- Programmable logic devices (PLDs)
 - Structured like memories
 - Used to implement combinational logic



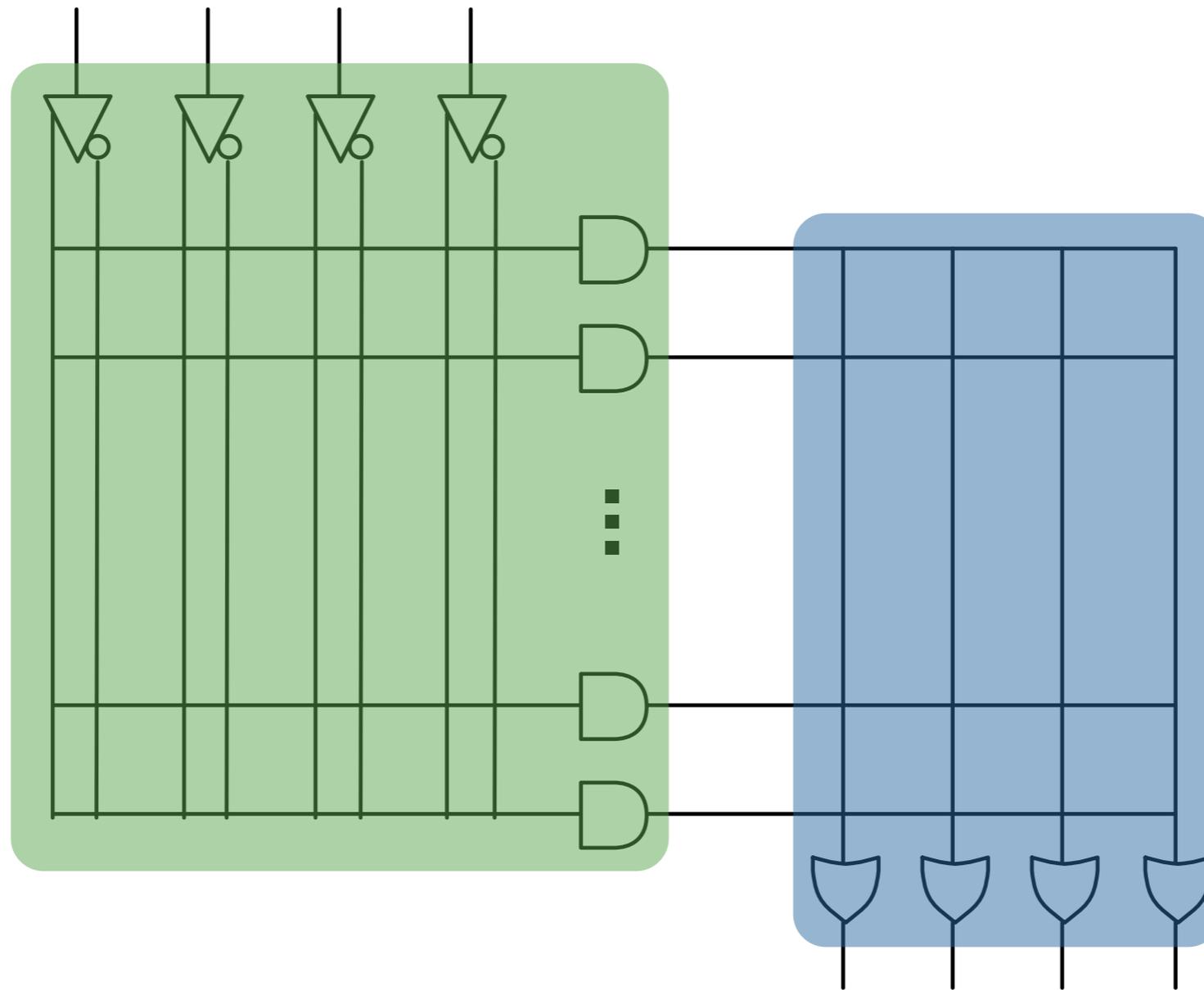
(a) Conventional symbol



(b) Array logic symbol

- “X” on array logic means wire connected to logic gate (e.g. above)
- Connections can be either permanent (e.g., fuse, mask) or not (e.g., Flash)

General PLD architecture

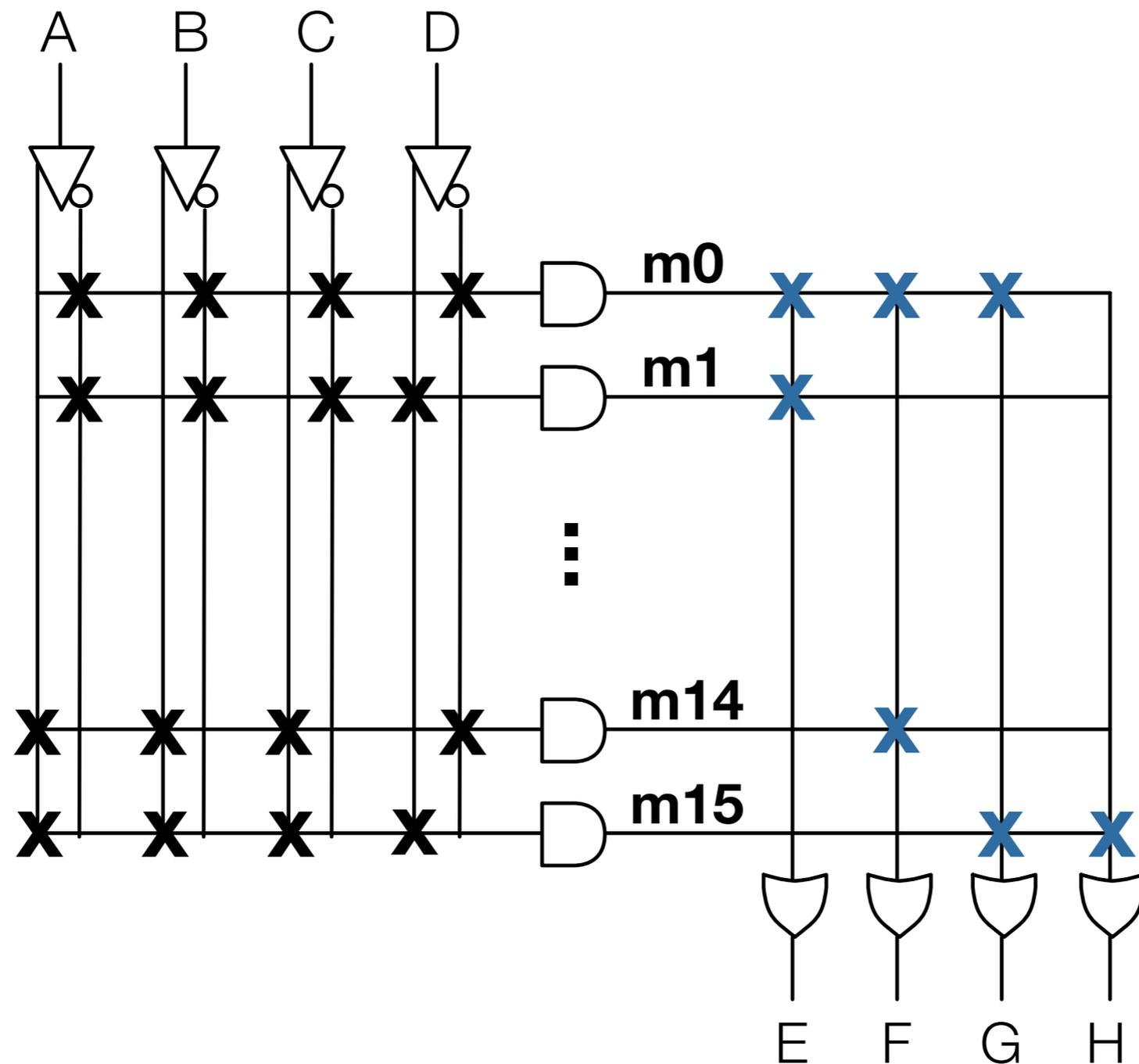


Fixed AND, programmable OR = Programmable ROM (PROM)

Programmable AND, fixed OR = Programmable Array Logic (PAL)

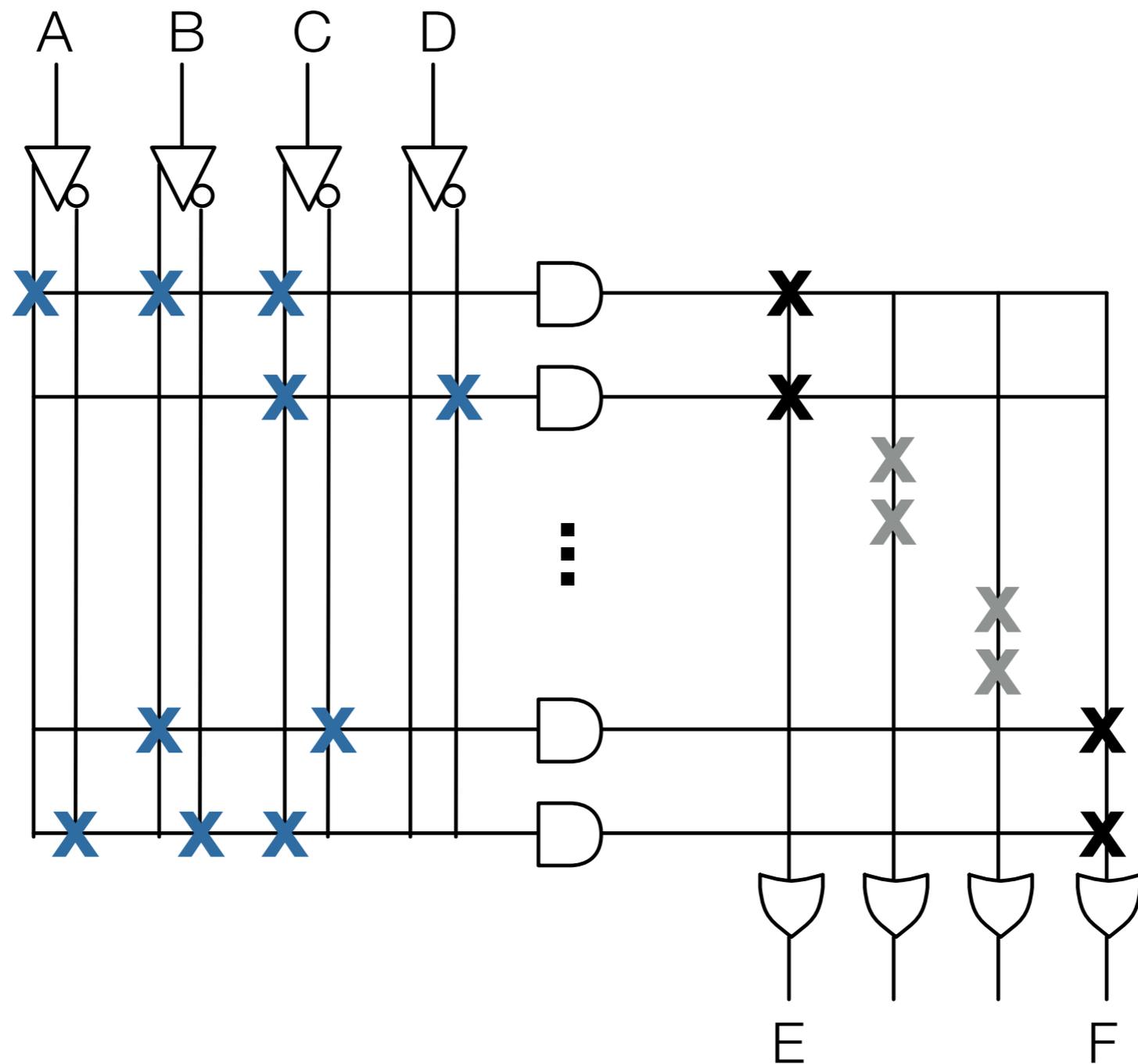
Programmable AND, programmable OR = Programmable Logic Array (PLA)

Programmable ROM (PROM)



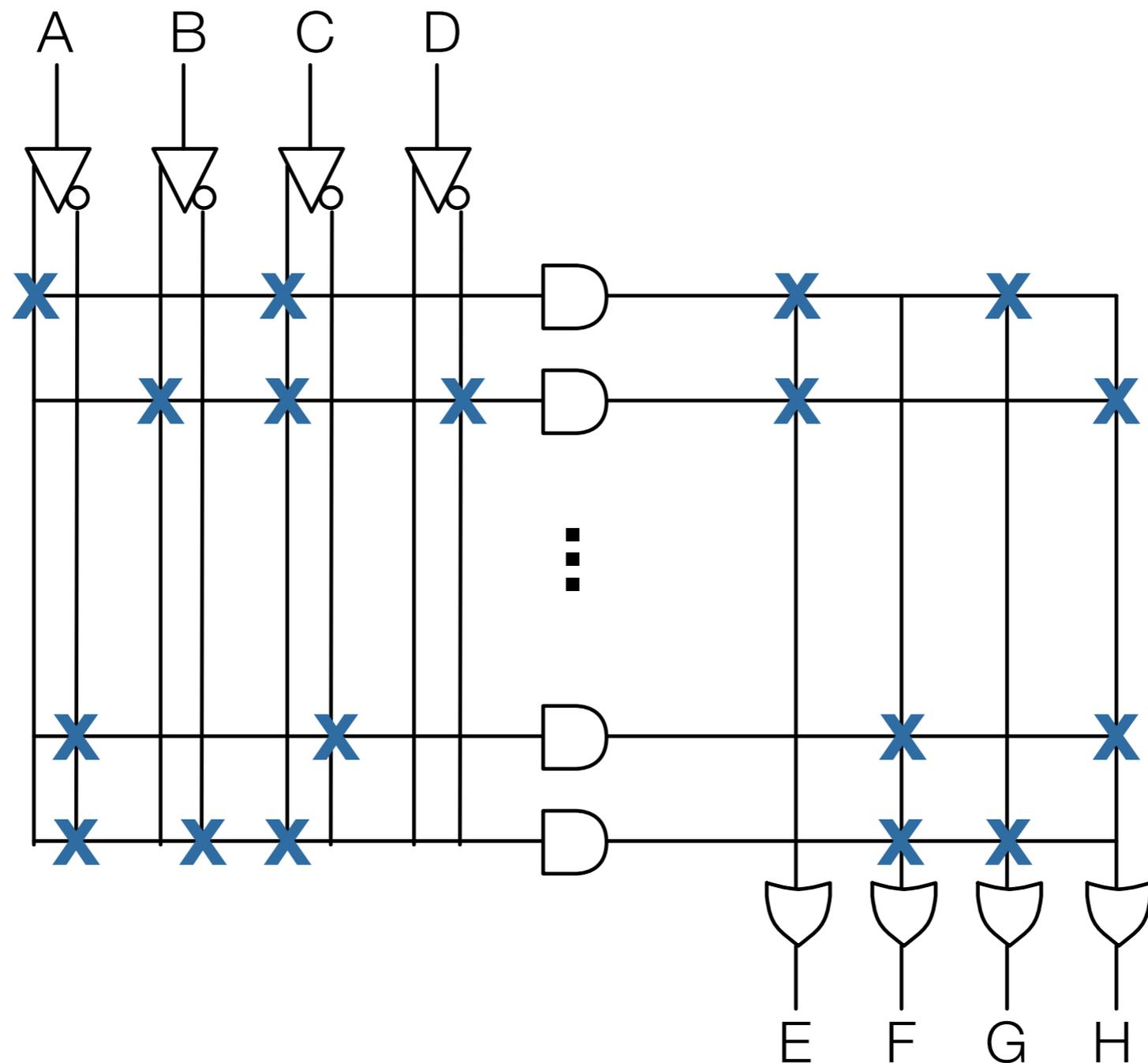
Fixed (X) AND, programmable (X) OR

Programmable Array Logic (PAL)



Programmable (X) AND, fixed (X) OR

Programmable Logic Array (PLA)



Programmable (X) AND, programmable (X) OR