ECE 2300
Digital Logic & Computer Organization

Fall 2016

More ALU
Multiplication
Memories

Cornell University

Arithmetic Logic Unit (ALU)

- Combinational logic circuit that combines a variety of operations into a single unit

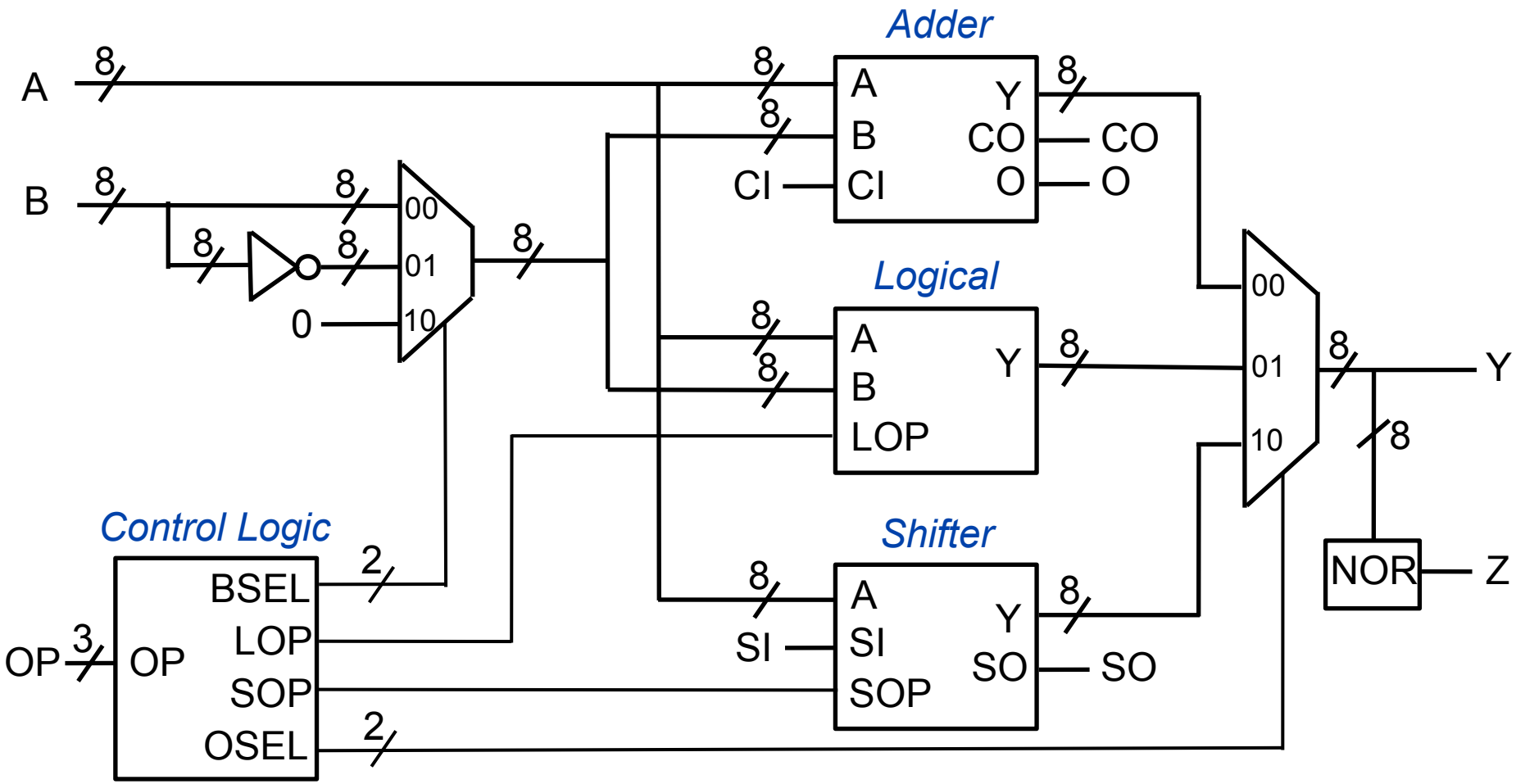
- Typical operations may include
  - Addition and subtraction
  - Logical (OR, AND)
  - Shift and rotate
  - Comparisons

- Computing core of a processor

Example 8-bit Arithmetic Logic Unit

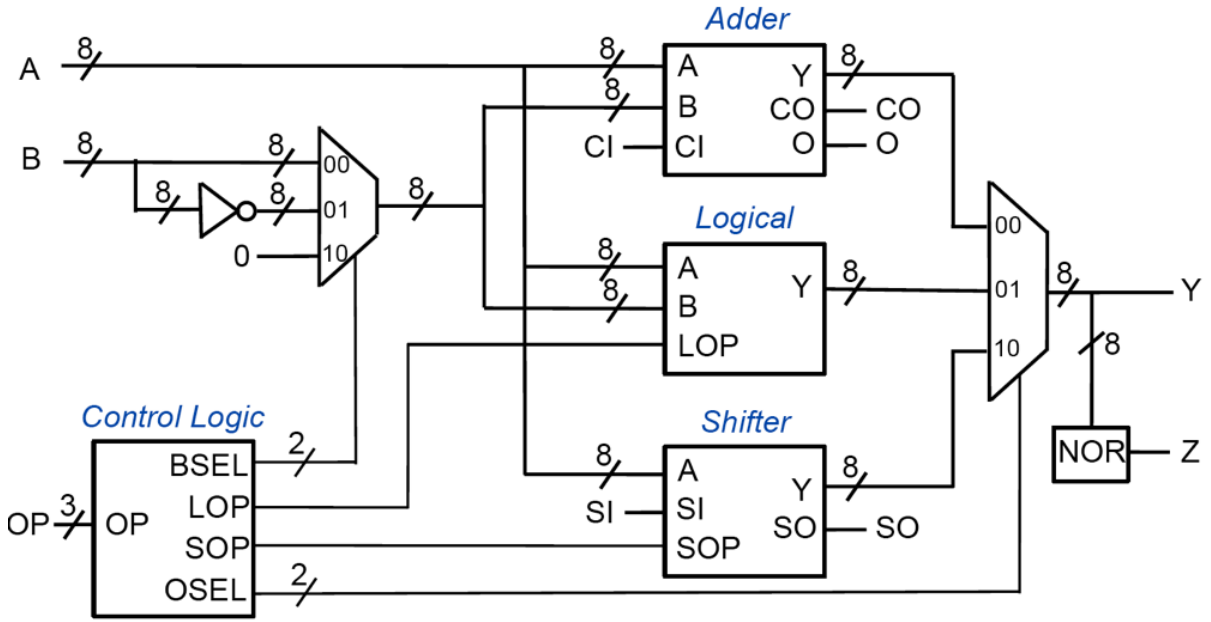
- **Operations**
  - Addition and Subtraction
  - Bitwise AND and OR
  - Shift Left and Shift Right
- **Inputs**
  - A, B, Carry In (CI)
  - Shift In (SI)
  - Code indicating operation to be performed (OP)
- **Outputs**
  - Y, Carry Out (CO)
  - Shift Out (SO)
  - Flags regarding the result of the operation

ALU Block Diagram



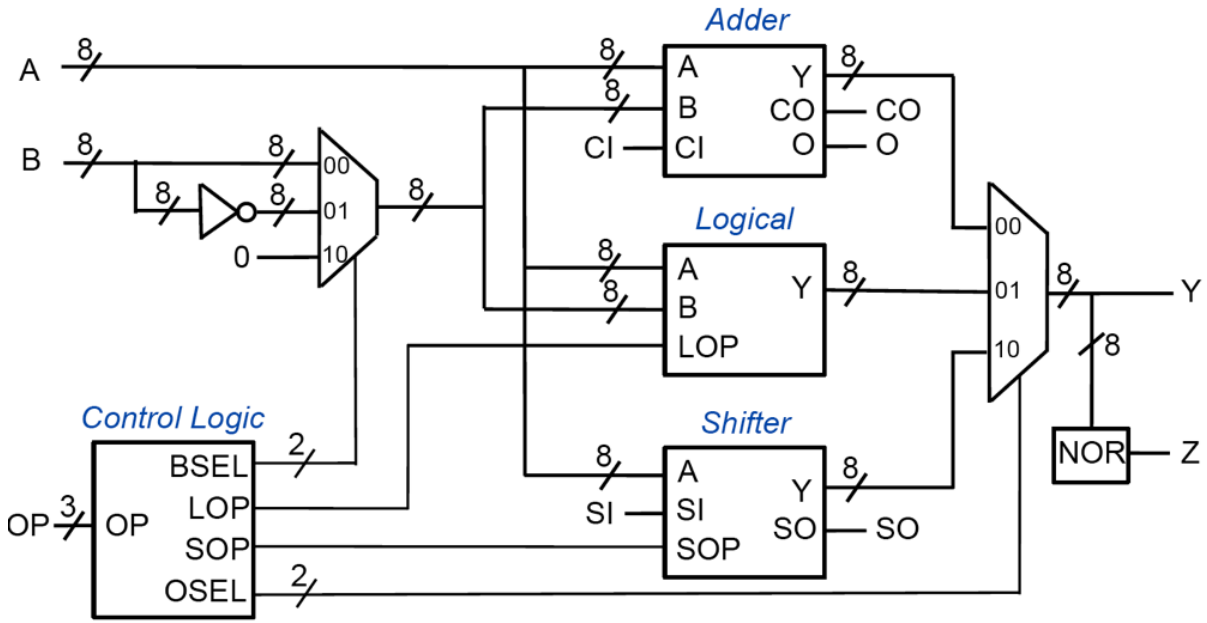
ALU Operations

NAME	OP	BSEL	CI	LOP	SOP	OSEL	Operation
ADD	000	00	0	-	-	00	Y = A + B + CI
SUB	001	01	1	-	-	00	Y = A + B' + CI
AND	011	00	-	0	-	01	Y = A AND B
OR	100	00	-	1	-	01	Y = A OR B
SHL	101	-	-	-	0	10	Y = A[6..0],SI
SHR	110	-	-	-	1	10	Y = SI,A[7..1]
PASS	111	10	0	-	-	00	Y = A + B + CI



Comparison Operations

- To compare A and B, perform A − B
  - If the result is 0, then A = B
  - Z flag set to 1 whenever ALU result is 0
  - Can check for A ≥ B and A < B by observing the MSB of the result (Y) of A − B



Multiplication

- **Form each partial product by multiplying a single digit of the multiplier by the multiplicand**

- **Add shifted partial products to get result**

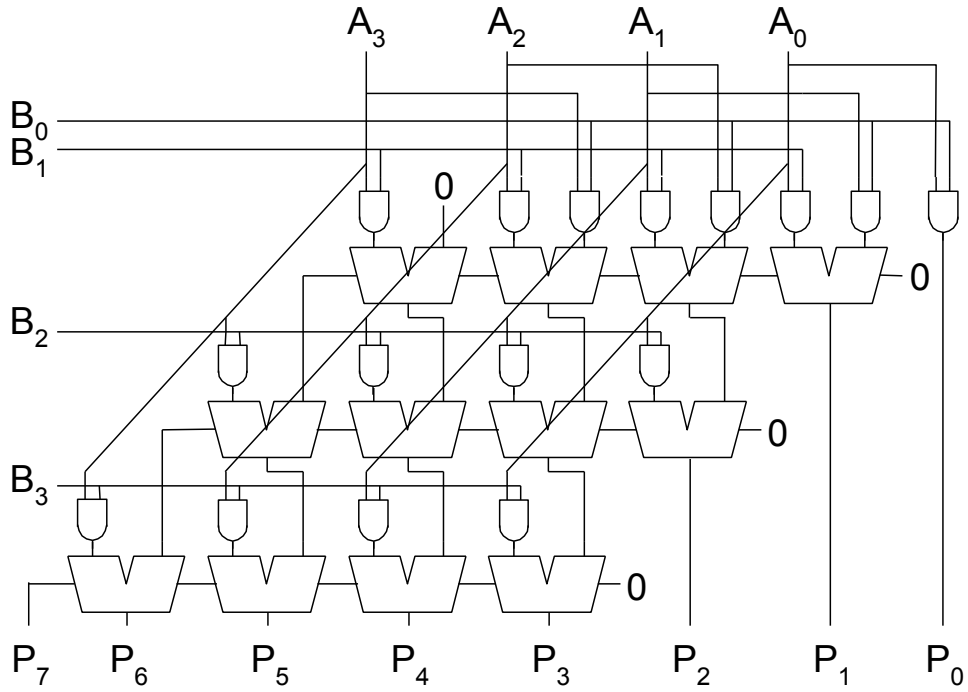
Decimal		Binary
230	multiplicand	0101
x 42	multiplier	x 0111
460	partial products	0101
+ 920		0101
9660		0101
		+ 0000
	result	0100011

230 x 42 = 9660

5 x 7 = 35

4 × 4 Multiplier

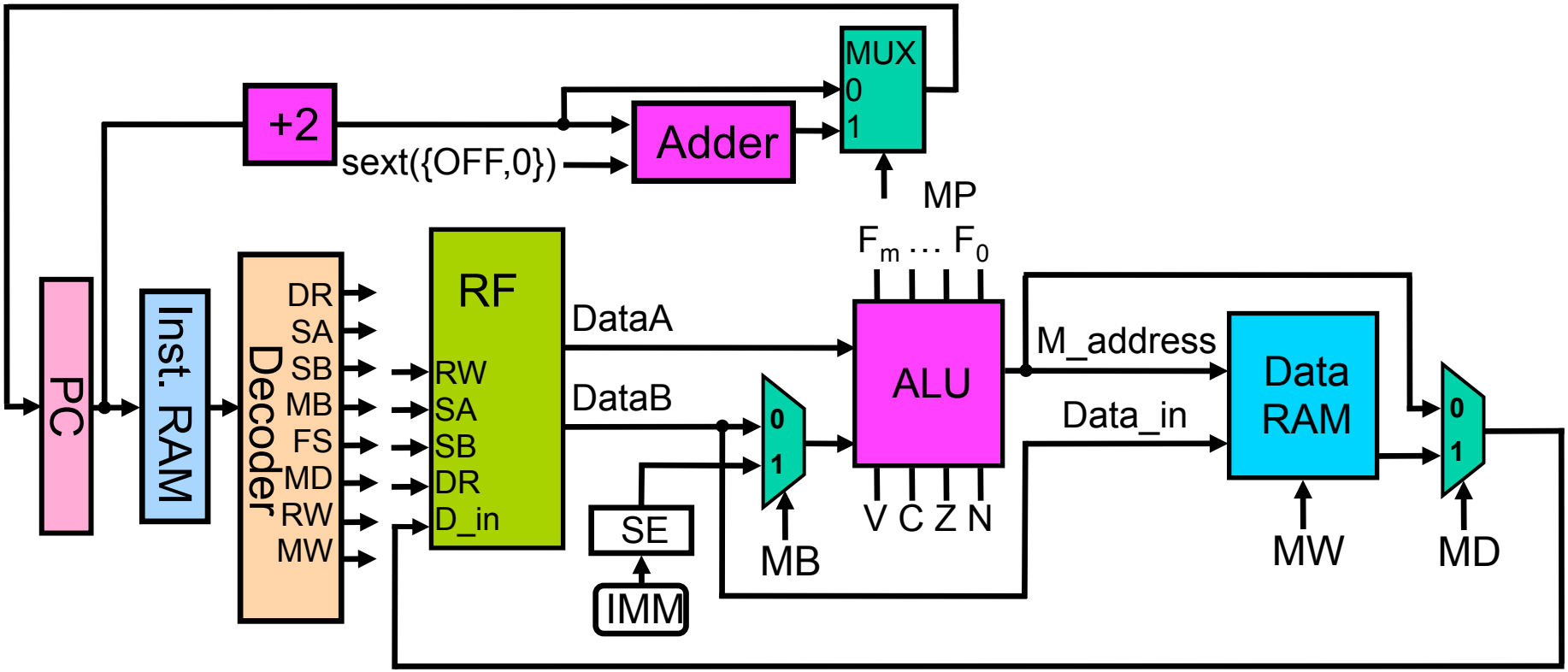
			A_3	A_2	A_1	A_0		
		x	B_3	B_2	B_1	B_0		
		A_3B_0	A_2B_0	A_1B_0	A_0B_0			
	A_3B_1	A_2B_1	A_1B_1	A_0B_1				
A_3B_2	A_2B_2	A_1B_2	A_0B_2					
+ A_3B_3	A_2B_3	A_1B_3	A_0B_3					
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0	



Memories in Digital Systems

- Memory is used in virtually all digital designs
- Microprocessors have lots of memory
  - Caches
  - Register files
  - Buffers, queues, etc
  - Arrays for logic functions
- Most microprocessor memory is SRAM
  - Very fast
- Main memory uses DRAM
  - Denser, lower cost/bit, less power

Our Microprocessor Needs Memory



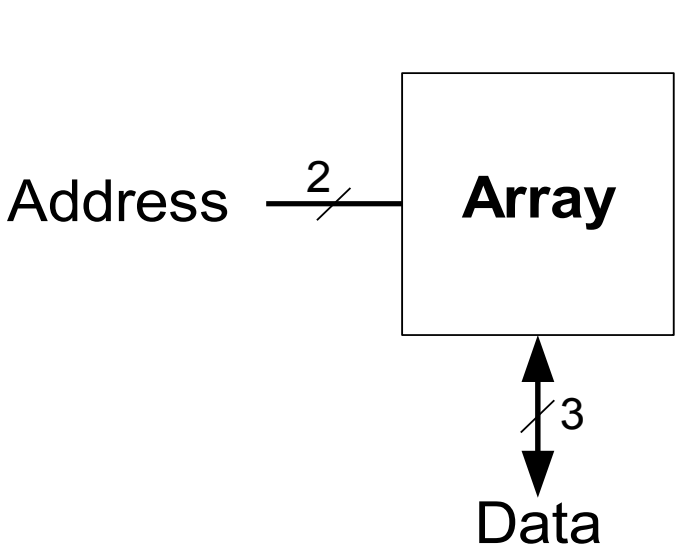
Types of Memories

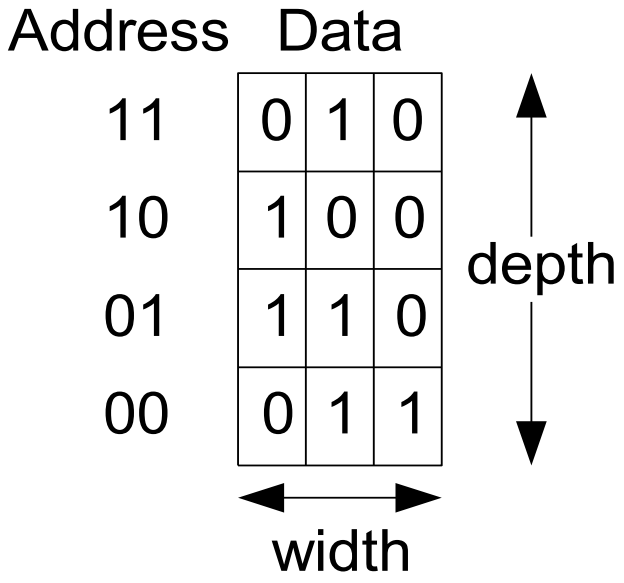
- **Random Access Memory (RAM)**
  - Read and write any location at similar speeds
  - Volatile: loses contents when powered off
  - SRAM, DRAM

- **Read-Only Memory (ROM)**
  - Truly read-only
    - Written in the factory, and never written after installation
  - Mostly read and rarely written
    - Much faster to read than write
  - Non-volatile
  - ROM, PROM, EPROM, EEPROM, Flash memory

General Memory Organization

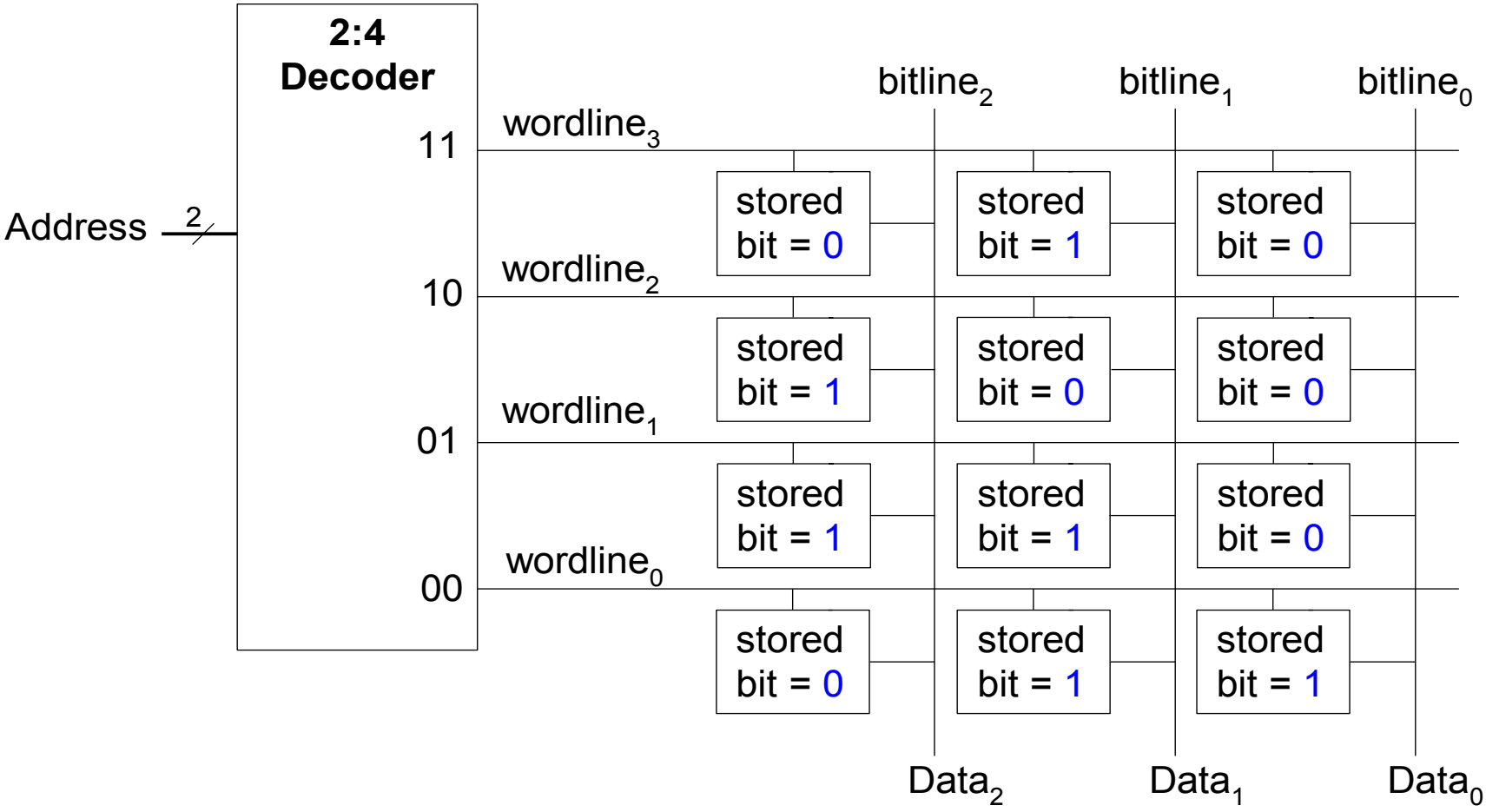
- Two dimensional array of *bit cells*
- Each bit cell stores 1 bit
- *Address* selects a row of *data* (multiple bit cells)





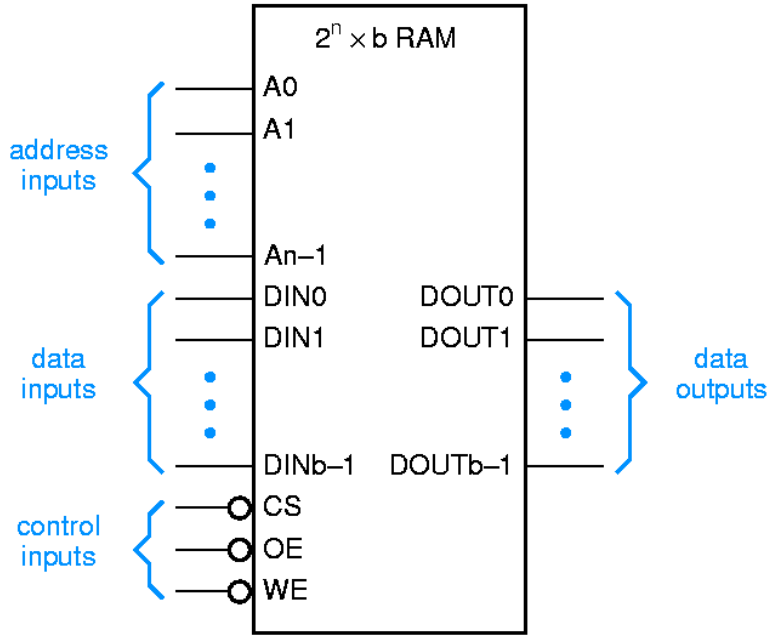
General Memory Organization

- **More detail**

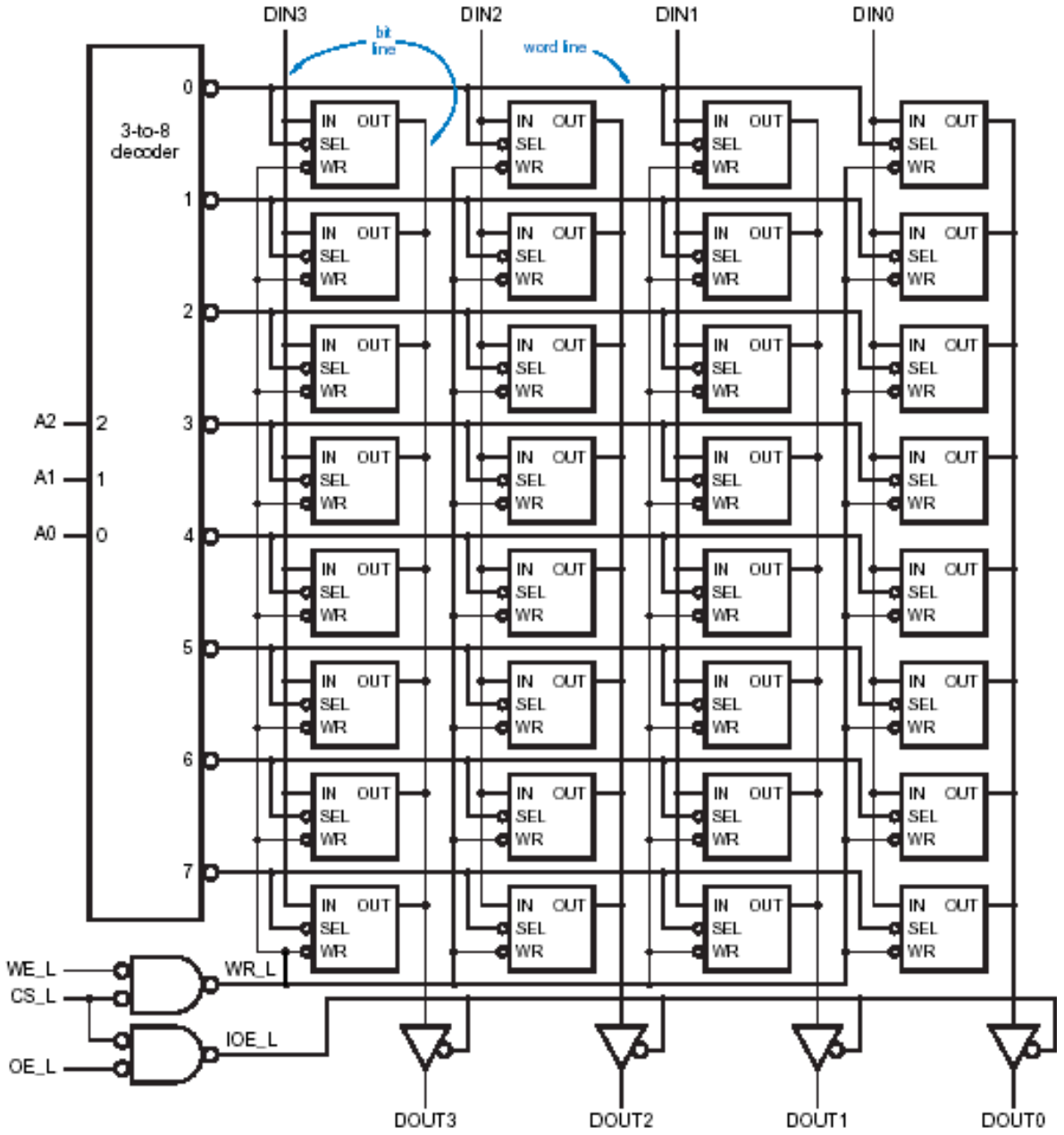


Static RAM (SRAM)

- Address: points to the location to read or write
- Data inputs: data to be written into memory
- Data outputs: data read from memory
- Write Enable (WE): tells whether to read or write
- Chip Select (CS): selects the chip for reading/writing
- Output Enable (OE): controls the data output tristate drivers

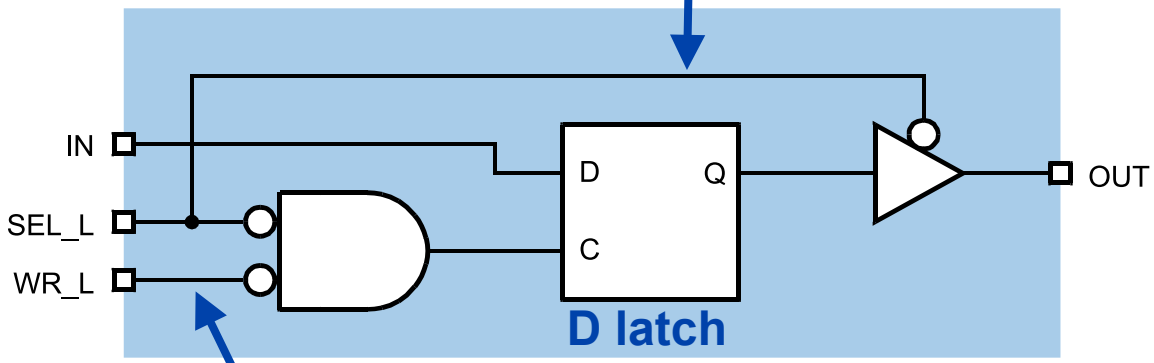


SRAM Internal Structure



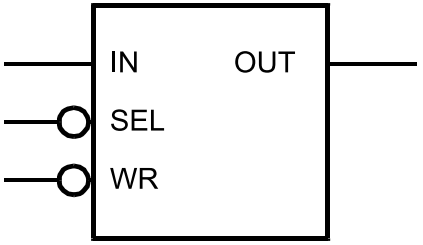
SRAM Cell

SEL_L asserted to access output (read)

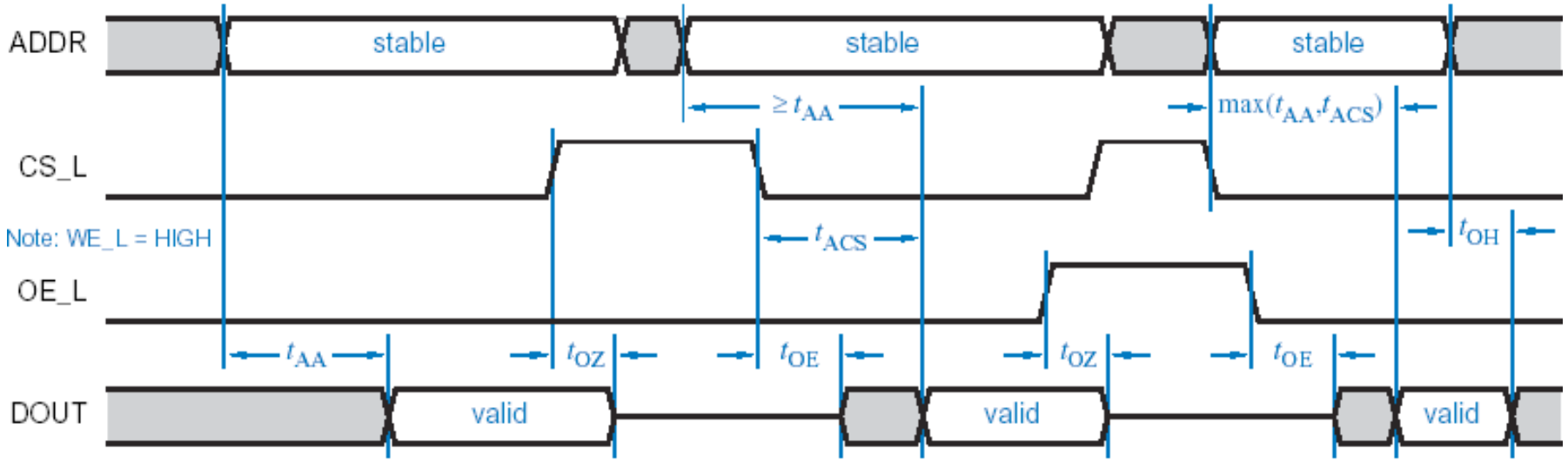


SEL_L and WR_L asserted to enable latch (write)

≡



SRAM Read Timing



t_{AA} Access time from address

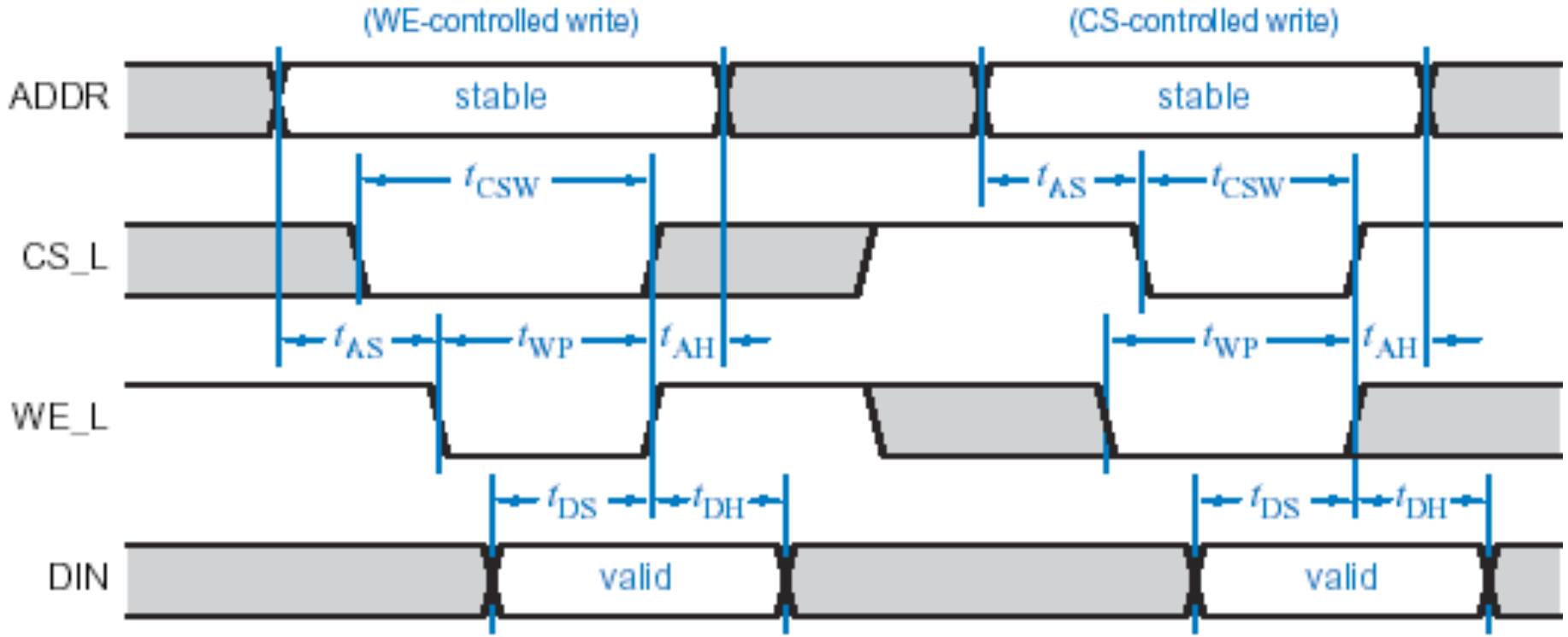
t_{ACS} Access time from chip select

t_{OE} Output enable time

t_{OZ} Output disable time

t_{OH} Output hold time

SRAM Write Timing



t_{AS} Address setup time before latest of CS_L & WE_L

t_{AH} Address hold time after CS_L & WE_L

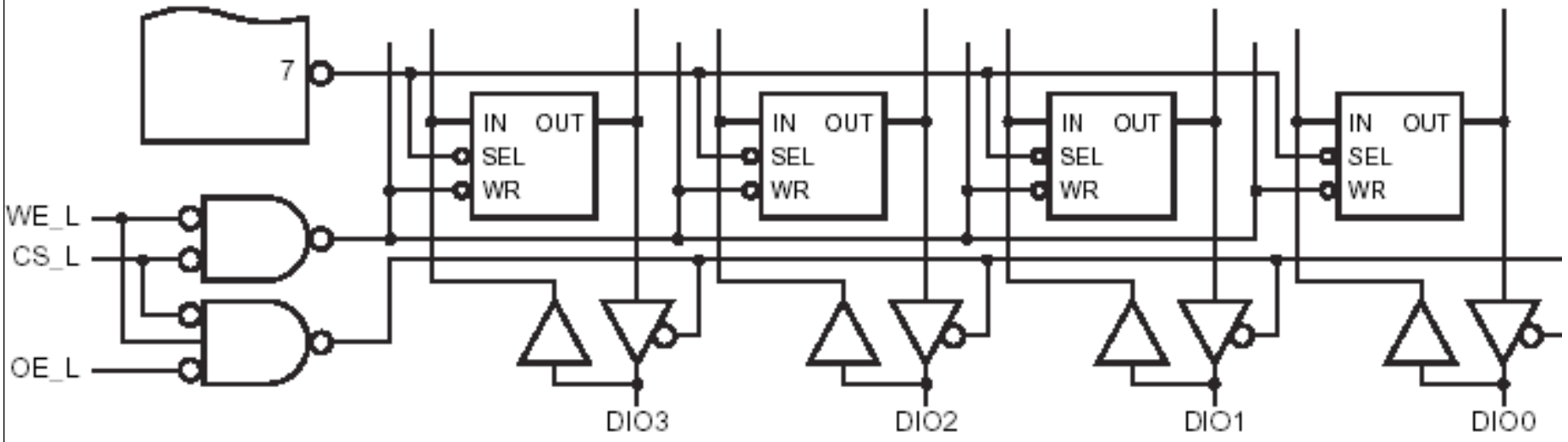
t_{CSW} Chip select setup before WE_L

t_{wp} Write pulse width

t_{DS} Data setup time before WE_L

t_{DH} Data hold time after WE_L deasserted

Bi-directional Data Pins



- Reduces the number of data pins by half
- Disable output buffer whenever WE_L asserted

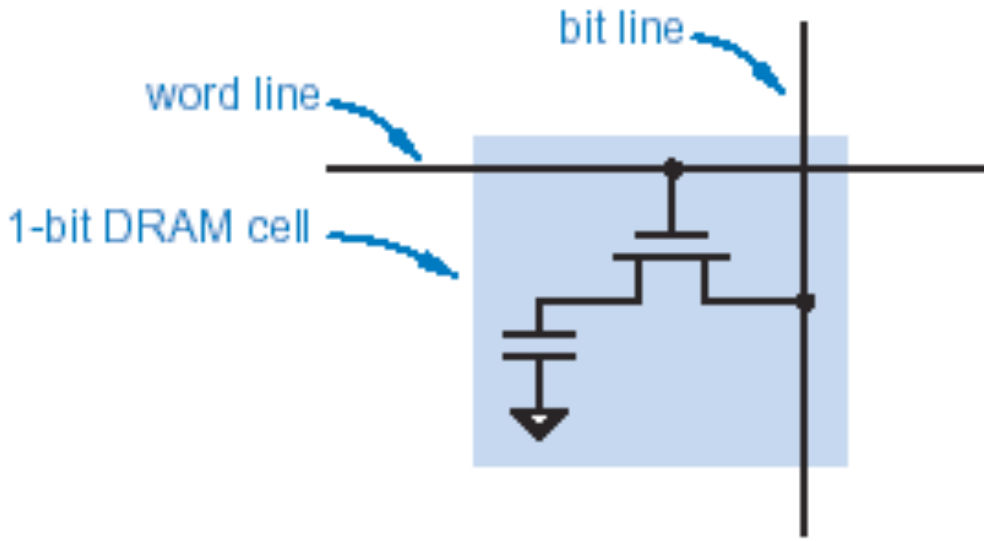
Dynamic RAM (DRAM)

- SRAM advantages and disadvantages
  - Very fast (+)
  - High power (−)
  - Relatively high area/bit (−)

- DRAM
  - Single transistor storage cell
  - Higher density ➔ lower cost/bit
  - Lower power/bit

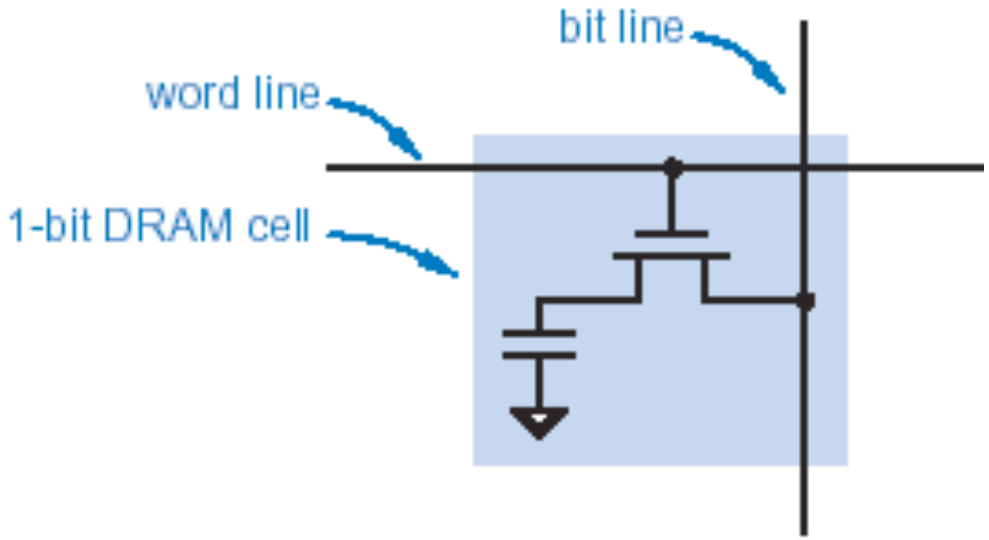
DRAM Bit Structure

- Capacitor accessed through a transistor
- Capacitor is charged through the bit line to store a 1
- Capacitor is discharged through the bit line to store a 0



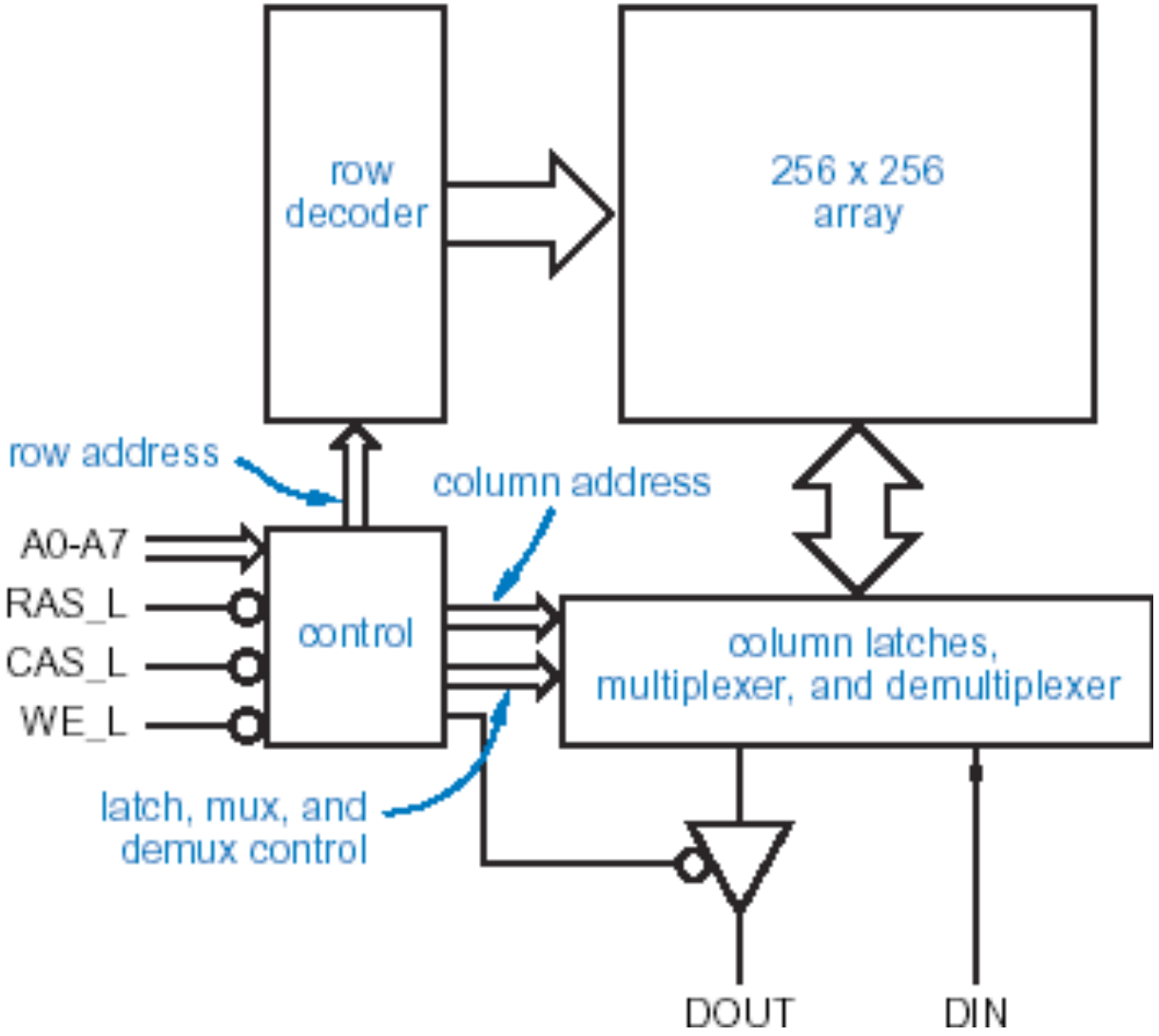
DRAM Read

- Bit line *precharged* halfway between 0 and 1
- Word line is asserted
- Capacitor voltage pulls the bit line slightly higher or lower
- *Sense amplifier* detects this small change (1 or 0)

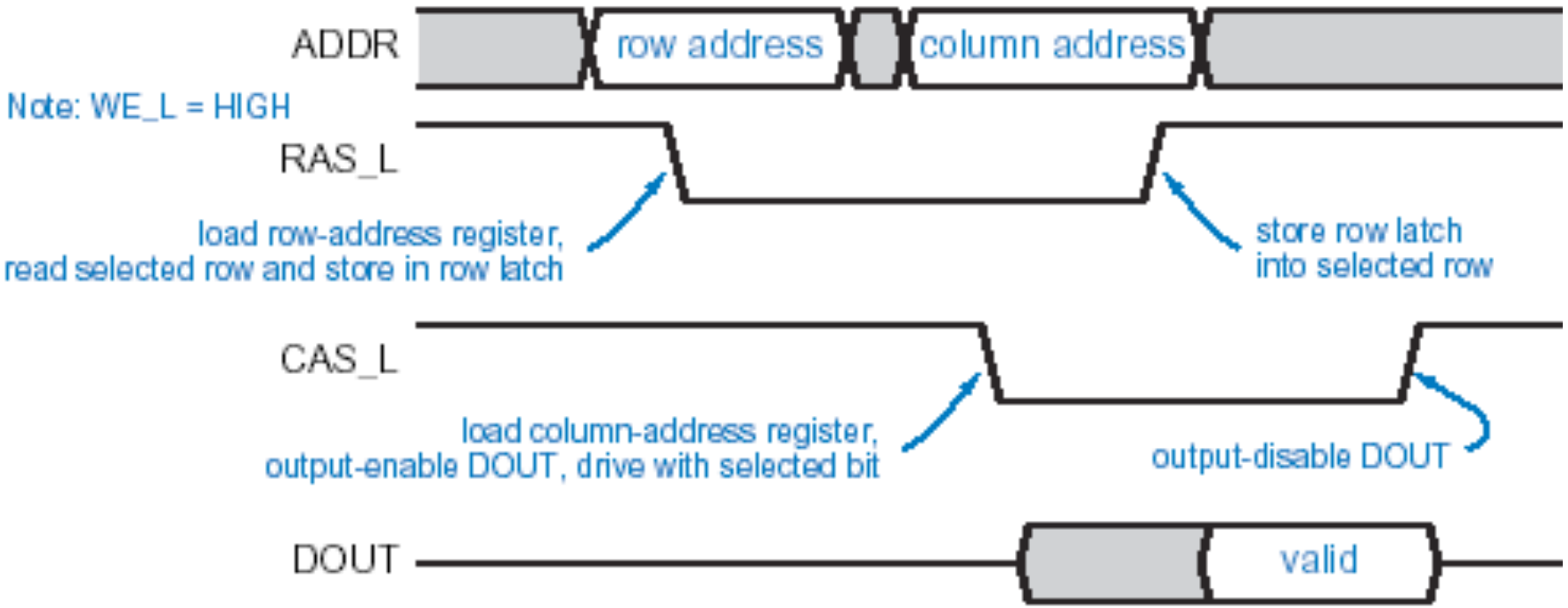


DRAM Organization (64K x 1)

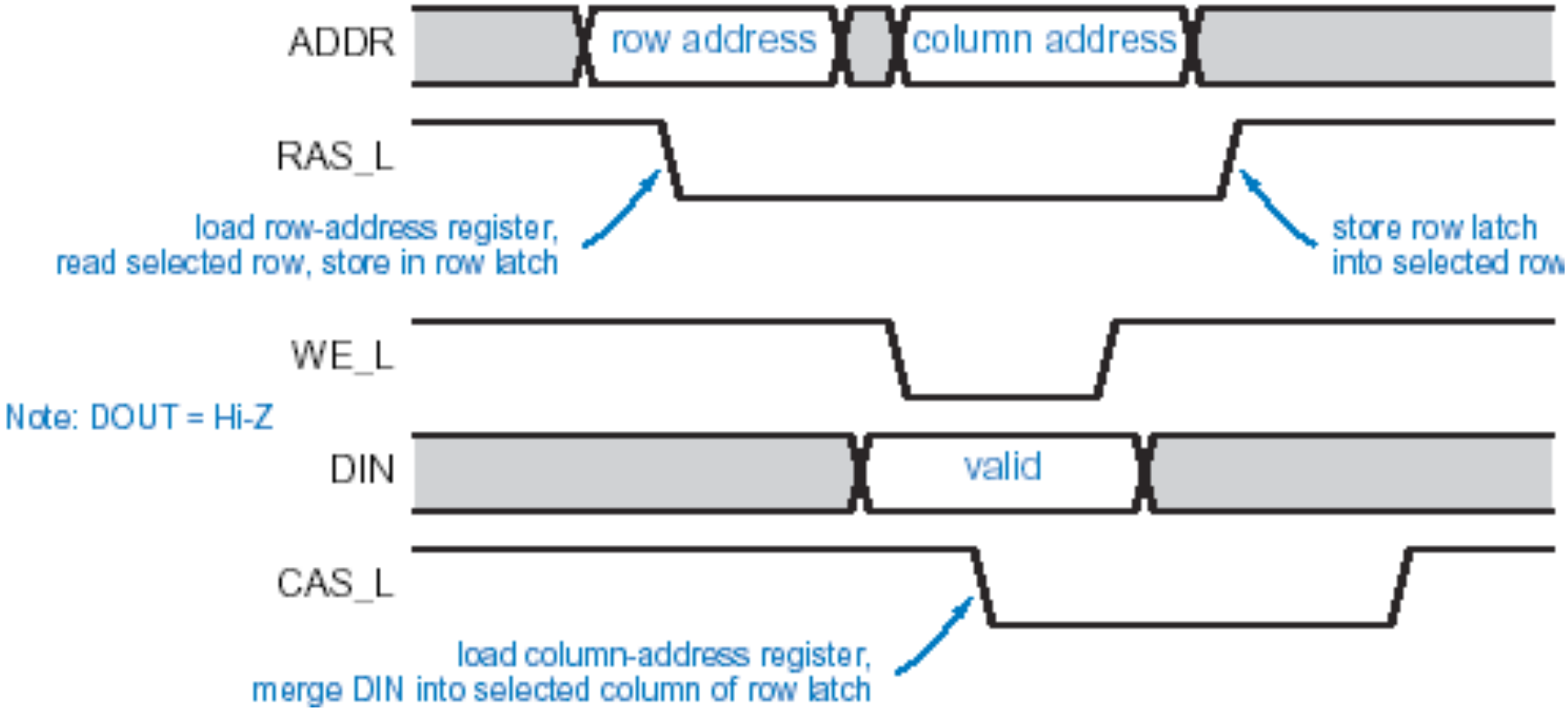
- **Multiplexed address inputs**
- **Row-address strobe (RAS)**
  - **Selects row of array**
- **Column-address strobe (CAS)**
  - **Selects subset of the row**



DRAM Read Cycle

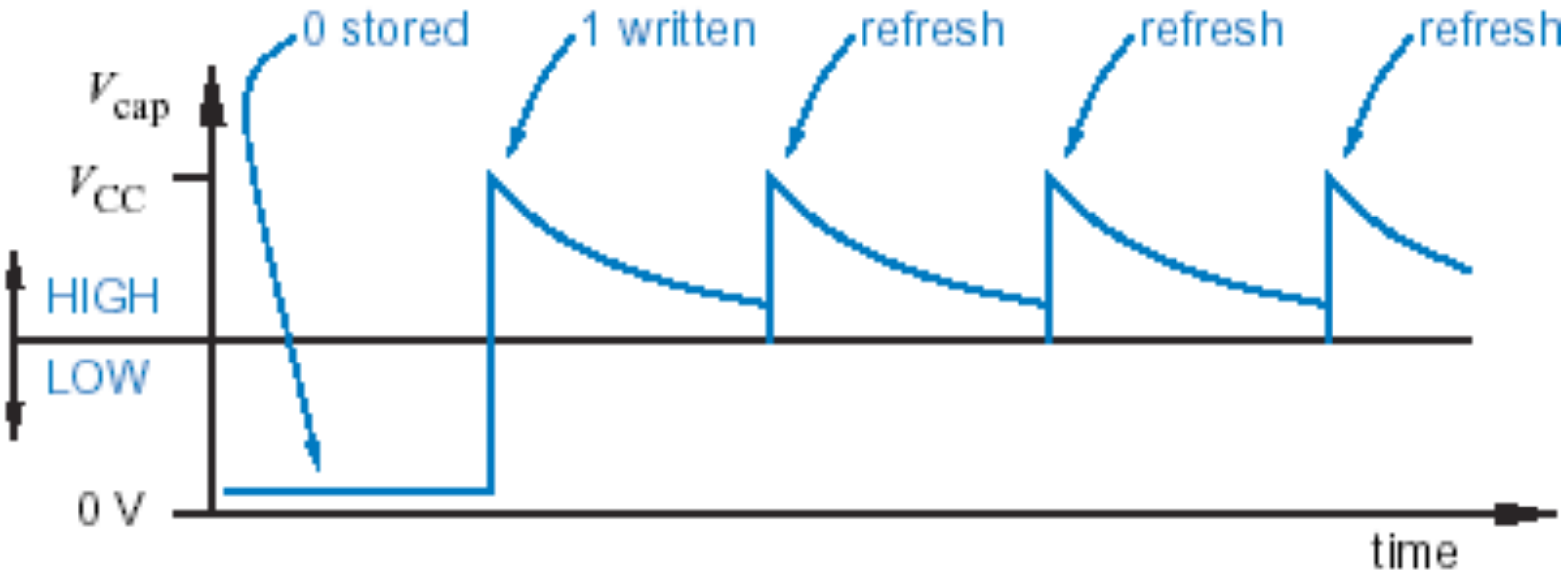


DRAM Write Cycle



DRAM Refresh

- Capacitors discharge over time
- *Refresh* cycles recharge each memory bit
- Each row periodically accessed using RAS, which restores the charge



DRAM Refresh

- Refresh an entire row at a time
- Old DRAMs → Each bit must be refreshed every 4 ms
  - 256 refresh operations for entire 64Kx1 DRAM
  - Initiate a refresh cycle every 15.6 microseconds
- New DRAMS → Each bit must be refreshed every 64 ms, but have 4096 rows
  - Still must initiate a refresh every 15.6 microseconds

Synchronous DRAM (SDRAM)

- D FFs on input and output signals
- Can “pipeline” multiple read and write operations
- Multiple banks that can be accessed concurrently
- DDR: data transferred on both rising and falling clock edges

Before Next Class

- H&H 7.1-7.3.1

Next Time

More Memories
Single Cycle Microprocessor