ECE 2300
Digital Logic & Computer Organization

Spring 2025

Memories

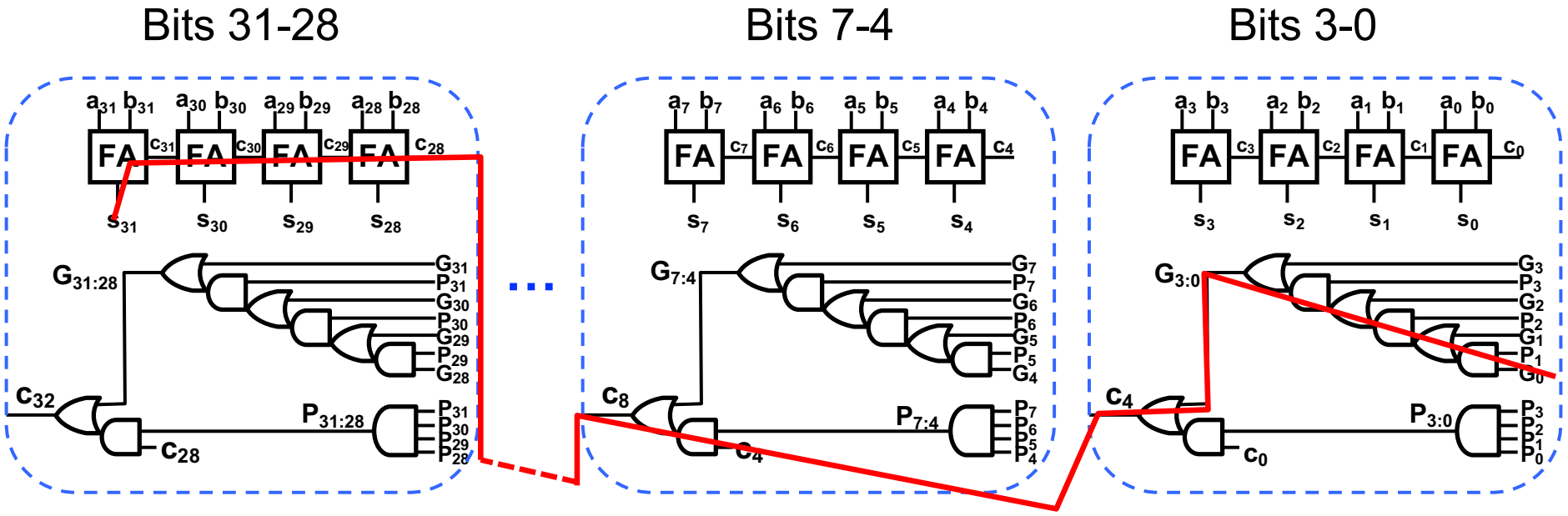
Cornell University

Announcements

- Final exam is scheduled on Saturday May 10 @ 9am

- Lab 3a due tomorrow

Review: Critical Path of 32-bit CLA

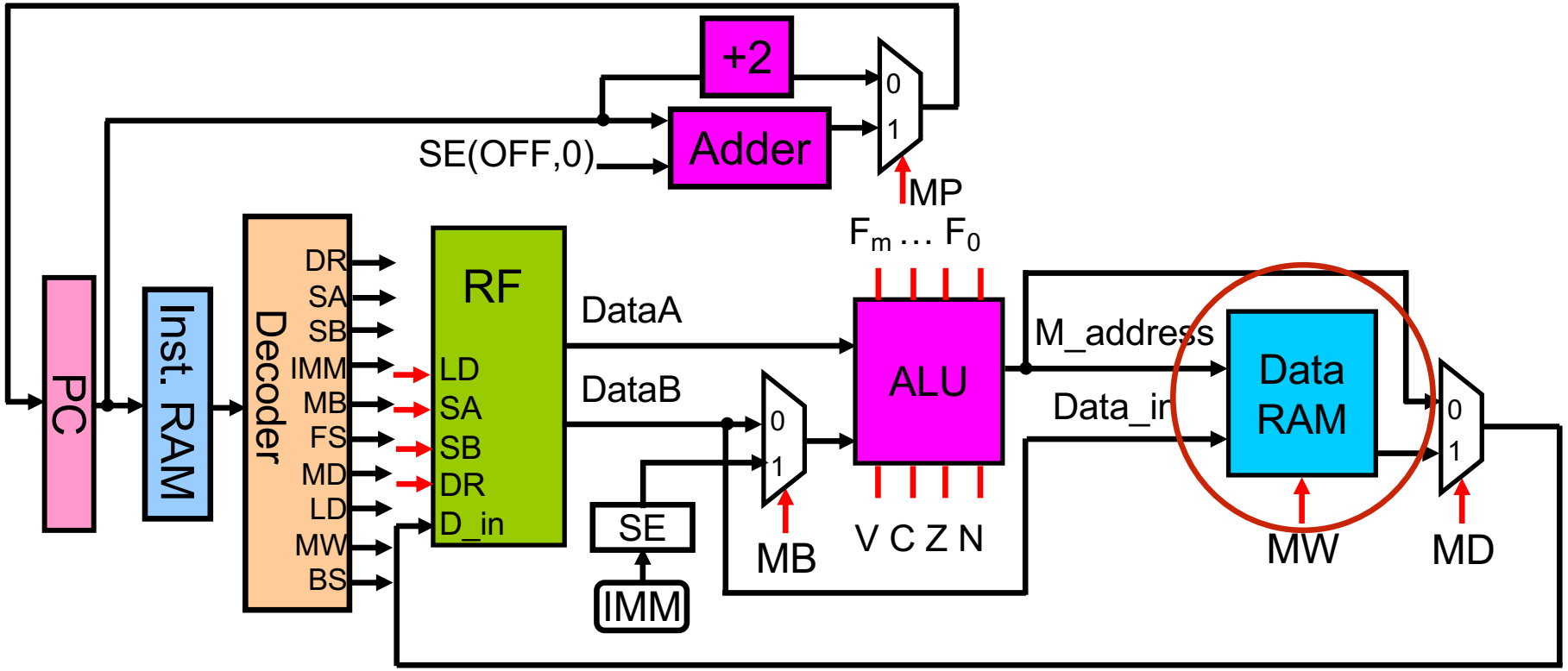


Carry generation and propagation functions (e.g., $G_{7:4}$, $P_{7:4}$) in a 4-bit CLA block do not depend on results from other CLA blocks

⇩

P's and G's for all CLA blocks are generated *in parallel*

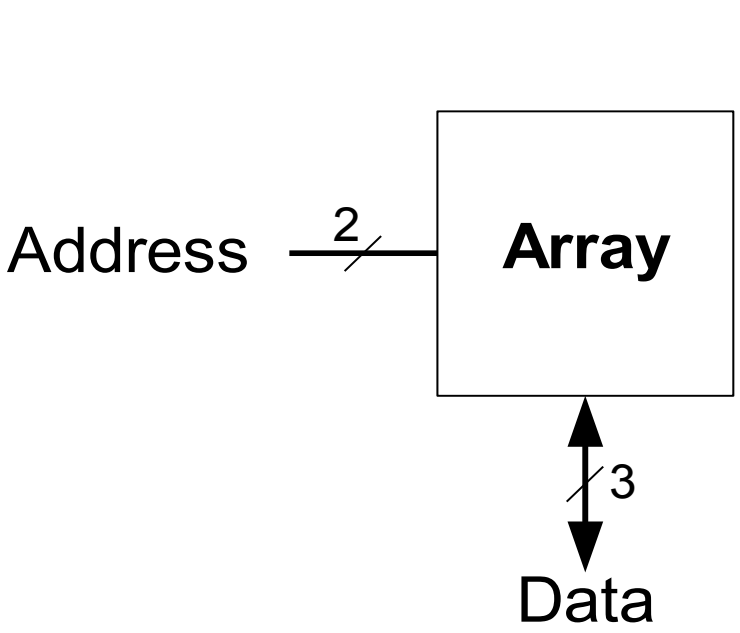
Our Microprocessor Needs (Large) Memory

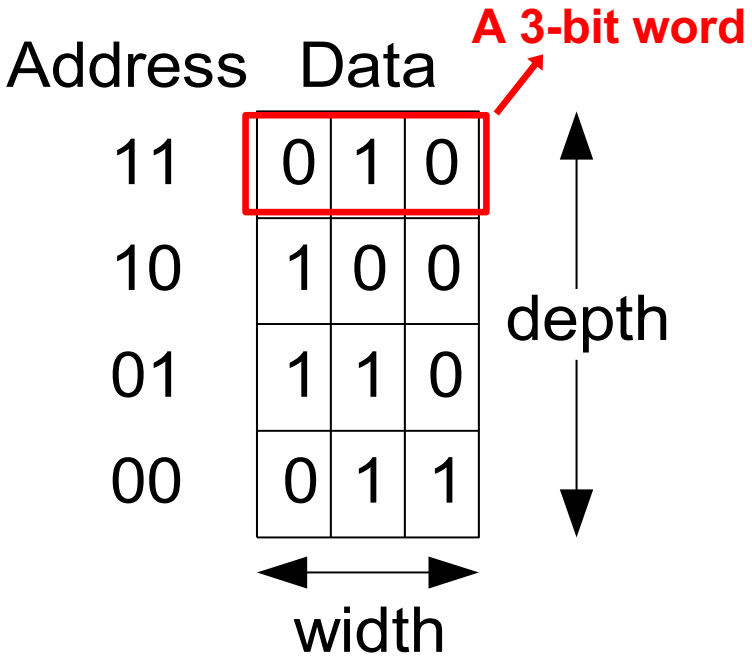


Memory Organization: An Abstract View

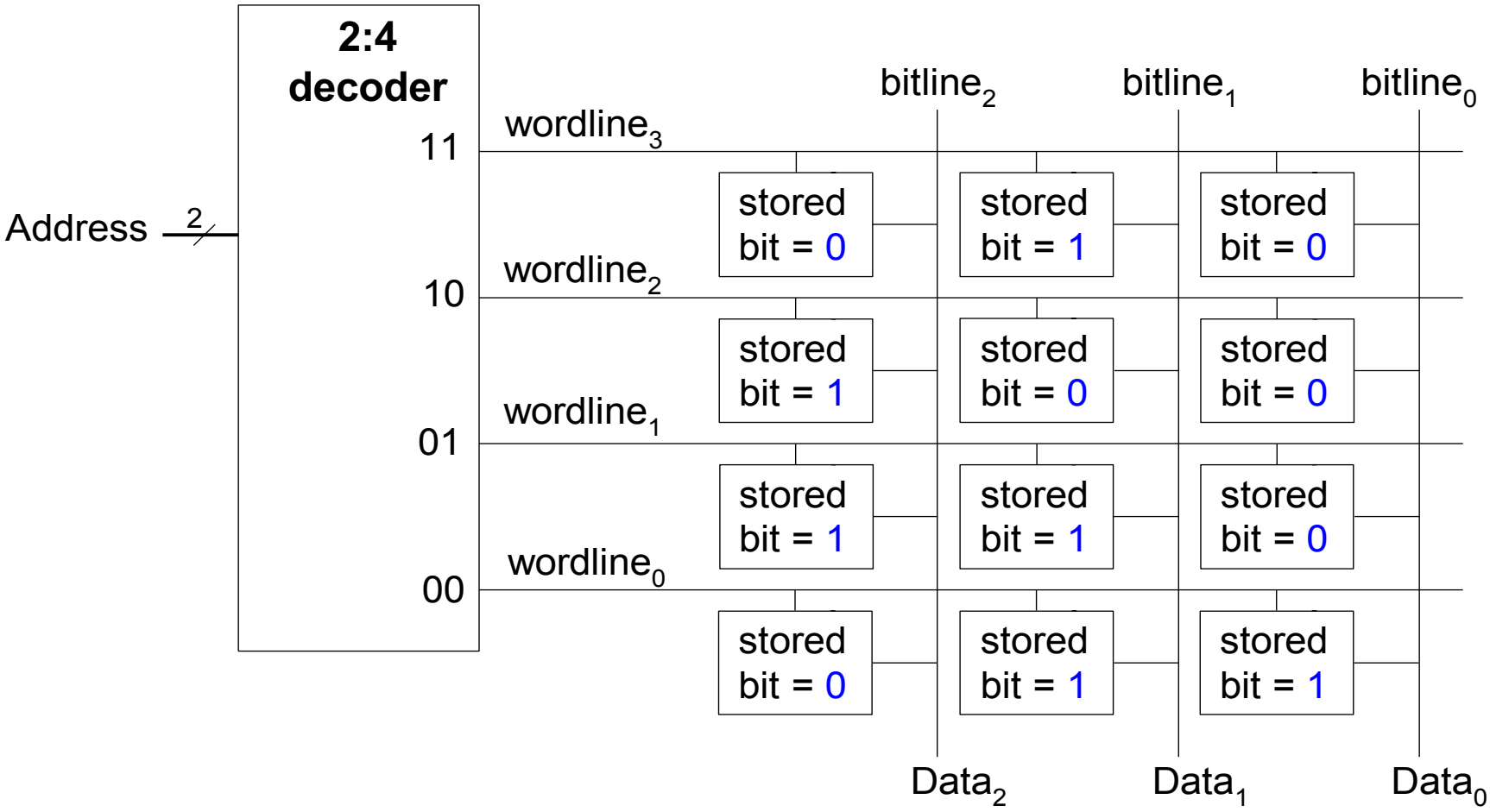
- Two dimensional array of bit cells
  - Each bit cell stores 1 bit

- Address selects a word with multiple bits





General Memory Structure



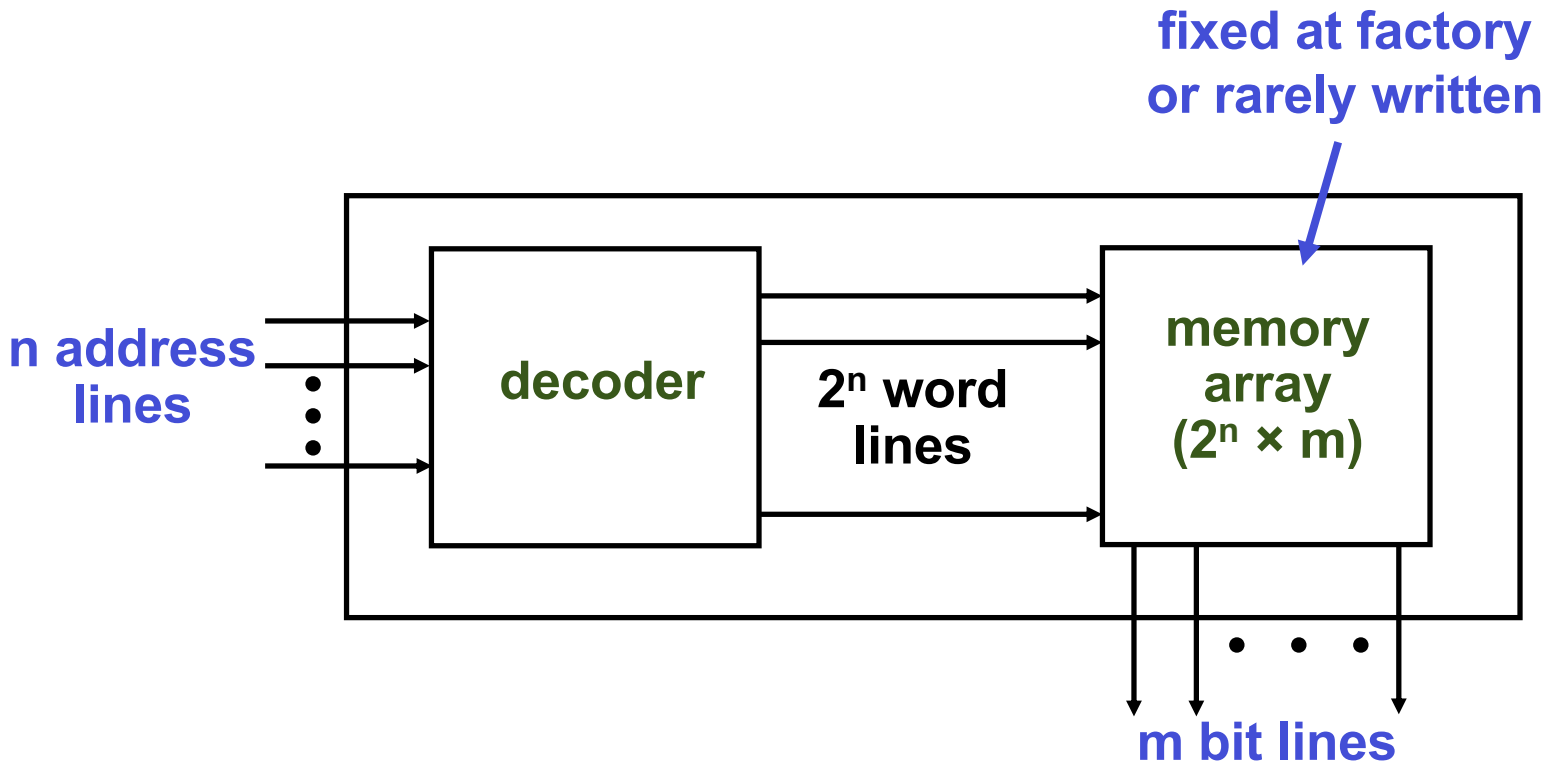
Only one wordline is asserted at any given time

Types of Memories

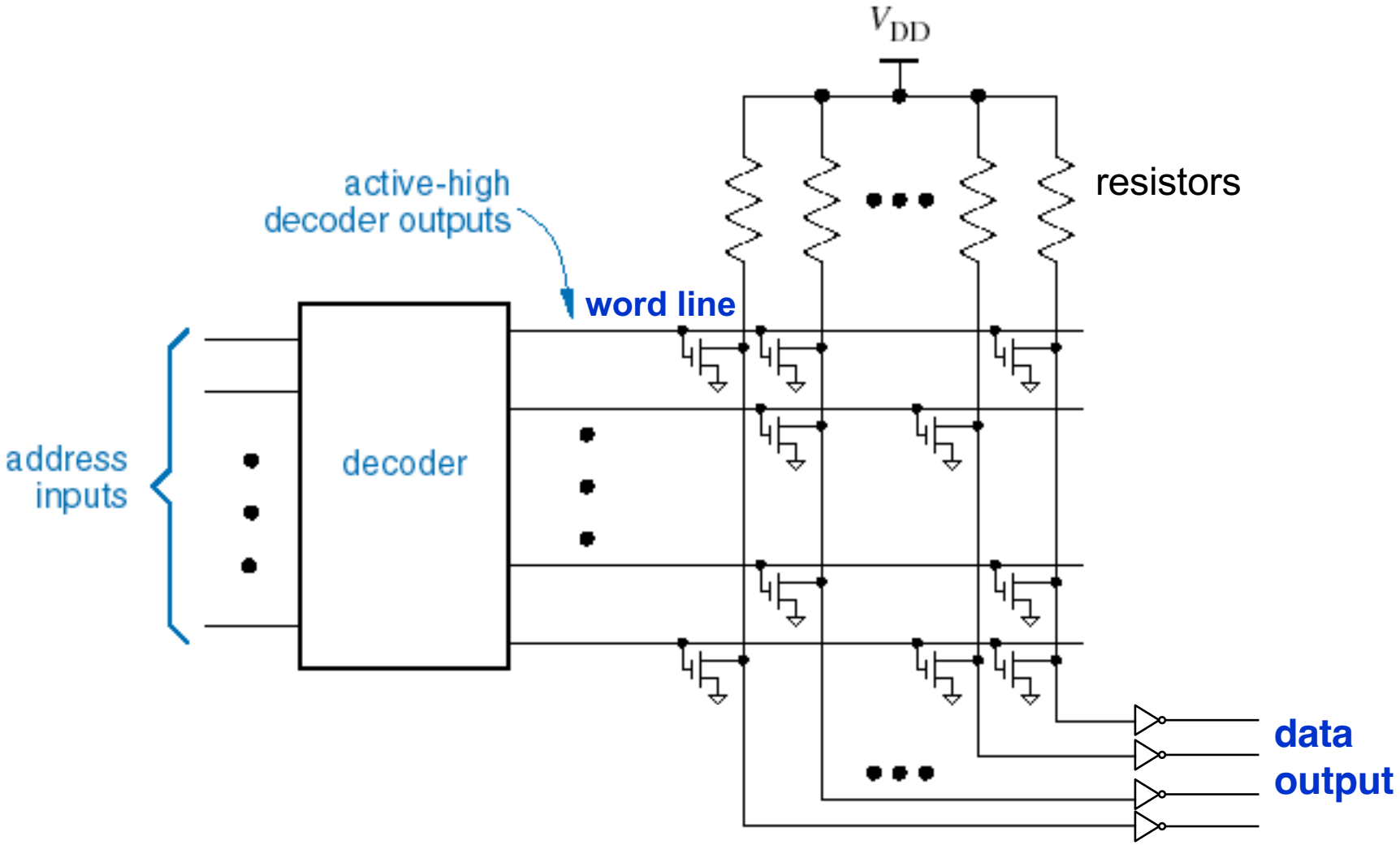
- **Read-Only Memory (ROM)**
  - Truly read-only
    - Written in the factory, and never written after installation
  - Mostly read and rarely written
    - Much faster to read than write
  - Usually non-volatile

- **Random Access Memory (RAM)**
  - Dynamic RAM (DRAM), Static RAM (SRAM)
  - Read and write any location at similar speeds
  - Typically volatile: loses contents when powered off
    - New classes of non-volatile/persistent RAMs are emerging

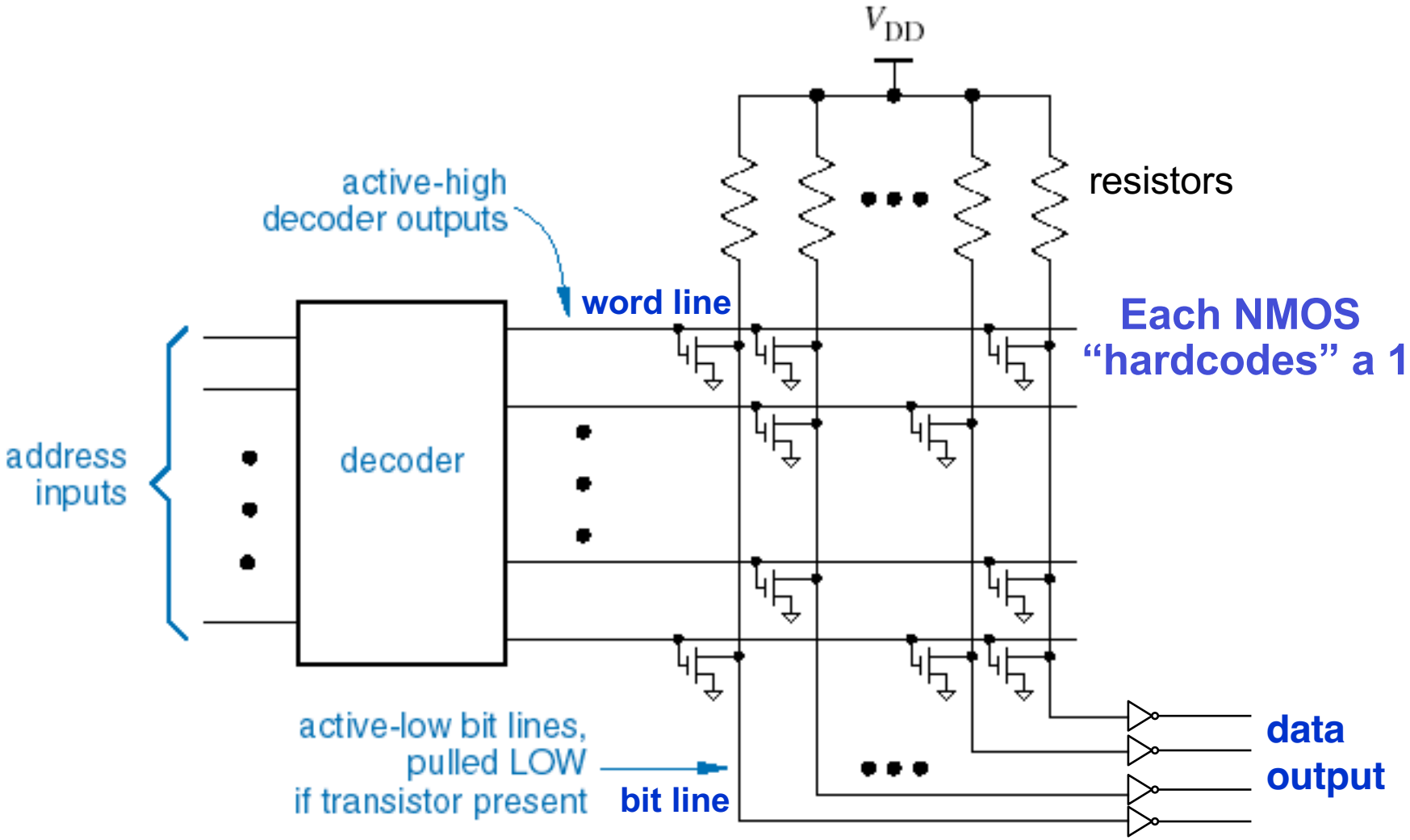
Read-Only Memory (ROM) Structure



Example ROM Implementation



Example ROM Implementation



Applications of ROM

- **Data & program storage**
  - e.g., Boot configuration for personal computers or application code for embedded systems

- **Combinational logic functions**
  - Lookup table
    - Address inputs = function inputs
    - Data outputs = function outputs

Using ROMs for Combinational Logic

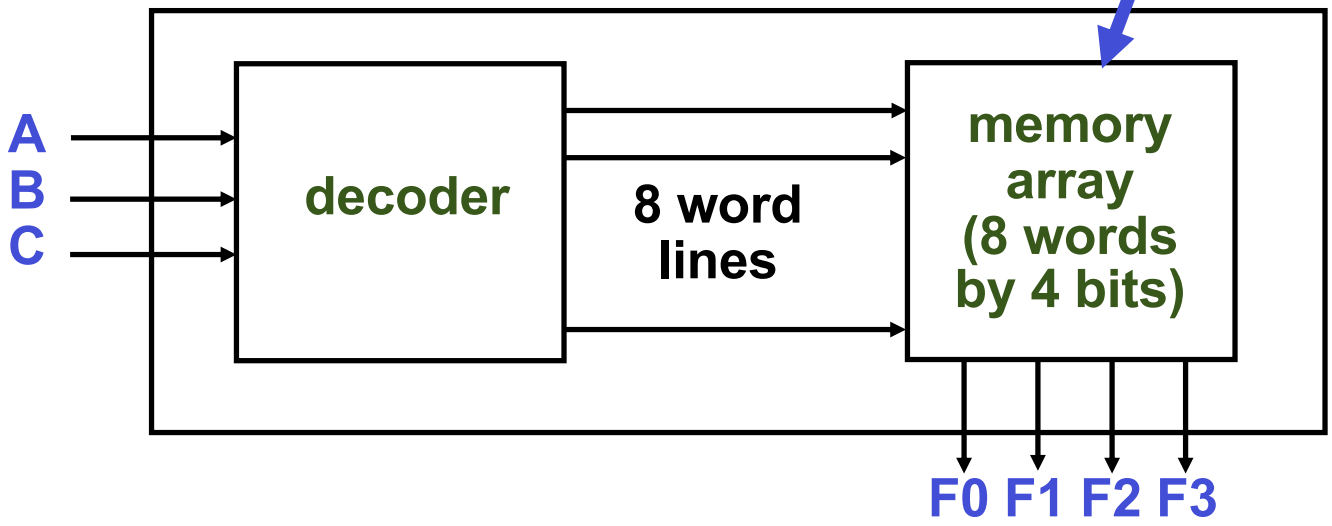
F0 = A' B' C + A B' C' + A B' C

F1 = A' B' C + A' B C' + A B C

F2 = A' B' C' + A' B' C + A B' C'

F3 = A' B C + A B' C' + A B C'

A	B	C	F0	F1	F2	F3
0	0	0	0	0	1	0
0	0	1	1	1	1	0
0	1	0	0	1	0	0
0	1	1	0	0	0	1
1	0	0	1	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	0	1
1	1	1	0	1	0	0



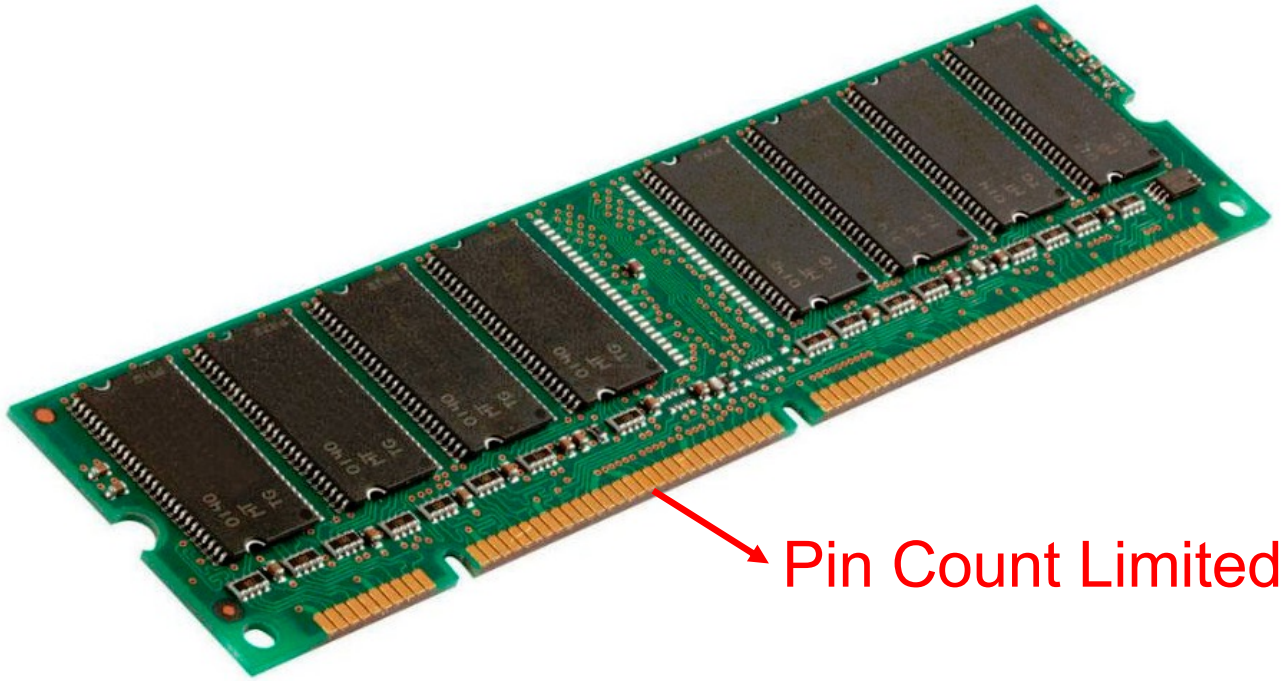
Types of Memories

- **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
  - Usually non-volatile

- **Random Access Memory (RAM)**
  - Dynamic RAM (DRAM), Static RAM (SRAM)
  - Read and write any location at similar speeds
  - Typically volatile: loses contents when powered off
    - Note: new emerging classes of non-volatile/persistent RAMs are emerging

Dynamic RAMs (DRAMs)

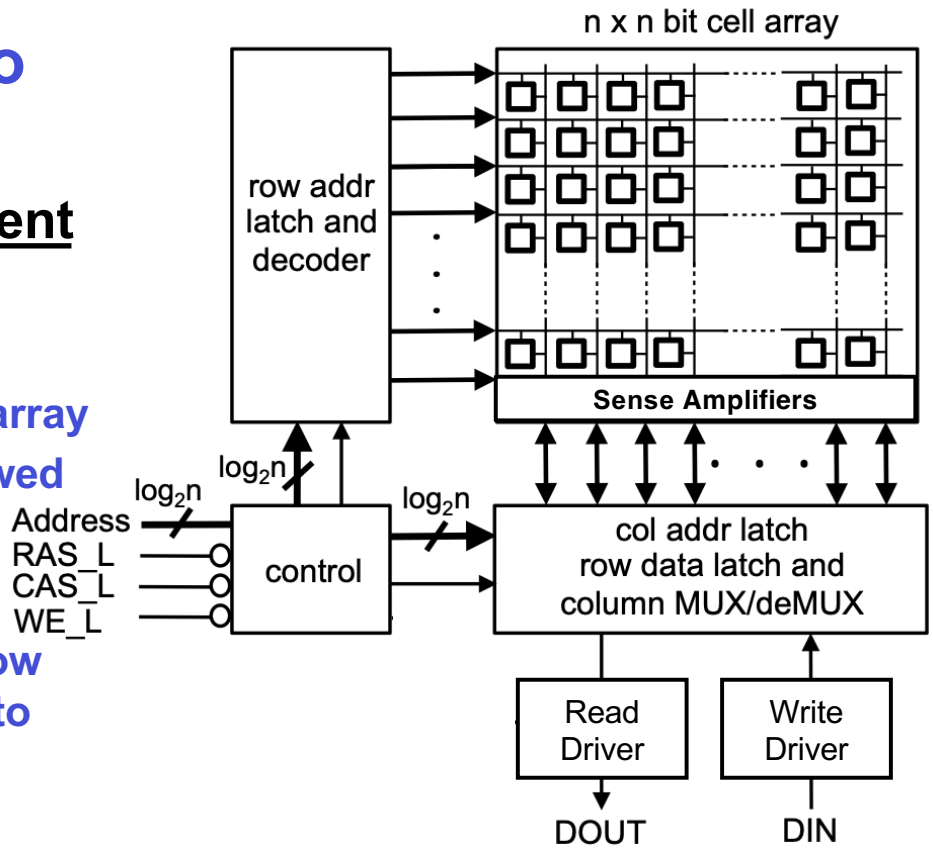
- DRAM modules are usually off-chip and mounted on a separate printed circuit board



- In a typical DRAM, the smallest unit is the memory cell, which forms arrays
  - Arrays make up banks; banks are organized into ranks

Common Organization of a DRAM Array*

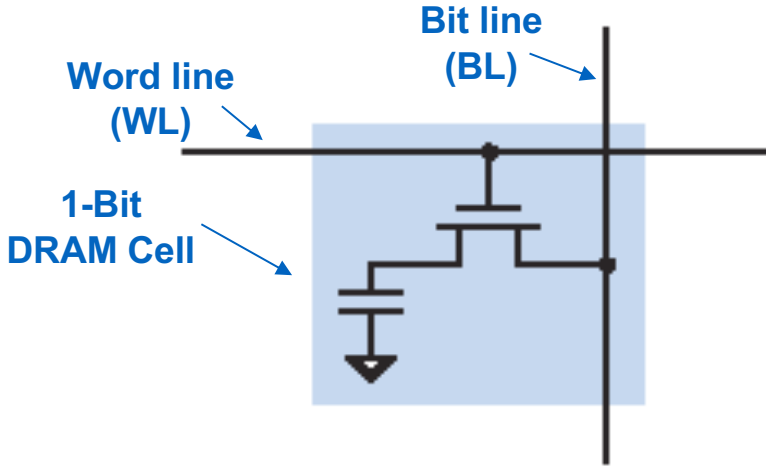
- **Multiplexed address inputs to reduce pin count**
  - **Row & column addresses are sent sequentially through a shared subset of pins**
    - **$\log_2$N bits each for an NxN bit cell array**
    - **Row address bits arrive first; followed by column address**
  - **Row access**
    - **Row-address strobe (RAS) loads row address bits into "row addr" latch to select 1 row**
  - **Column access**
    - **Column-address strobe (CAS) loads column address bits into "col addr" latch to select 1 column bit**



*Optional reading: "How Does DRAM Work?" by Branch Education

DRAM Bit Cell

- Capacitor accessed through a transistor
- Capacitor is charged or discharged to produce a 1 or a 0
- DRAM cells loses their state overtime and must be refreshed periodically, hence the name *dynamic*

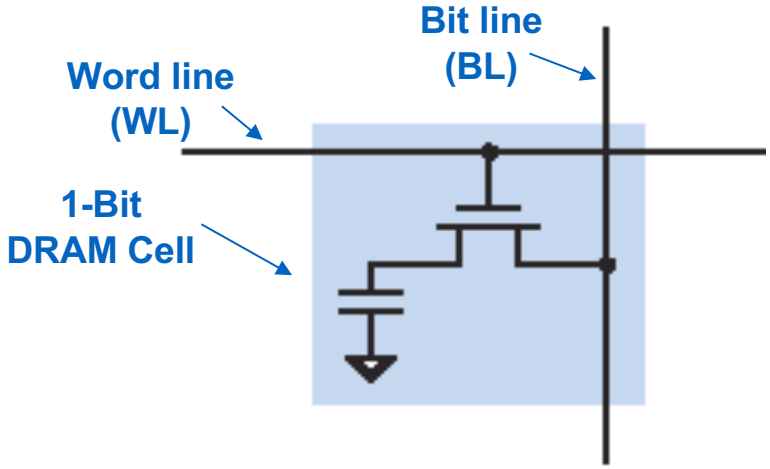


1-Transistor 1-Capacitor (**1T1C**) DRAM Cell

Write a DRAM Bit Cell

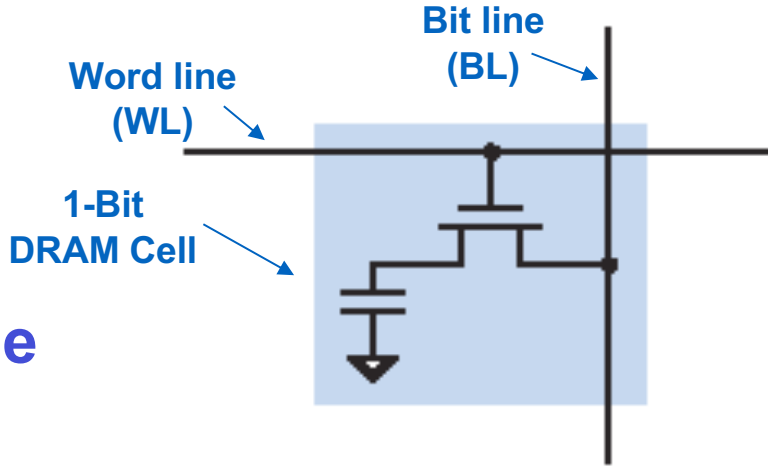
- **Word line is asserted**

- **Bit line is driven with the desired value**
  - **Capacitor is charged by the bit line to store a 1**
  - **Or discharged by the bit line to store a 0**



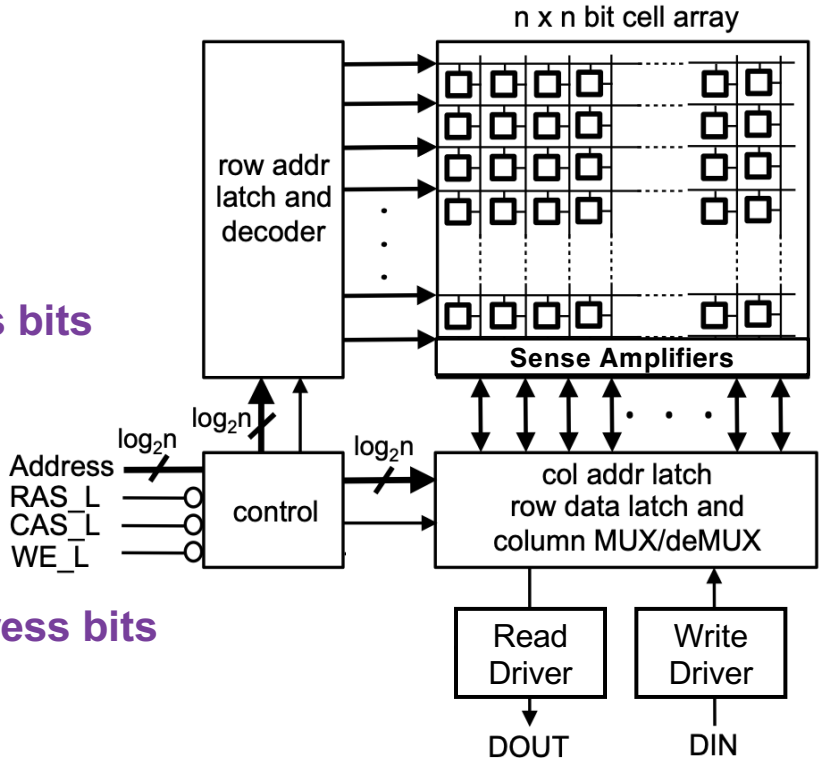
Read a DRAM Bit Cell

- Word line is asserted
- Bit line is *precharged* halfway between 0 and 1



- Capacitor voltage pulls the bit line slightly higher or lower
  - This also destroys the stored value (which the sense amplifier restores later)
- Sense amplifier detects this small change and amplifies it
  - If the bit line voltage increases, the sense amplifier drives it high (1); otherwise, it pulls it to ground (0).

Example: Read Access to a DRAM Array



A DRAM bit cell array with 1024 rows x 1024 columns (1M bits) [1]

(1) **10 row address bits arrive first;**

RAS asserted

(2) **1024 bits are read out to row data latch**

(3) **10 column address bits arrive later;**

CAS asserted

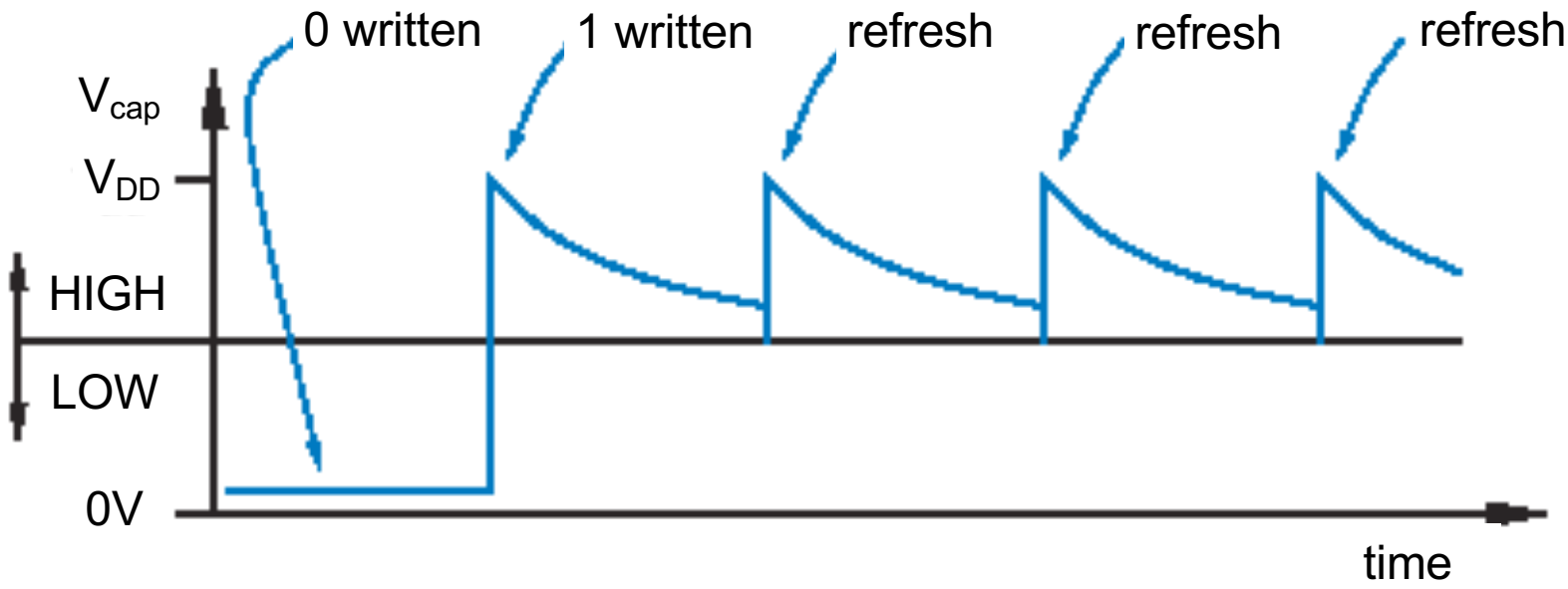
(4) **1 bit returned through column MUX**

To retrieve a word, multiple DRAM arrays are read in parallel [1]

[1] In a modern DRAM, a bit cell array typically has more rows than columns (e.g., 64K rows x 8K columns). The columns are organized into groups of memory cells (typically group size=8), and all cells in a group share the same address and are read or written simultaneously.

DRAM Refresh

- **Capacitors discharge over time**
  - **Bit cells must be periodically refreshed (e.g., every 64ms, technology dependent)**
- ***Refresh* cycles recharge each memory bit**
  - **An entire row is refreshed at a time – each row periodically accessed using RAS (i.e., read access), which restores the charge**

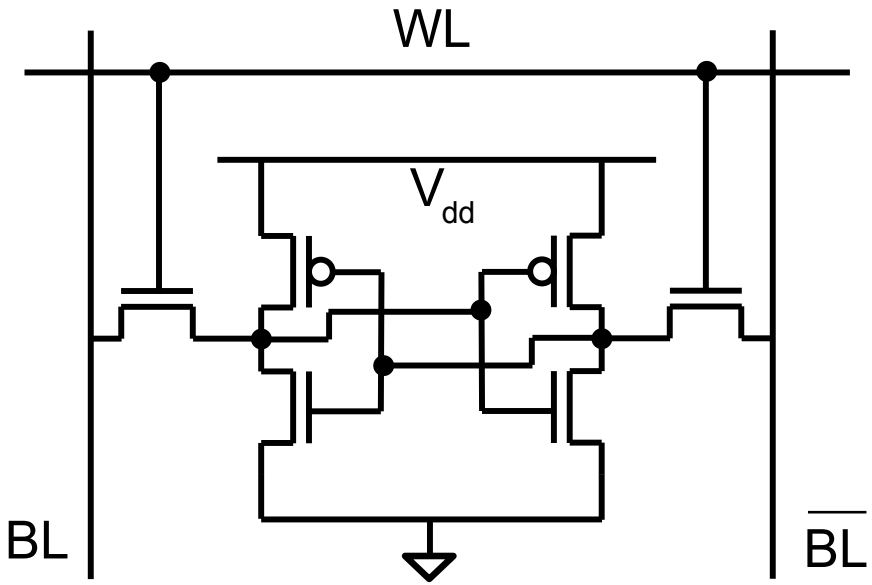


Static RAM (SRAM) Bit Cell

- **One cell requires six transistors**
  - **The core is two cross-coupled inverters**
- **Maintaining the state of the cell requires a constant power**
  - **The cell is stable; no refresh cycles needed**

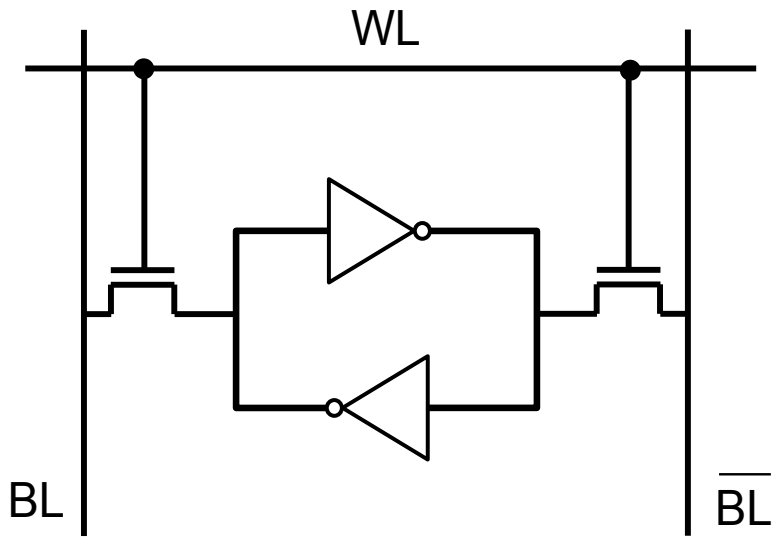
6-Transistor (**6T**)
SRAM Cell

BL = bit line
WL = word line



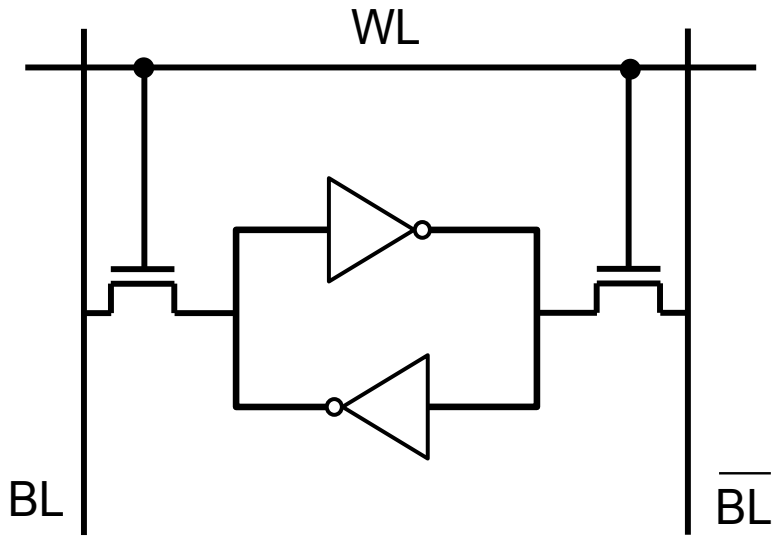
SRAM Write

- Drive one bit line high, the other low (depending on the desired value)
- Then turn on word line
- Bit lines over power cell with new value

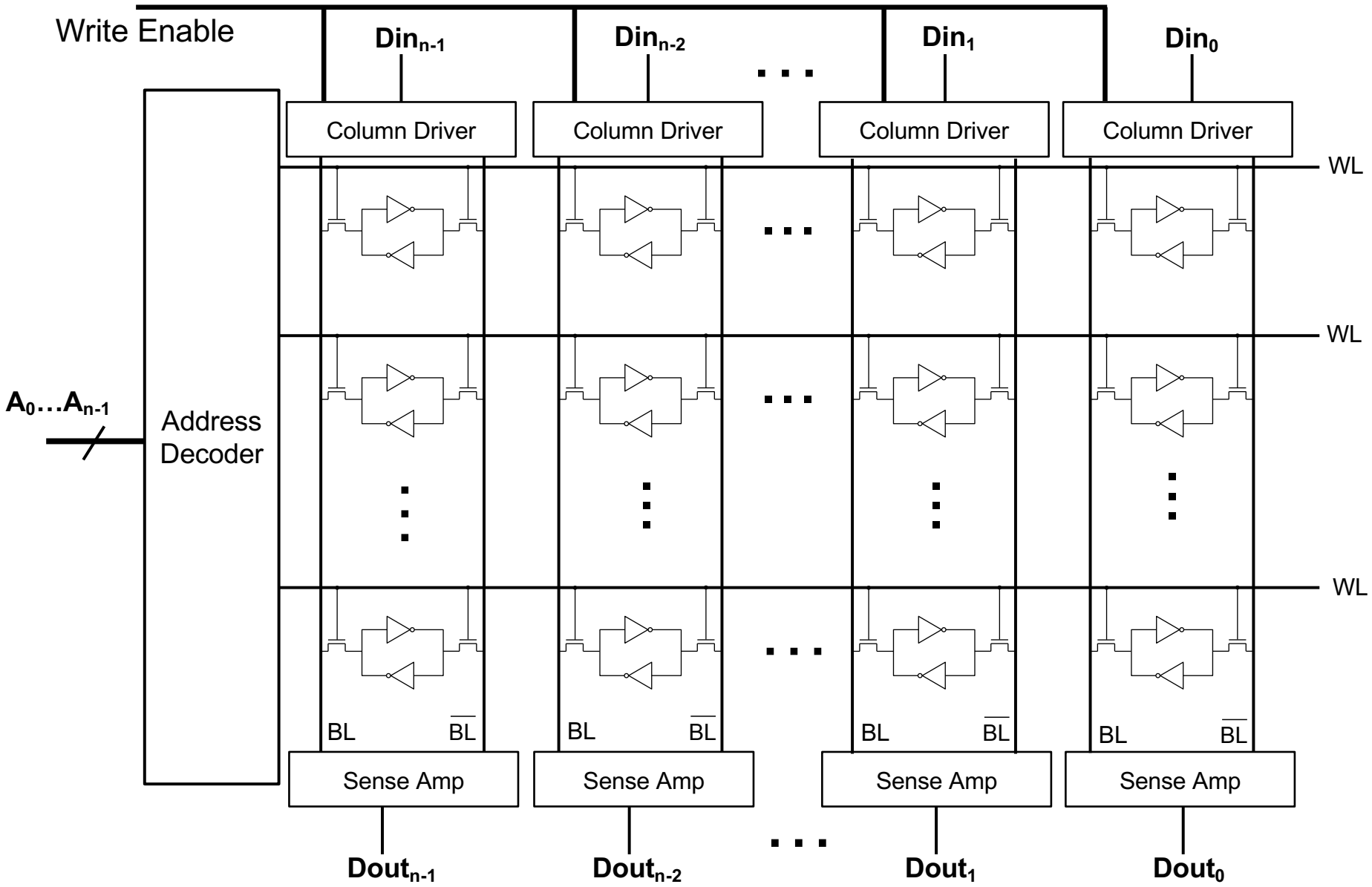


SRAM Read

- Precharge both bit lines high
- Then turn on word line
- One of the two bit lines will be pulled down by the cell
  - Change detected by sensor amplifier



SRAM Architecture*



*Optional reading: [Understanding SRAM](#) by SemiEngineering

Next Class

Single Cycle Microprocessor
(H&H 7.1-7.3.4)