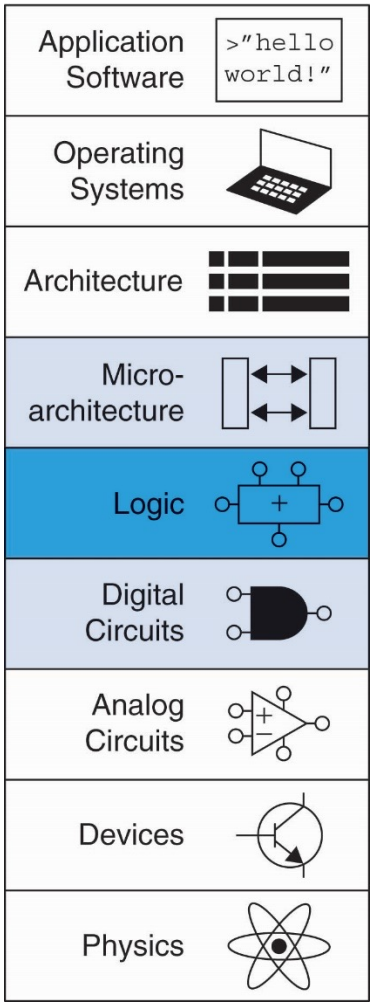
Chapter 5 :: Digital Building Blocks

Digital Design and Computer Architecture

David Money Harris and Sarah L. Harris

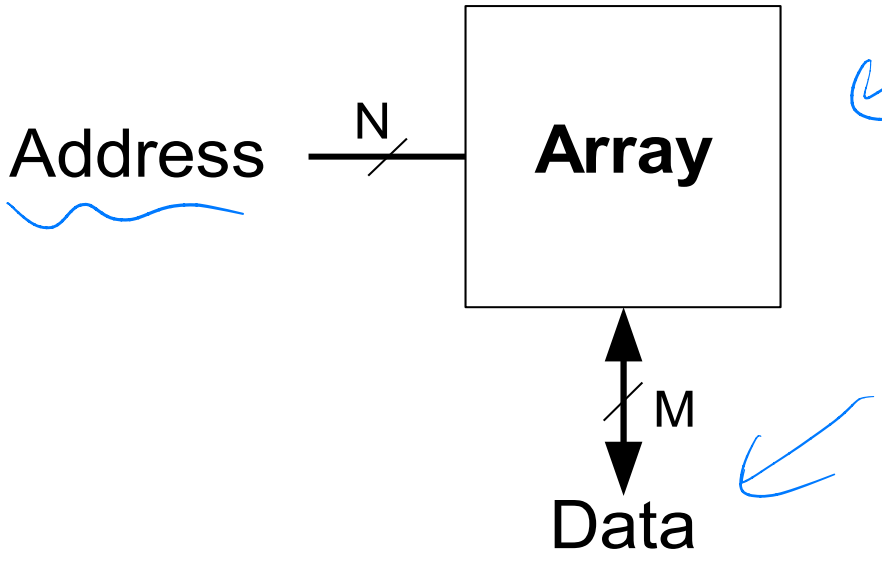
Chapter 5 :: Topics

- 5.1 Introduction
- 5.2 Arithmetic Circuits
- 5.3 Number Systems
- 5.4 Sequential Building Blocks
- 5.5 Memory Arrays
- 5.6 Logic Arrays



5.5 Memory Arrays

- Efficiently store large amounts of data
- Three common types:
  - Dynamic random access memory (DRAM)
  - Static random access memory (SRAM)
  - Read only memory (ROM)
- An *M*-bit data value can be read or written at each unique *N*-bit address.

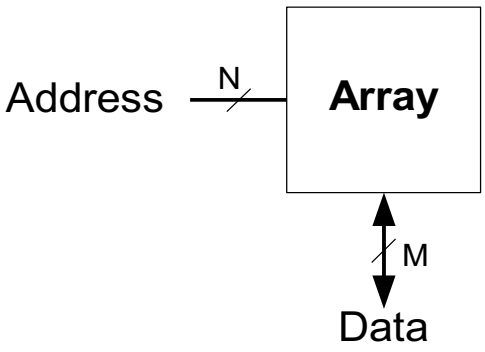


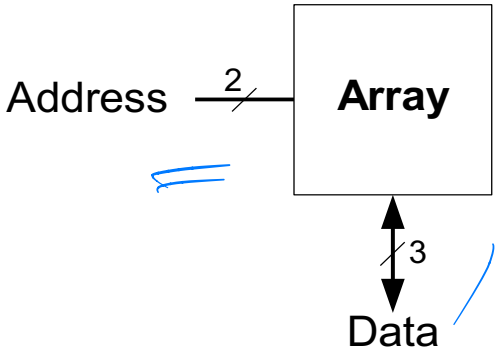
Memory Arrays

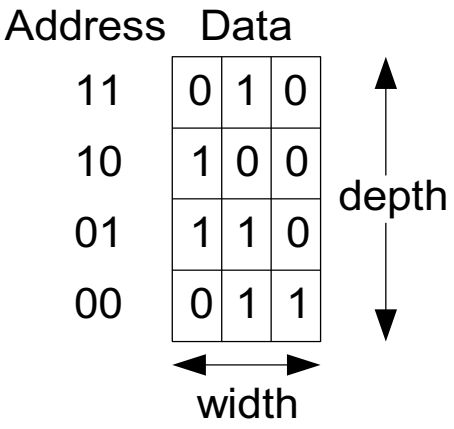
(spreadsheet matrix)

- Two-dimensional array of bit cells
- Each bit cell stores one bit
- An array with N address bits and M data bits:
  - 2^N rows and M columns
  - **Depth:** number of rows (number of words)
  - **Width:** number of columns (size of word)
  - **Array size:** depth × width = $2^N \times M$

word = 1 "unit" of data



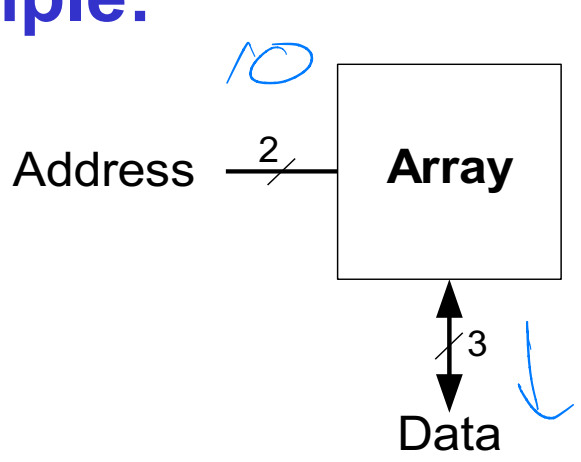


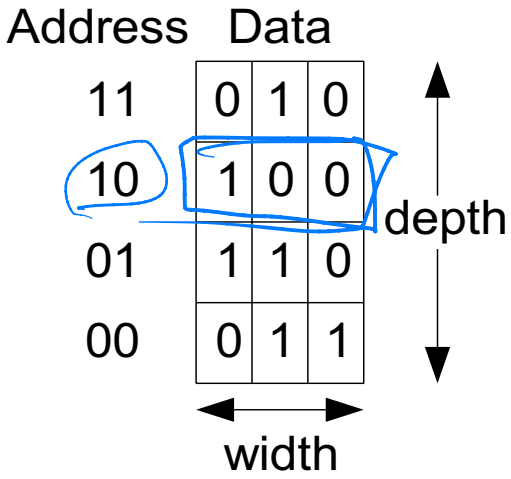


Memory Array: Example

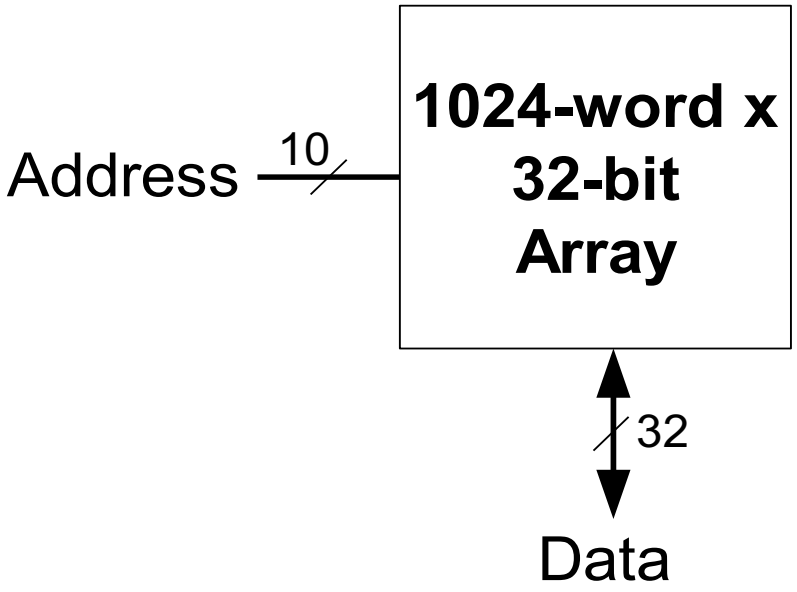
- $2^2 \times 3$-bit array
- Number of words: 4
- Word size: 3-bits
- For example, the 3-bit word stored at address 10 is 100

Example:





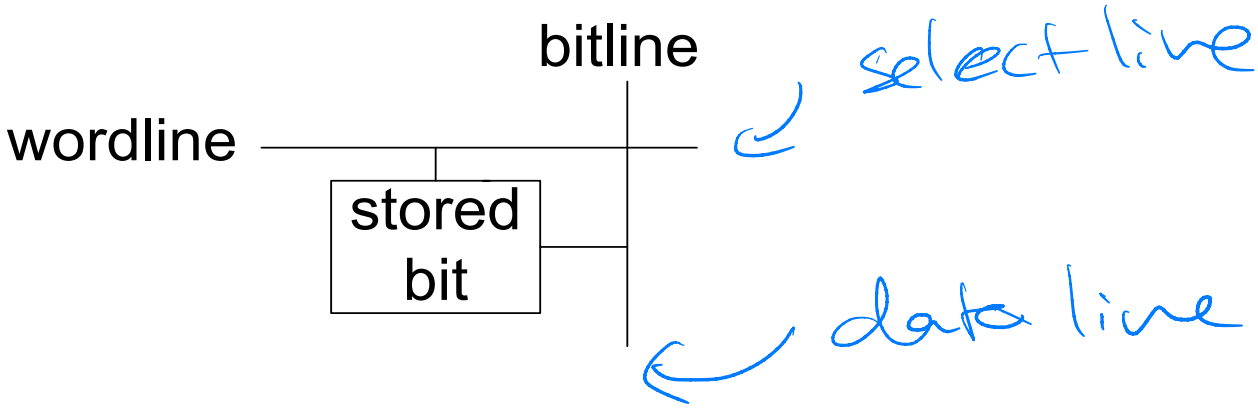
Memory Arrays



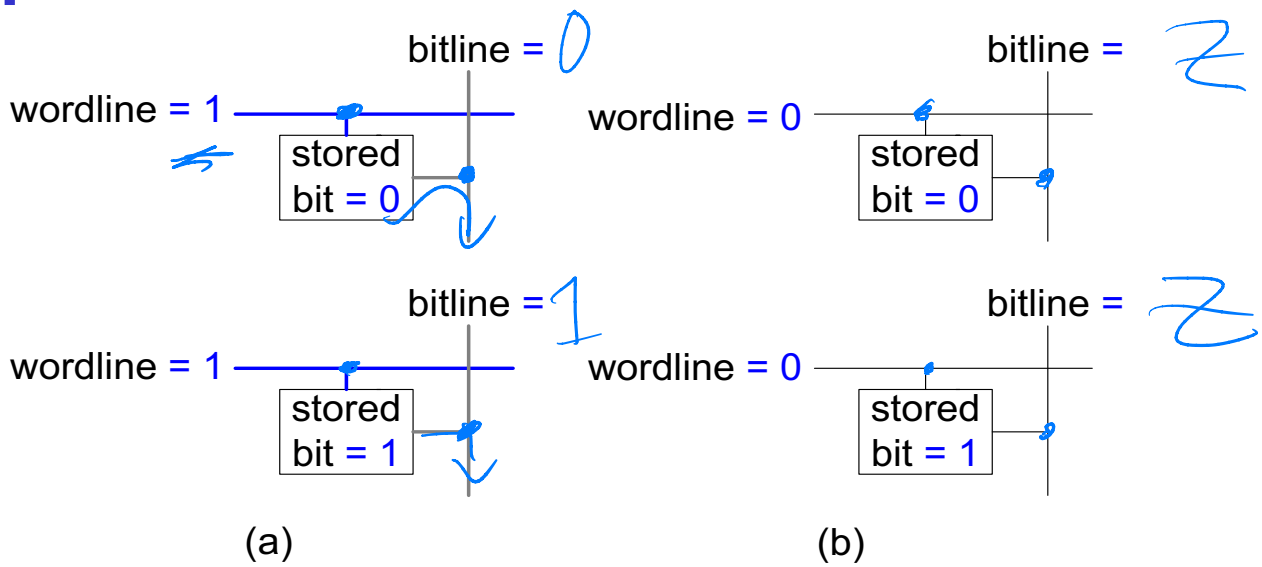
1024 x 32 ≃ 32 Kb

= 4 KB 1 Byte = 8 bits

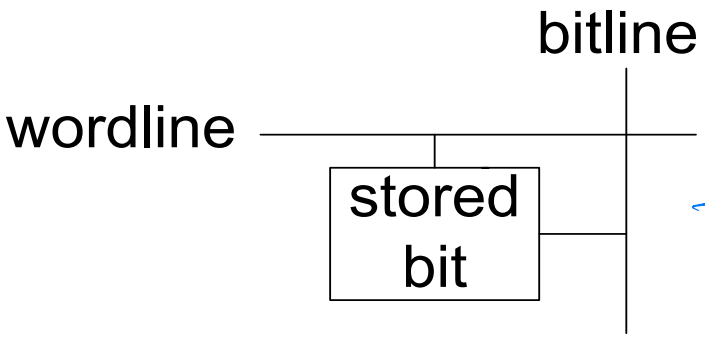
Memory Array Bit Cells



Example:



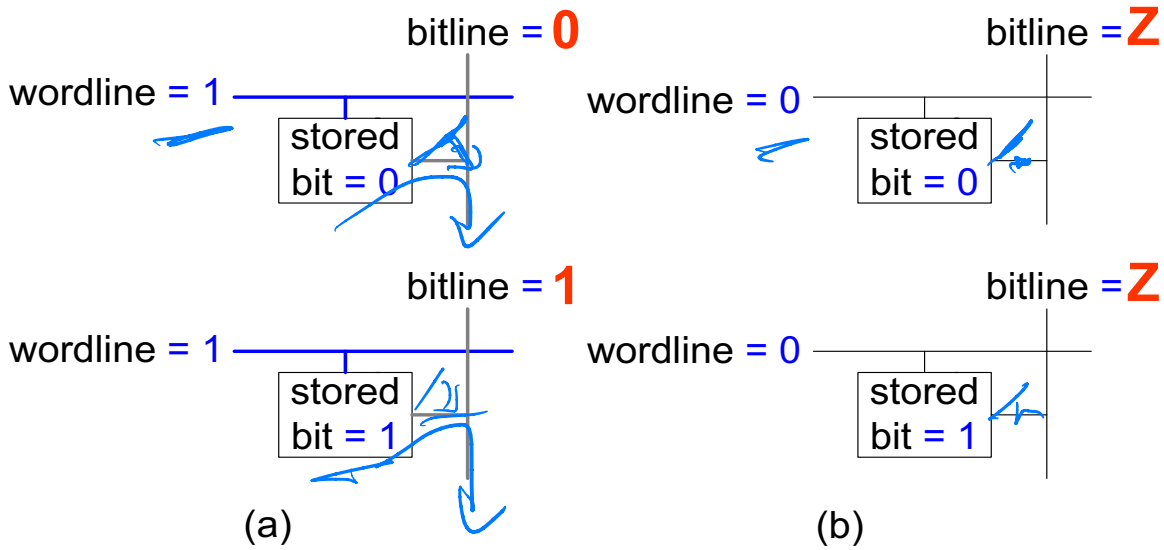
Memory Array Bit Cells



To read a bit cell, the bitline is initially left floating(Z). Then the wordline is turned on, allowing th stored value to drive the bitline to 0 or 1.

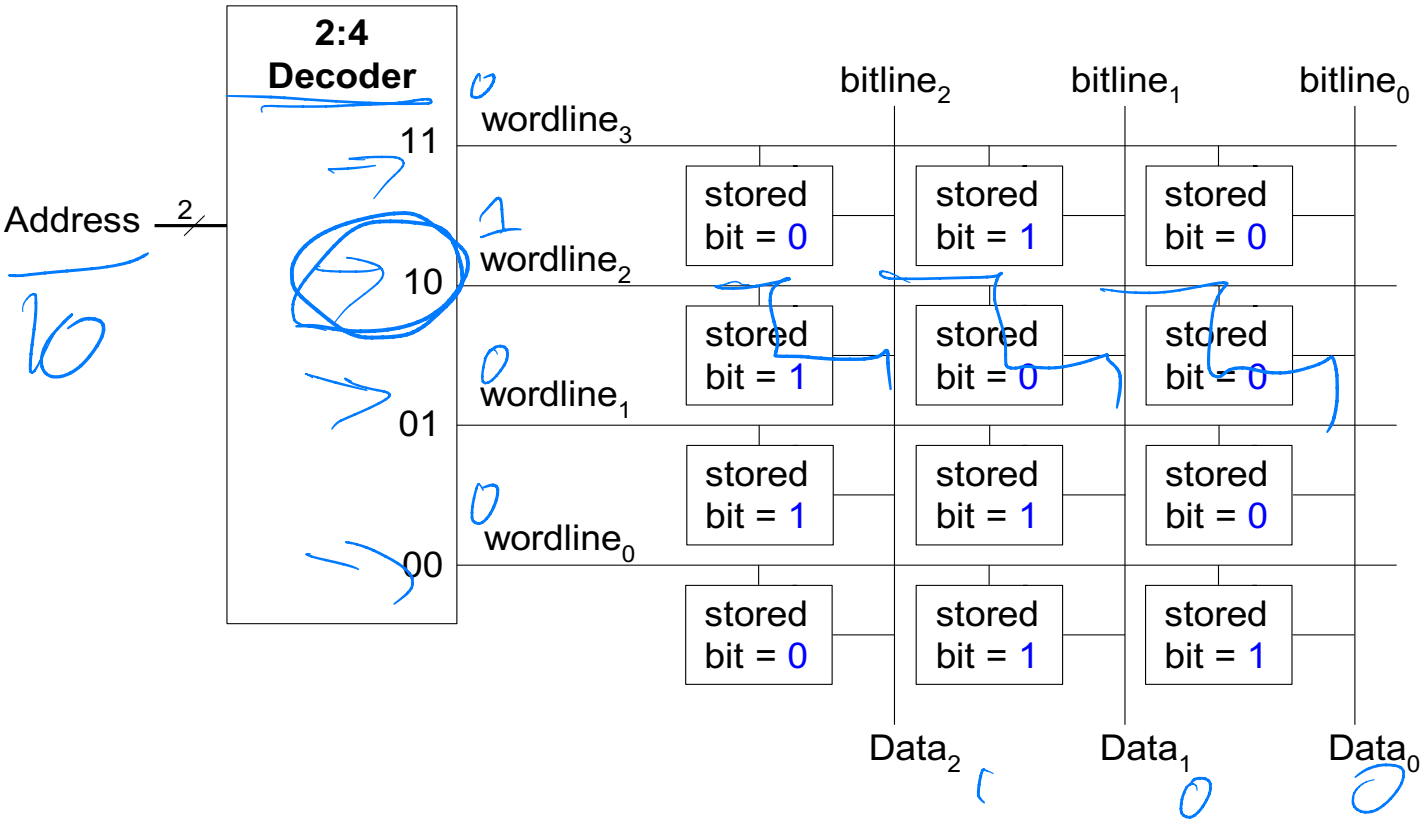
To write a bit cell, the bitline is strongly driven to the desired value. Then the wordline is turned ON, the bitline overpowers the contents of the bitcell.

Example:



Memory Array

- **Wordline:**
  - similar to an enable
  - allows a single row in the memory array to be read or written
  - corresponds to a unique address
  - only one wordline is HIGH at any given time



Types of Memory

- Random access memory (RAM): volatile
- Read only memory (ROM): nonvolatile

needs to be powered

maintains state w/o power

RAM: Random Access Memory

- **Volatile:** loses its data when the power is turned off
- Read and written quickly
- Main memory in your computer is RAM (DRAM)

Historically called *random* access memory because any data word can be accessed as easily as any other (in contrast to sequential access memories such as a tape recorder)

ROM: Read Only Memory

- **Nonvolatile:** retains data when power is turned off
- Read quickly, but writing is impossible or slow
- Flash memory in cameras, thumb drives, and digital cameras are all ROMs

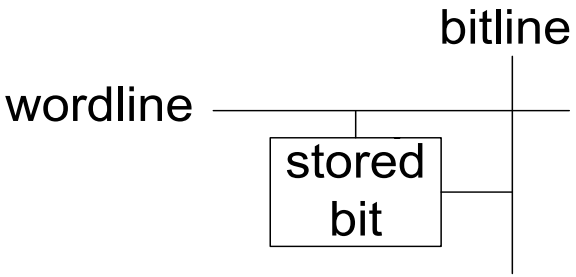
Historically called *read only* memory because ROMs were written at manufacturing time or by burning fuses. Once ROM was configured, it could not be written again. This is no longer the case for Flash memory and other types of ROMs.

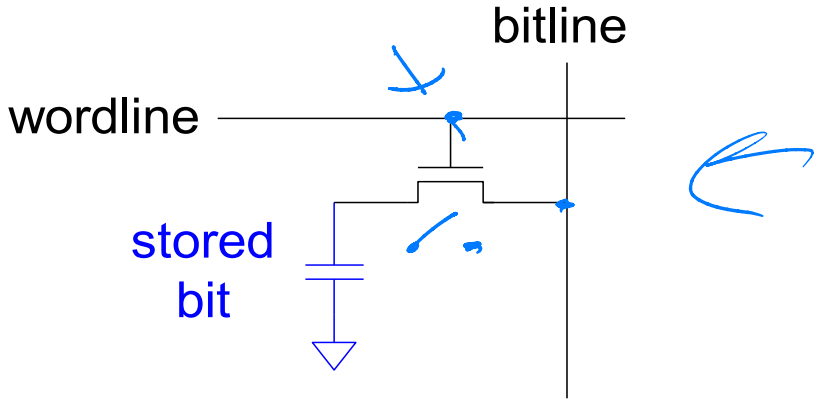
Types of RAM

- Two main types of RAM:
  - Dynamic random access memory (DRAM)
  - Static random access memory (SRAM)
- Differ in how they store data:
  - DRAM uses a capacitor
  - SRAM uses cross-coupled inverters

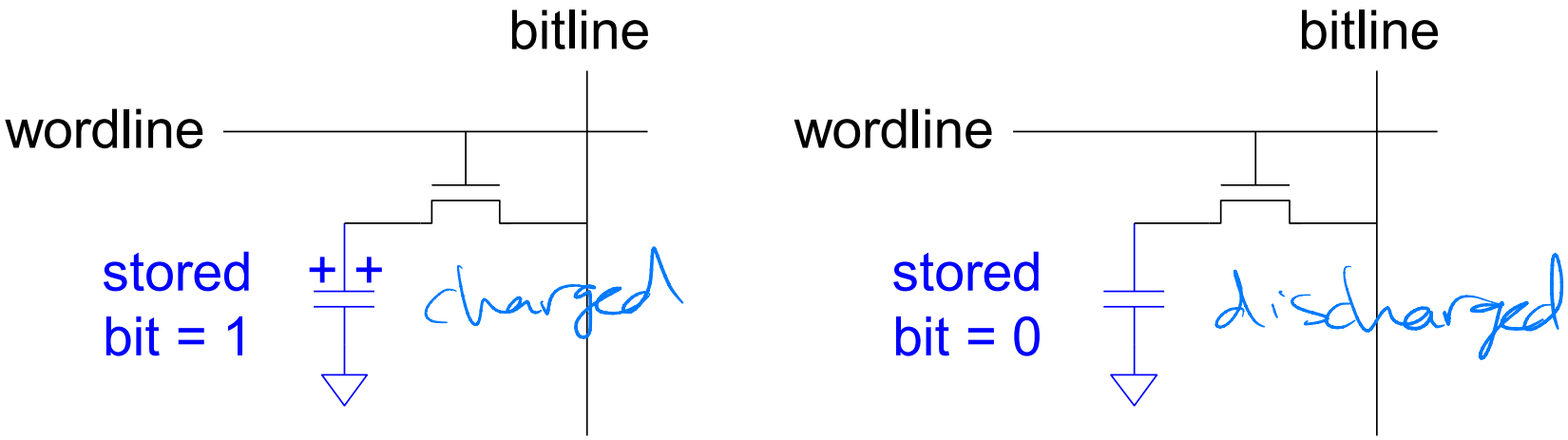
5.5.2 DRAM

- Data bits stored on a capacitor
- Called *dynamic* because the value needs to be refreshed (rewritten) periodically and after being read:
  - Charge leakage from the capacitor degrades the value
  - Reading destroys the stored value

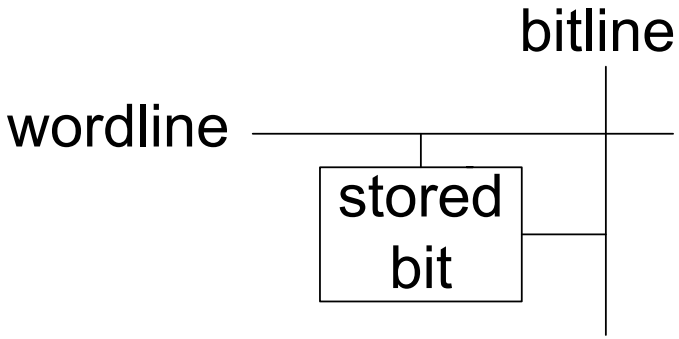


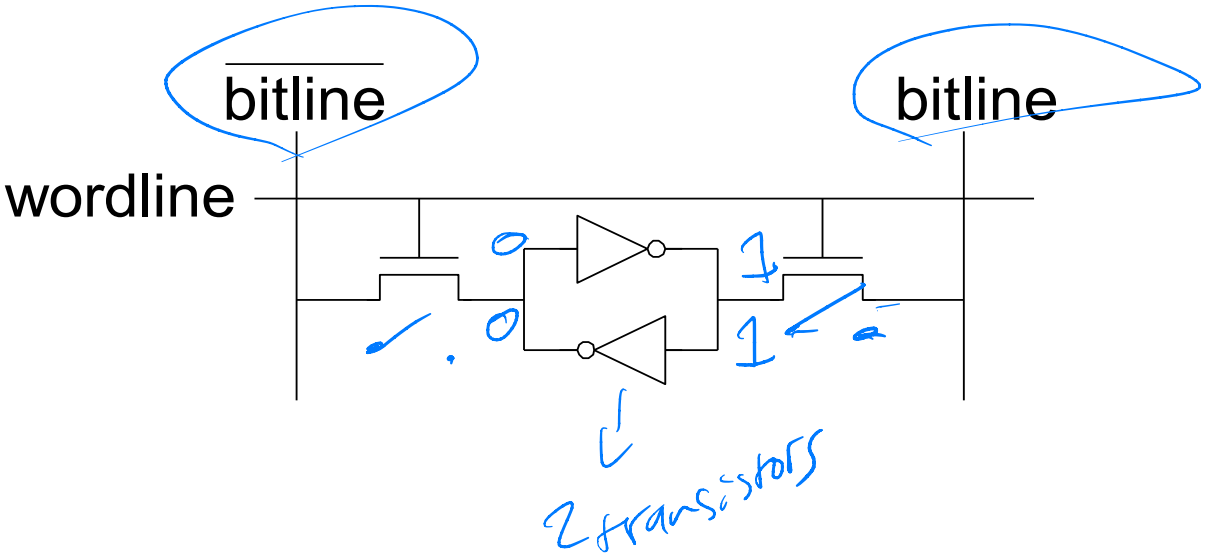


DRAM

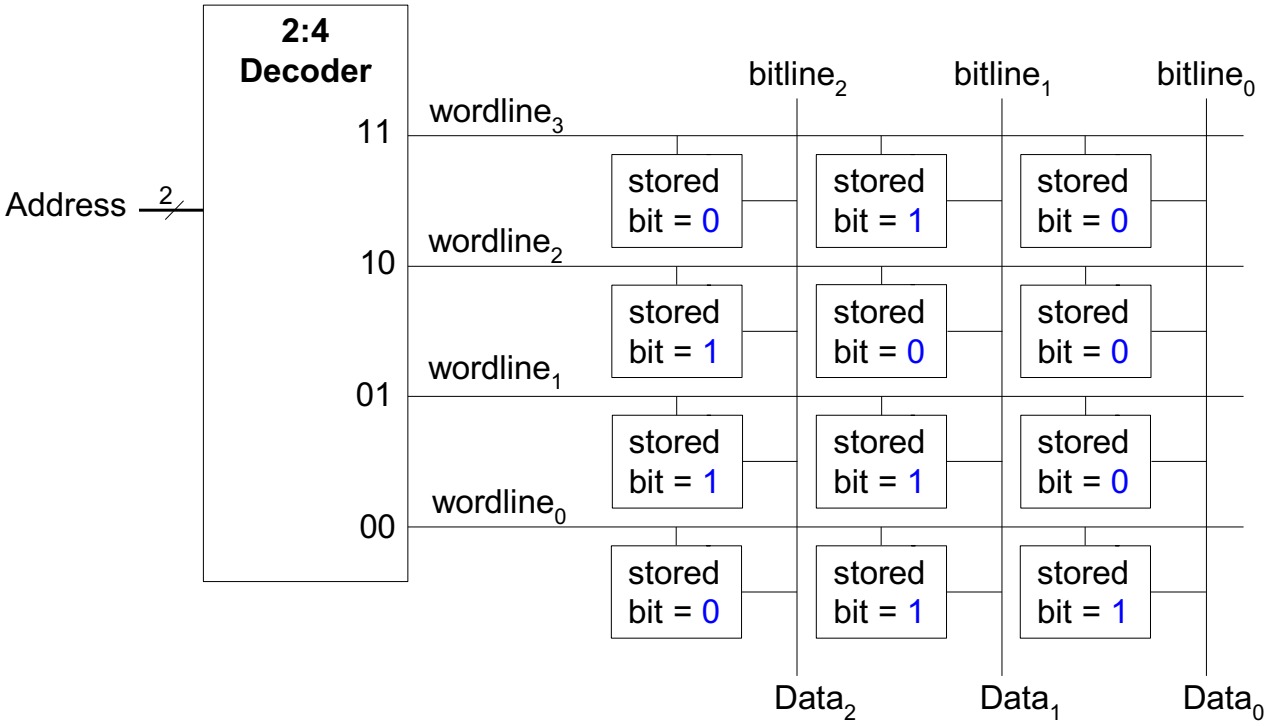

bitline
wordline
stored
bit = 1
+ +
charged
bitline
wordline
stored
bit = 0
discharged

5.5.3 SRAM

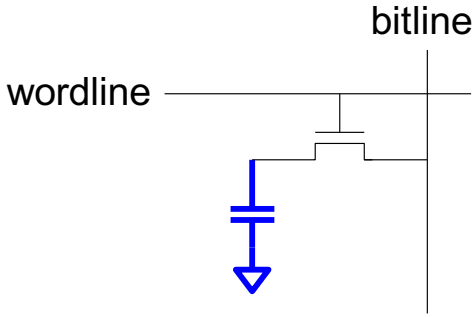

bitline
wordline
stored
bit


bitline
bitline
wordline
0
1
0
1
2 transistors

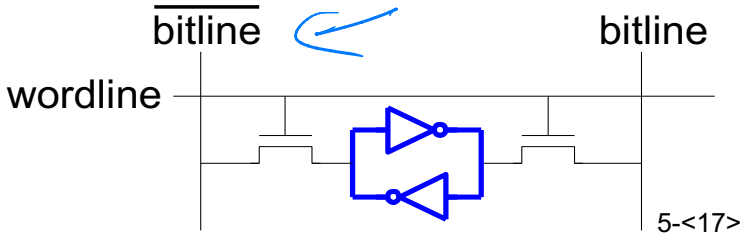
Memory Arrays



DRAM bit cell:



SRAM bit cell:



Memory Arrays – Area vs Delay

- DRAM
  - Slow (refresh)
  - Cheap
  - Large capacity
- SRAM
  - Fast
  - Expensive
  - Small capacity
- Registers
  - Fastest
  - Very expensive
  - Smallest capacity

Memory Ports

All memories have one or more *ports*. Each port gives read and/or write access to one memory address. The previous examples are all single-ported memories.

Multiported memories can access several addresses simultaneously. Figure 5.43 shows a three-ported memory with two read ports and one write port. Port 1 reads the data from address *A1* onto the read data output *RD1*. Port 2 reads the data from address *A2* onto *RD2*. Port 3 writes the data from the write data input *WD3* into address *A3* on the rising edge of the clock if the write enable *WE3* is asserted.

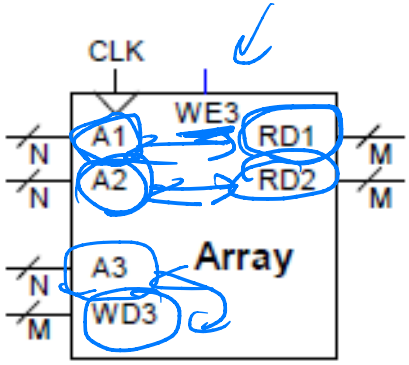
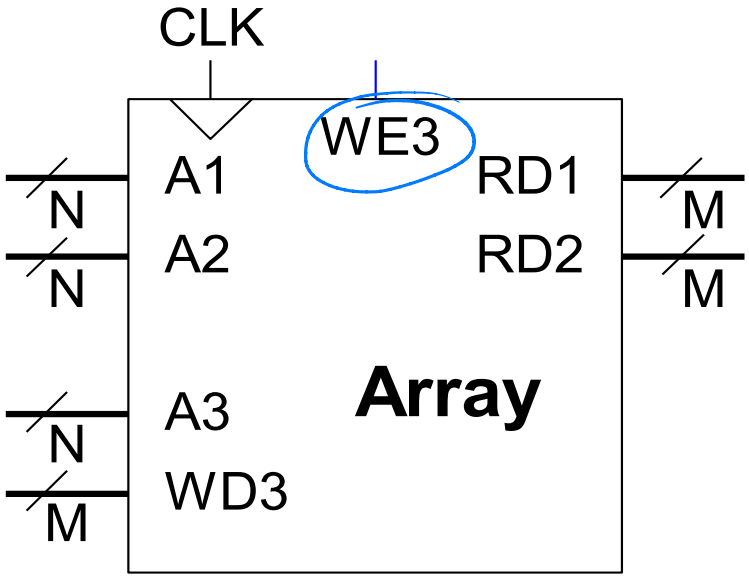


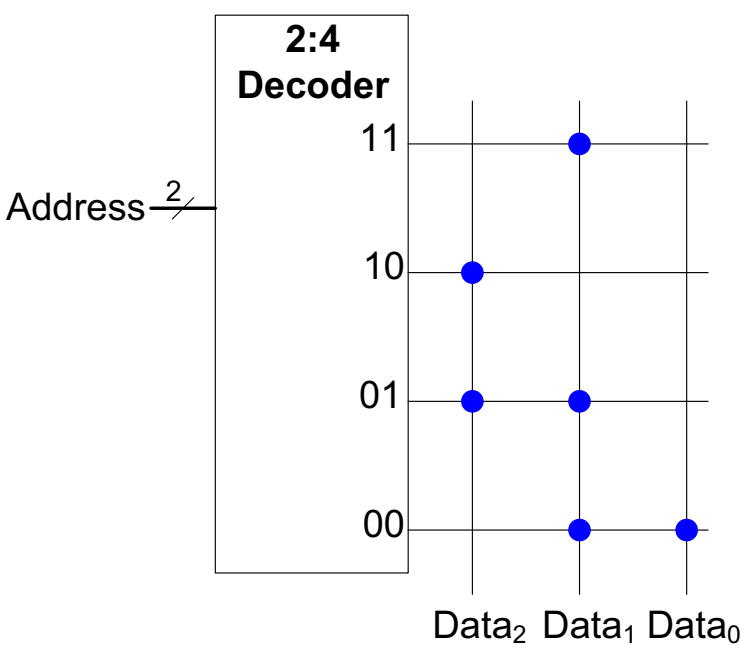
FIGURE 5.43 3-ported memory

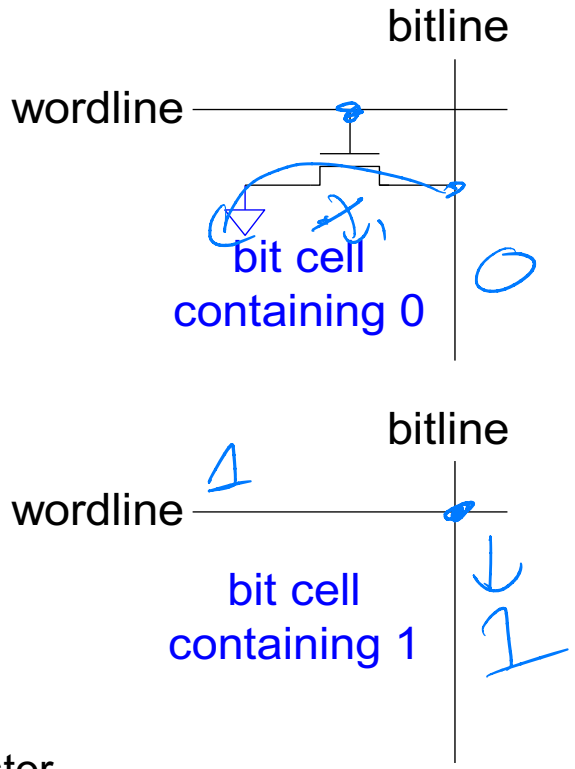
Multi-ported Memories

- **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*
  - Often use SRAM array instead of register bank



5.5.6 ROMs: Dot Notation

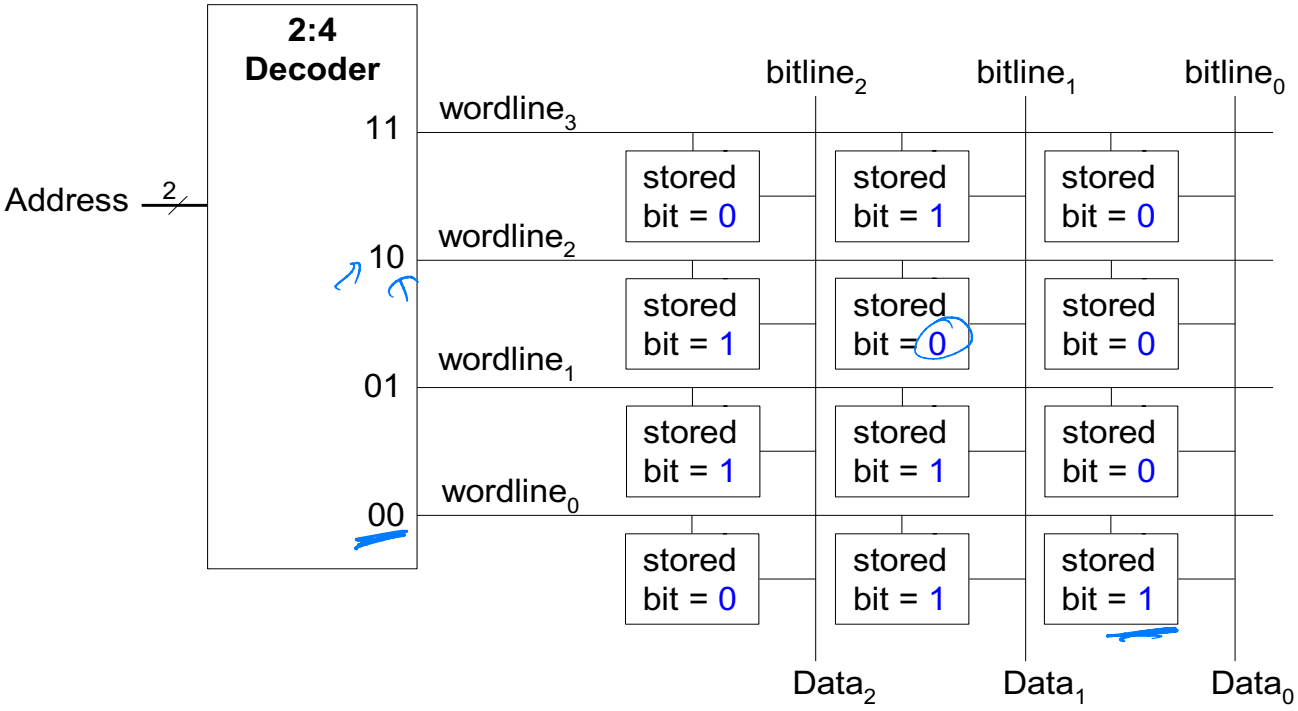




ROM: stores a bit as the presence or absence of a transistor.
To read ROM, bitline is initially set to be weak HIGH
If the transistor is present, it pulls the bitline LOW
If the transistor is absent, the bitline remains High

Dot notation: a dot indicates that the bit is one. Ex: for address: 11: data: 010

5.5.7 Logic with Any Memory Array



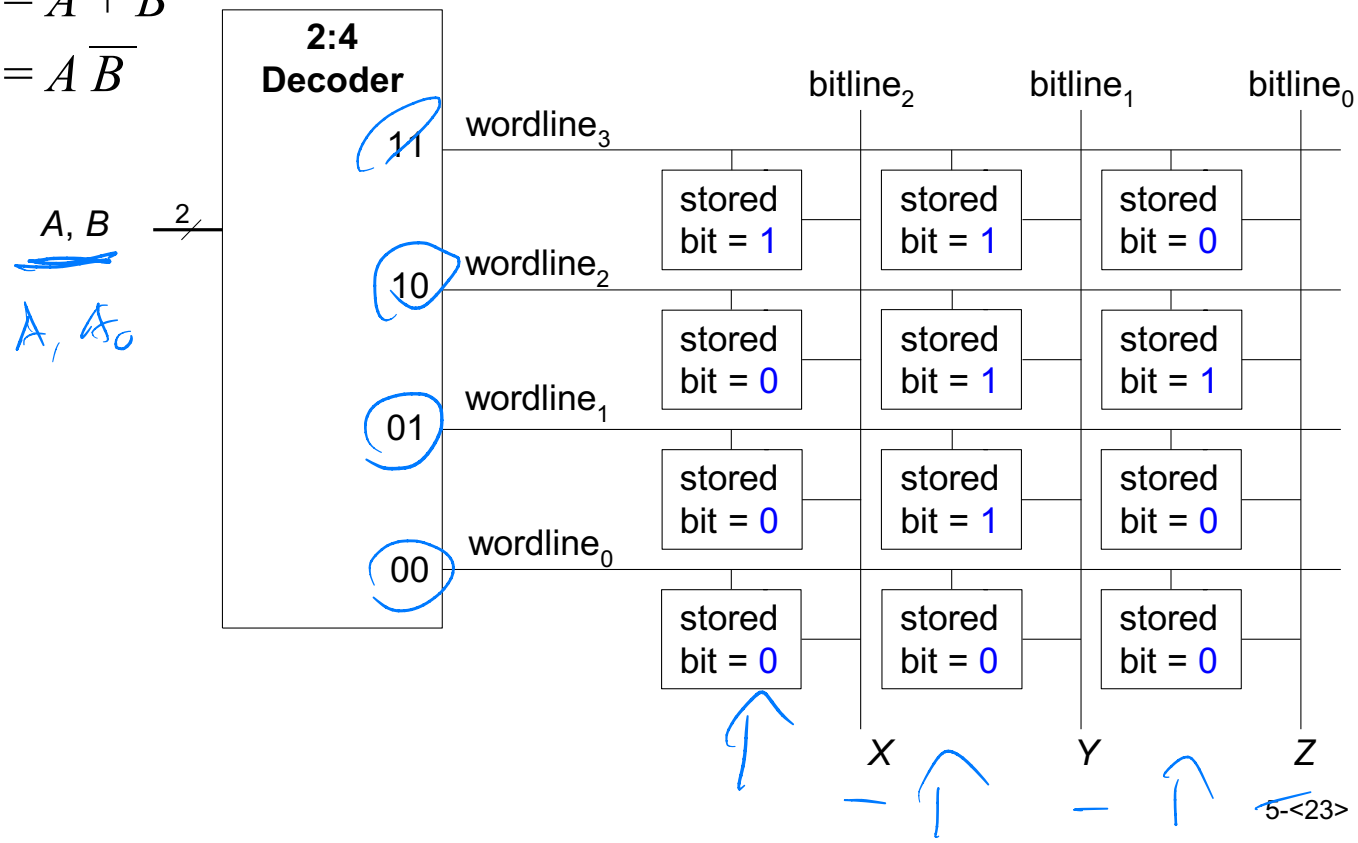
$$Data_2 = A_1 \oplus A_0$$

$$Data_1 = A_1 + A_0$$

$$Data_0 = \overline{A_1}A_0$$

Logic with Memory Arrays

- Implement the following logic functions using a $2^2 \times 3$-bit memory array:
  - $X = AB$
  - $Y = A + B$
  - $Z = A\,\overline{B}$



Logic with Memory Arrays

- Memory arrays used to perform logic is called Lookup table
- ***lookup tables* (LUTs):** look up output at each input combination (address). Look up table is a reflection of logic truth table.

Truth Table

A	B	Y
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**