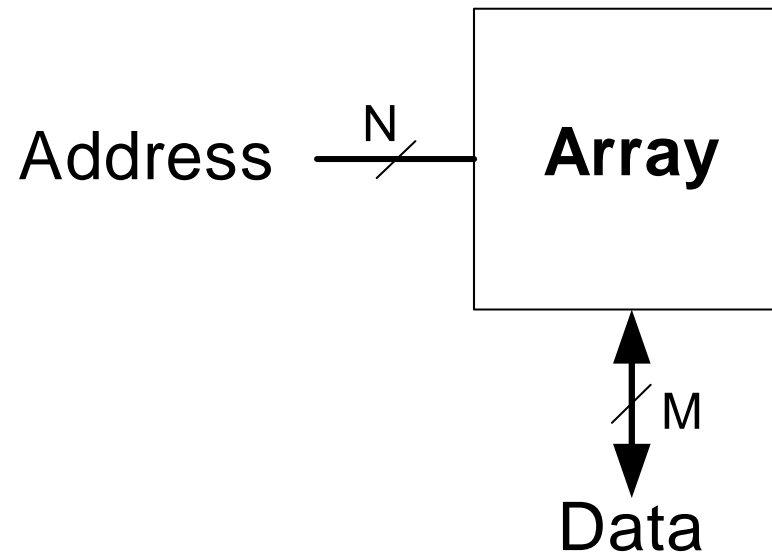


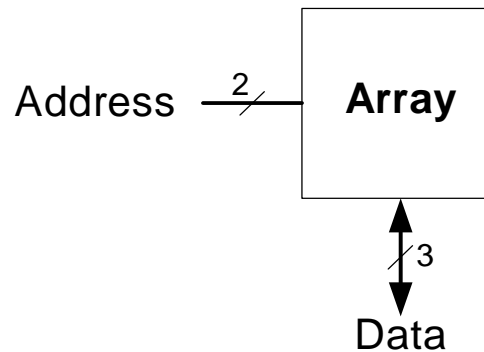
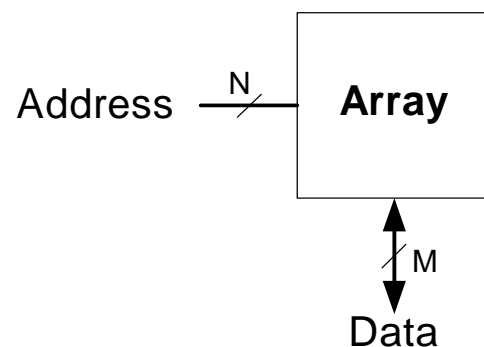
Memory Arrays

- 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

- 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 \times width = $2^N \times M$

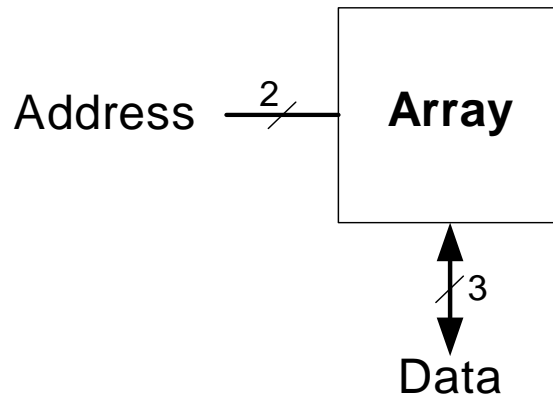


Address	Data			
11	0	1	0	depth ↑ ↓
10	1	0	0	
01	1	1	0	
00	0	1	1	
	width ←→			

Memory Array: Example

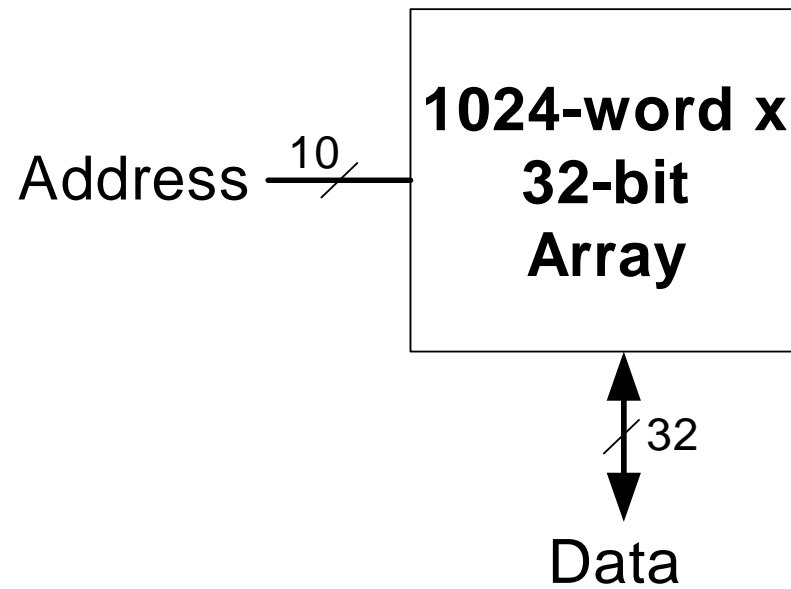
- $2^2 \times 3$ -bit array
- Number of words: 4
- Word size: 3-bits
- For example, the 3-bit word stored at address 10 is 100

Example:

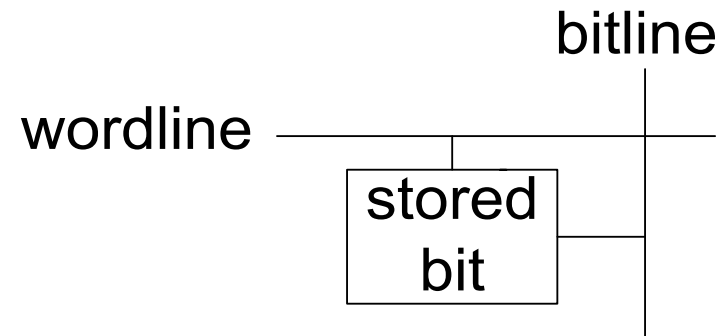


Address	Data			
11	0	1	0	depth ↑ ↓
10	1	0	0	
01	1	1	0	
00	0	1	1	
	width ←→			

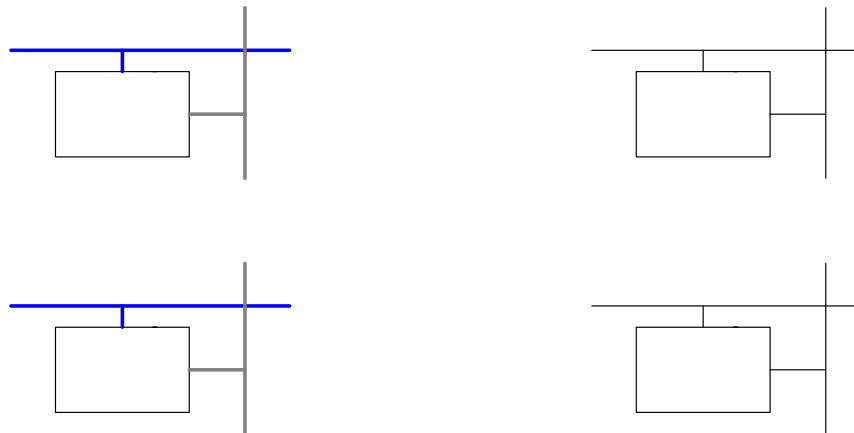
Memory Arrays



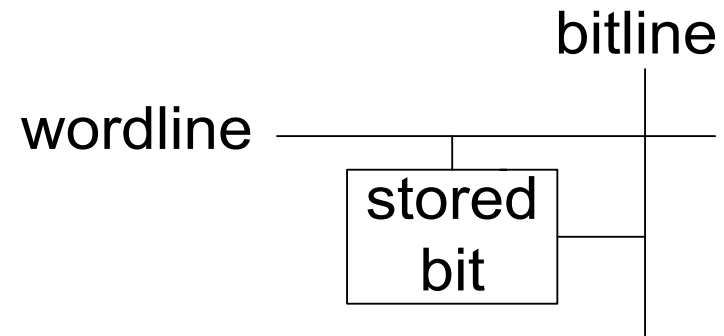
Memory Array Bit Cells



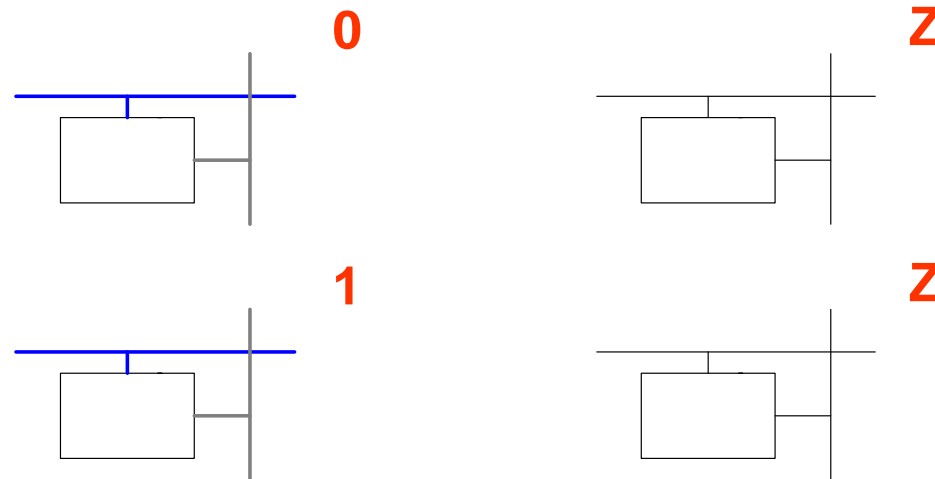
Example:



Memory Array Bit Cells



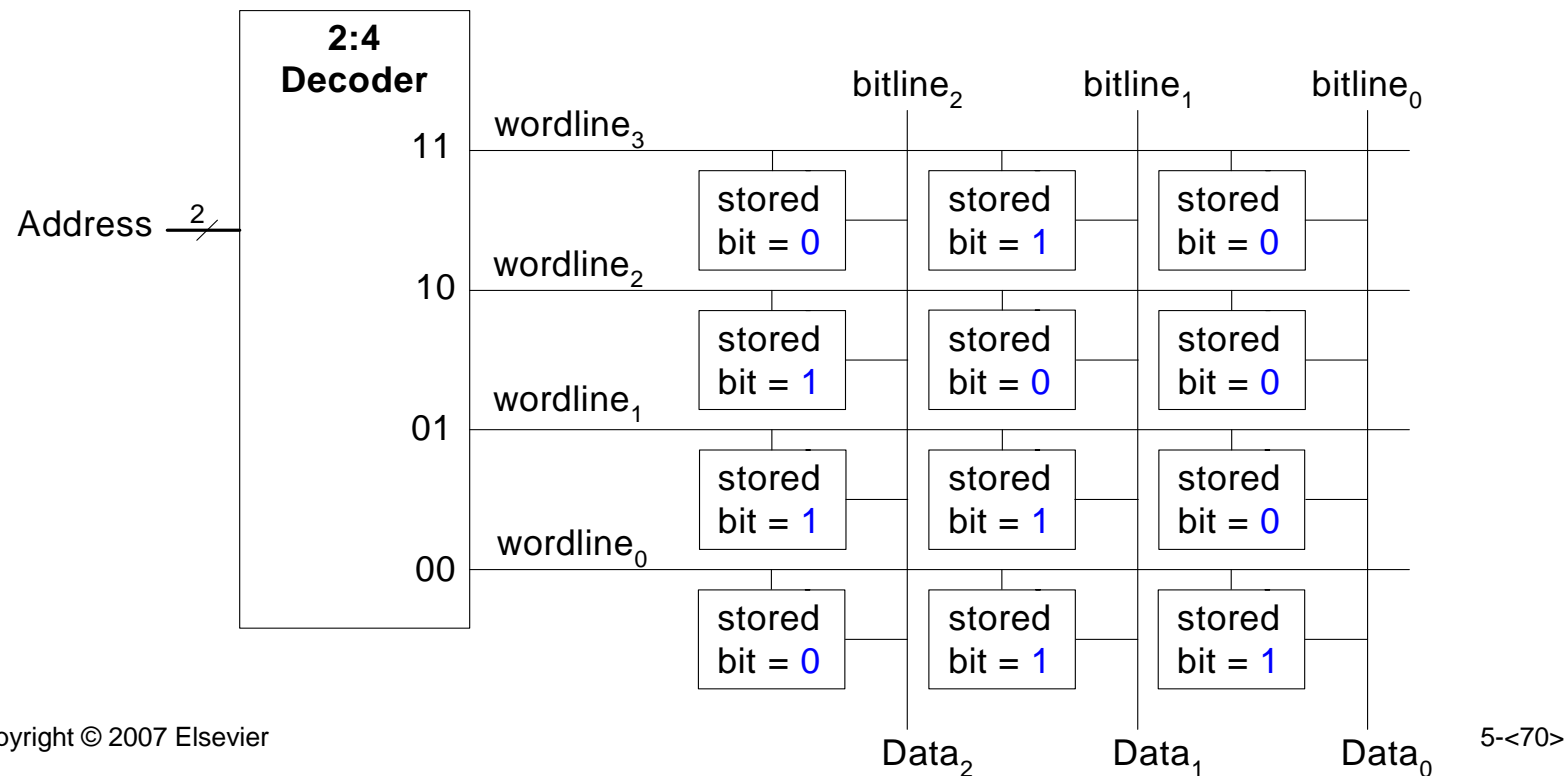
Example:



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**

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

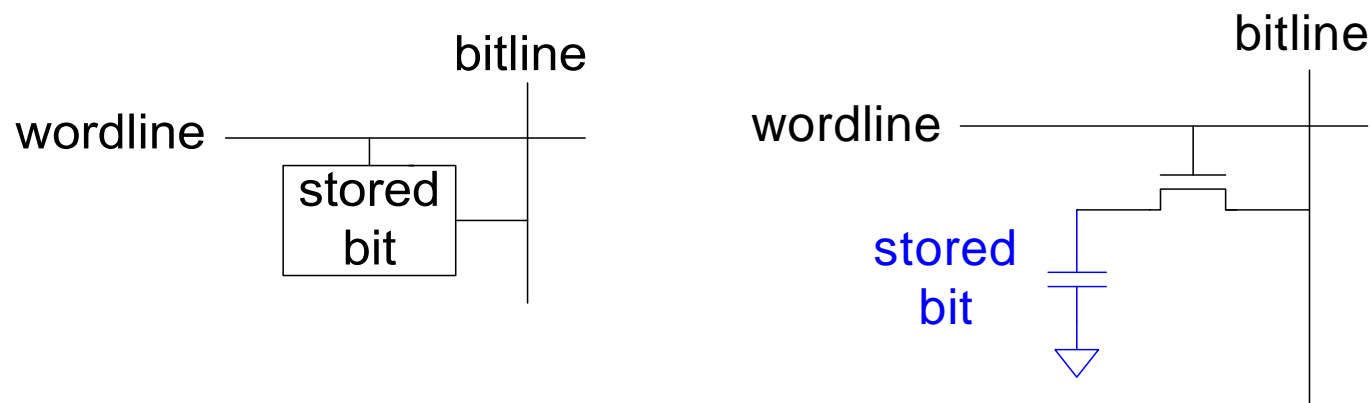
Robert Dennard, 1932 -

- Invented DRAM in 1966 at IBM
- Others were skeptical that the idea would work
- By the mid-1970's DRAM was in virtually all computers

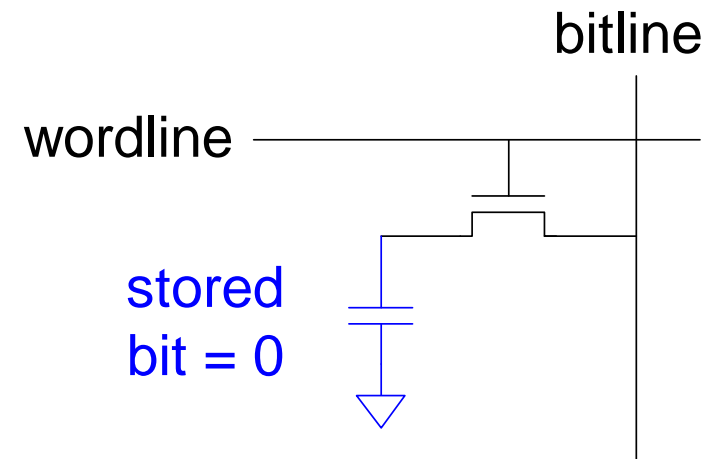
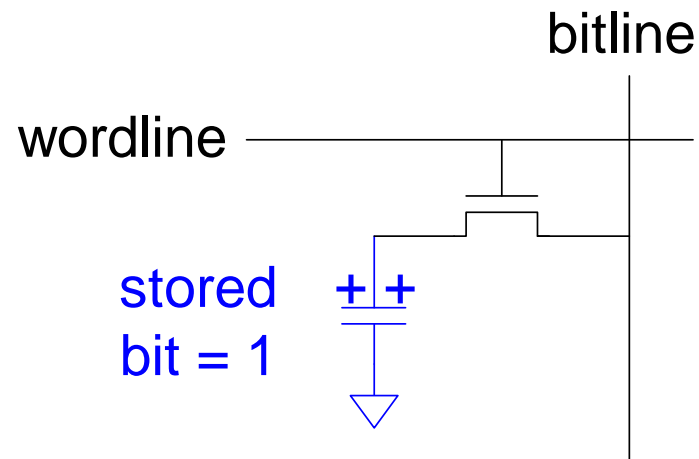


DRAM

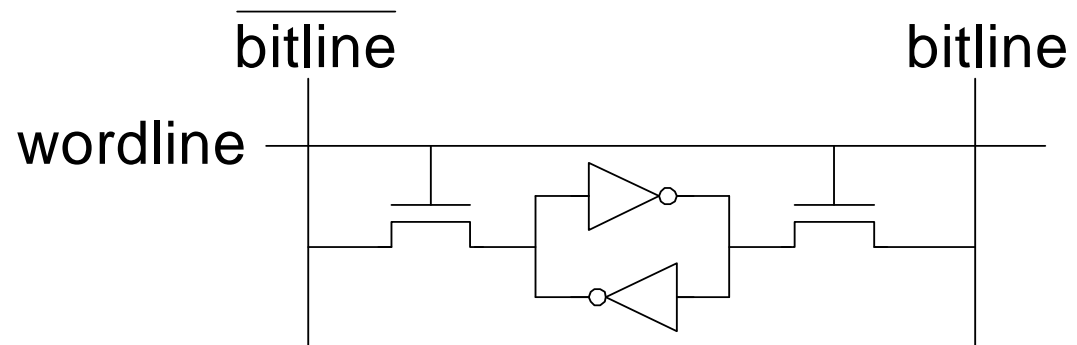
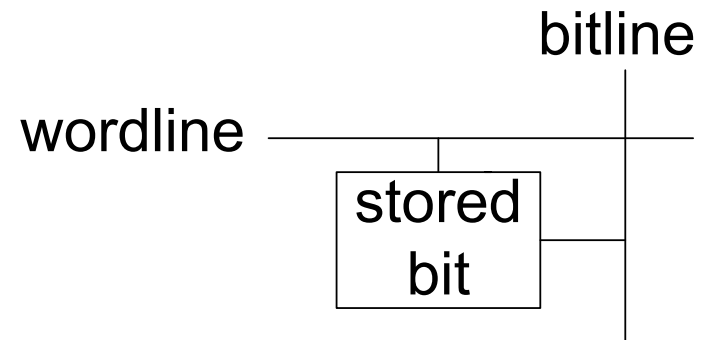
- Data bits stored on a capacitor
- Called *dynamic* because the value needs to be refreshed (rewritten) periodically and after being read:
 - Charge leakage from the capacitor degrades the value
 - Reading destroys the stored value



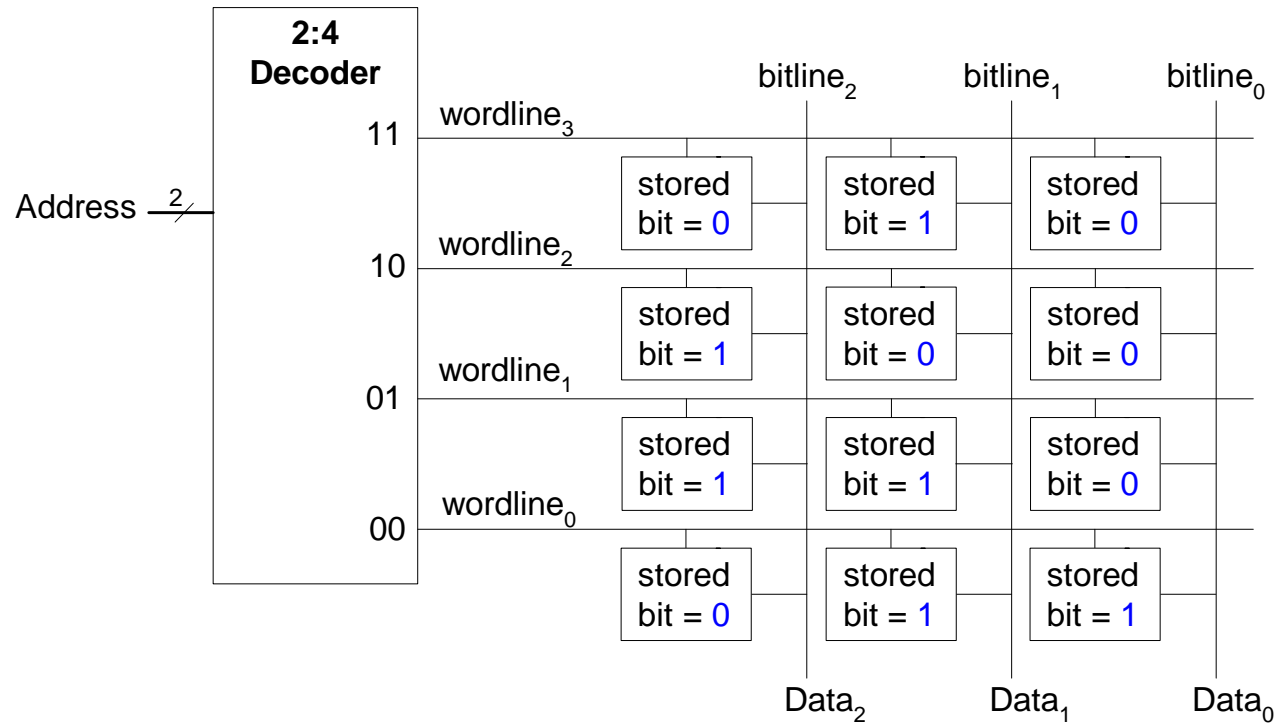
DRAM



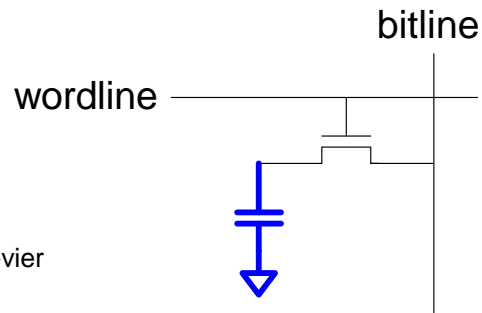
SRAM



Memory Arrays

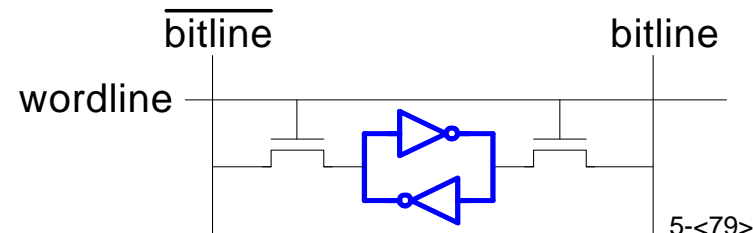


DRAM bit cell:



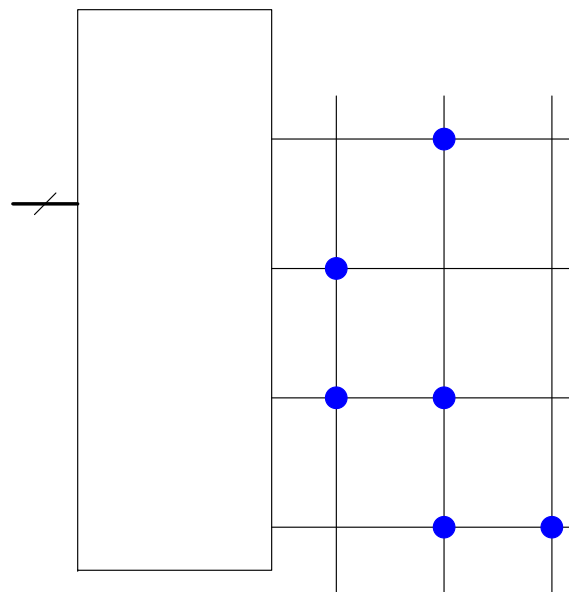
Copyright © 2007 Elsevier

SRAM bit cell:



5-<79>

ROMs: Dot Notation

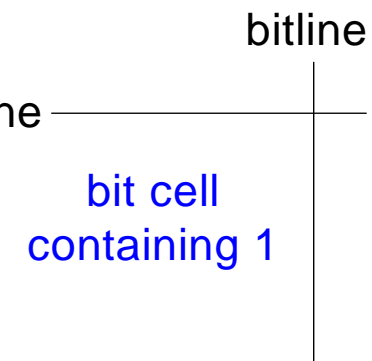
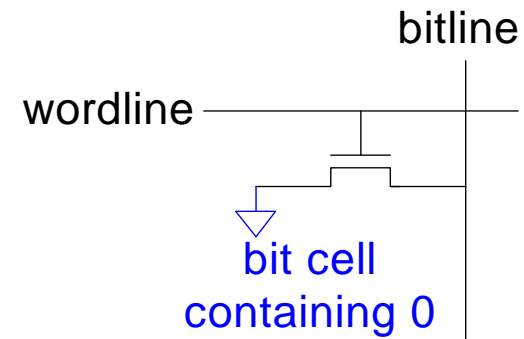


Address ²

2:4
Decoder

11

10

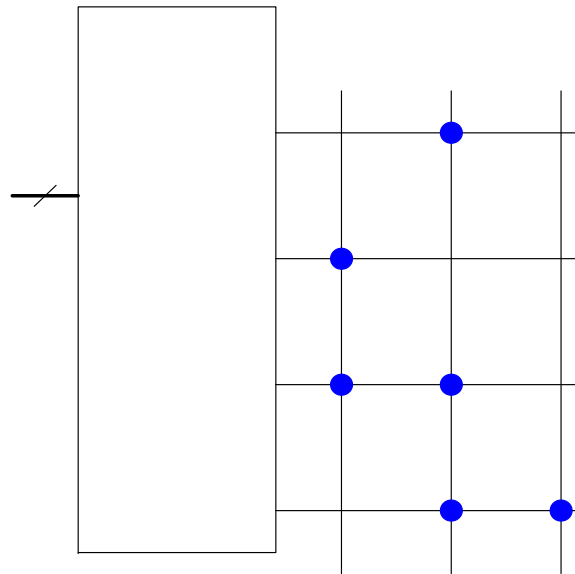


Fujio Masuoka, 1944-

- Developed memories and high speed circuits at Toshiba from 1971-1994.
- Invented Flash memory as an unauthorized project pursued during nights and weekends in the late 1970's.
- The process of erasing the memory reminded him of the flash of a camera
- Toshiba slow to commercialize the idea; Intel was first to market in 1988
- Flash has grown into a \$25 billion per year market.



ROM Storage



Address Data

11	0	1	0
10	1	0	0
01	1	1	0
00	0	1	1

depth

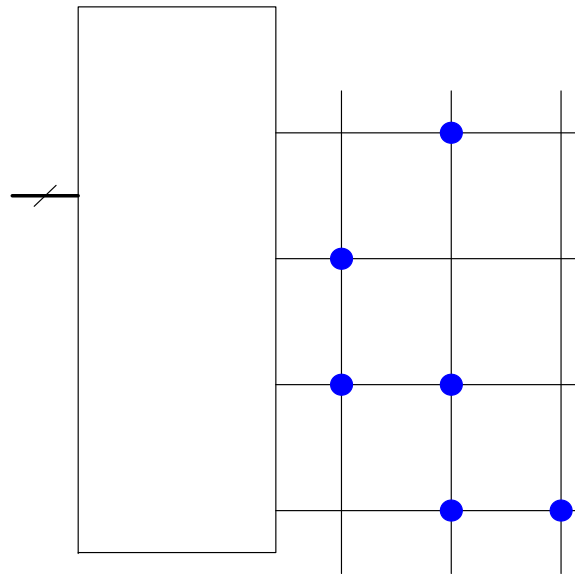
2:4
Decoder width

11

Address ²

10

ROM Logic



$$Data_2 = A_1 \oplus A_0$$

$$Data_1 = \overline{A_1} + A_0$$

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

2:4

Decoder

11

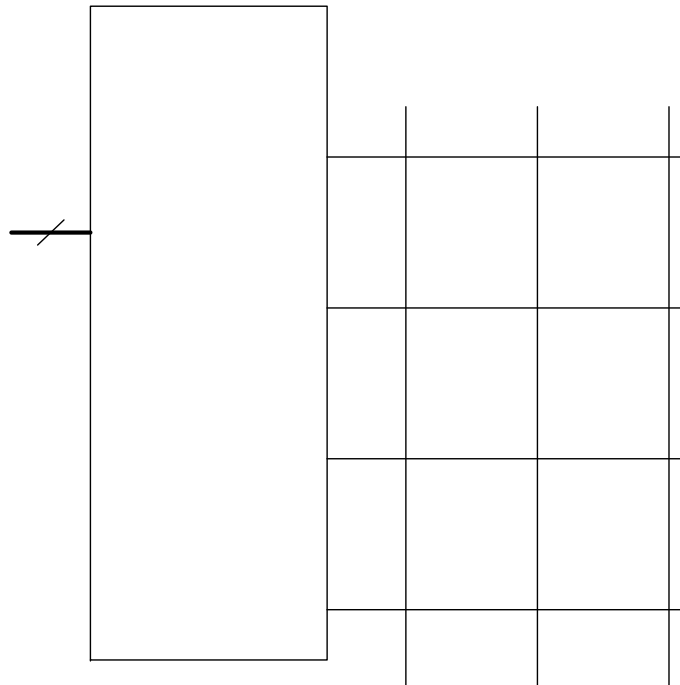
Address ²

10

Example: Logic with ROMs

- Implement the following logic functions using a $2^2 \times 3$ -bit ROM:

- $X = AB$
- $Y = A + B$
- $Z = \overline{AB}$



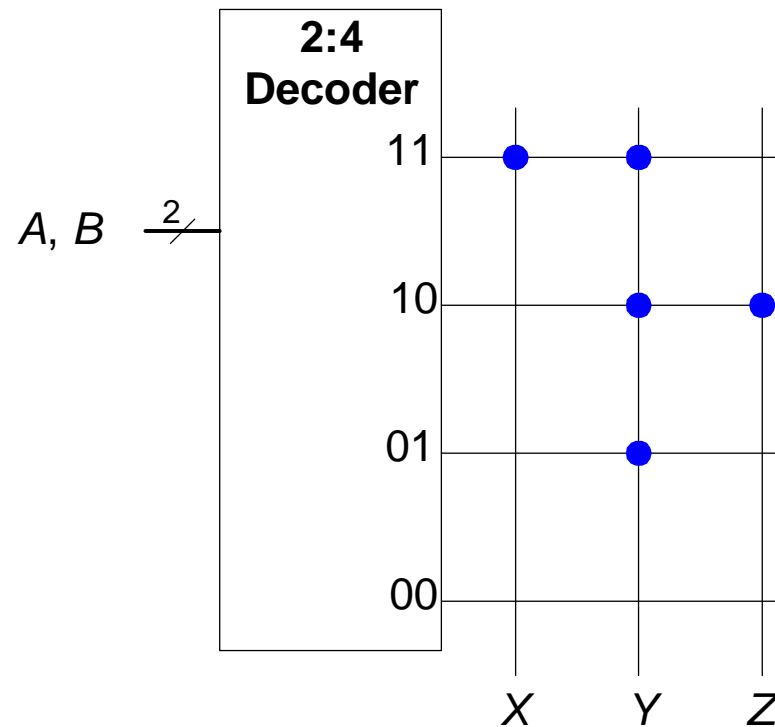
2:4

Decod

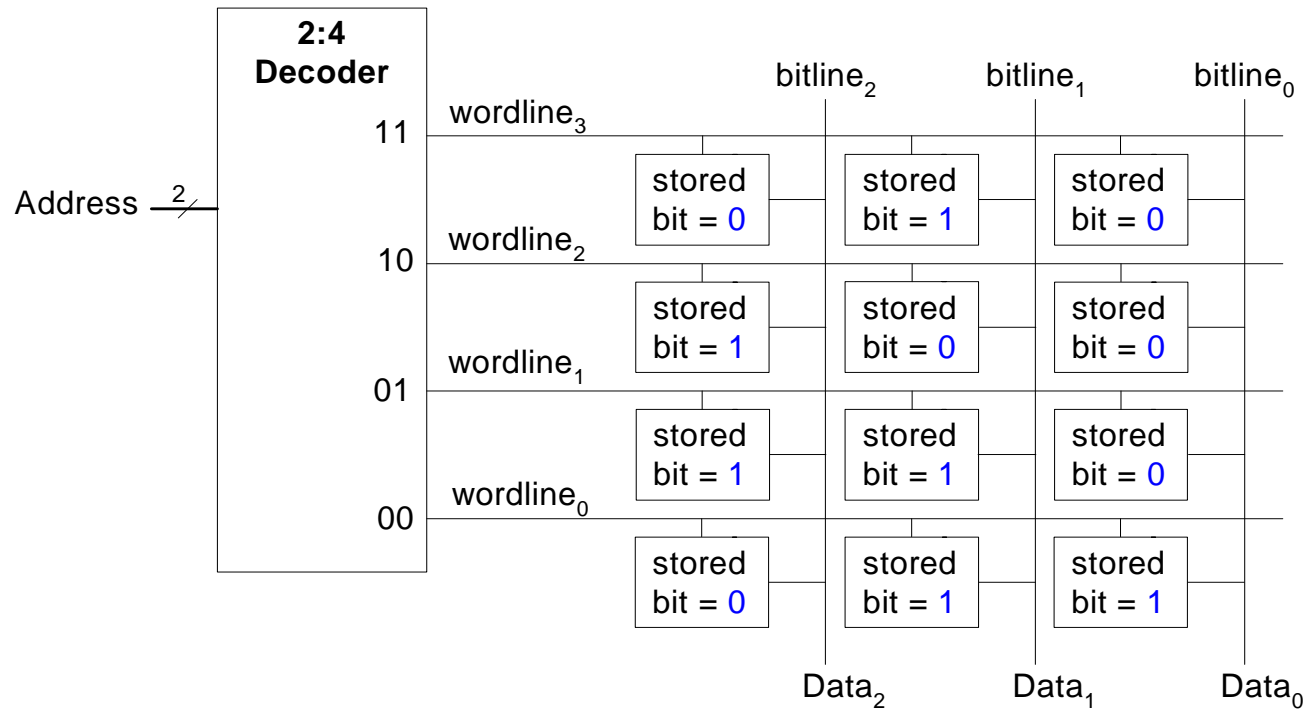
Example: Logic with ROMs

- Implement the following logic functions using a $2^2 \times 3$ -bit ROM:

- $X = AB$
- $Y = A + B$
- $Z = A \overline{B}$



Logic with Any Memory Array



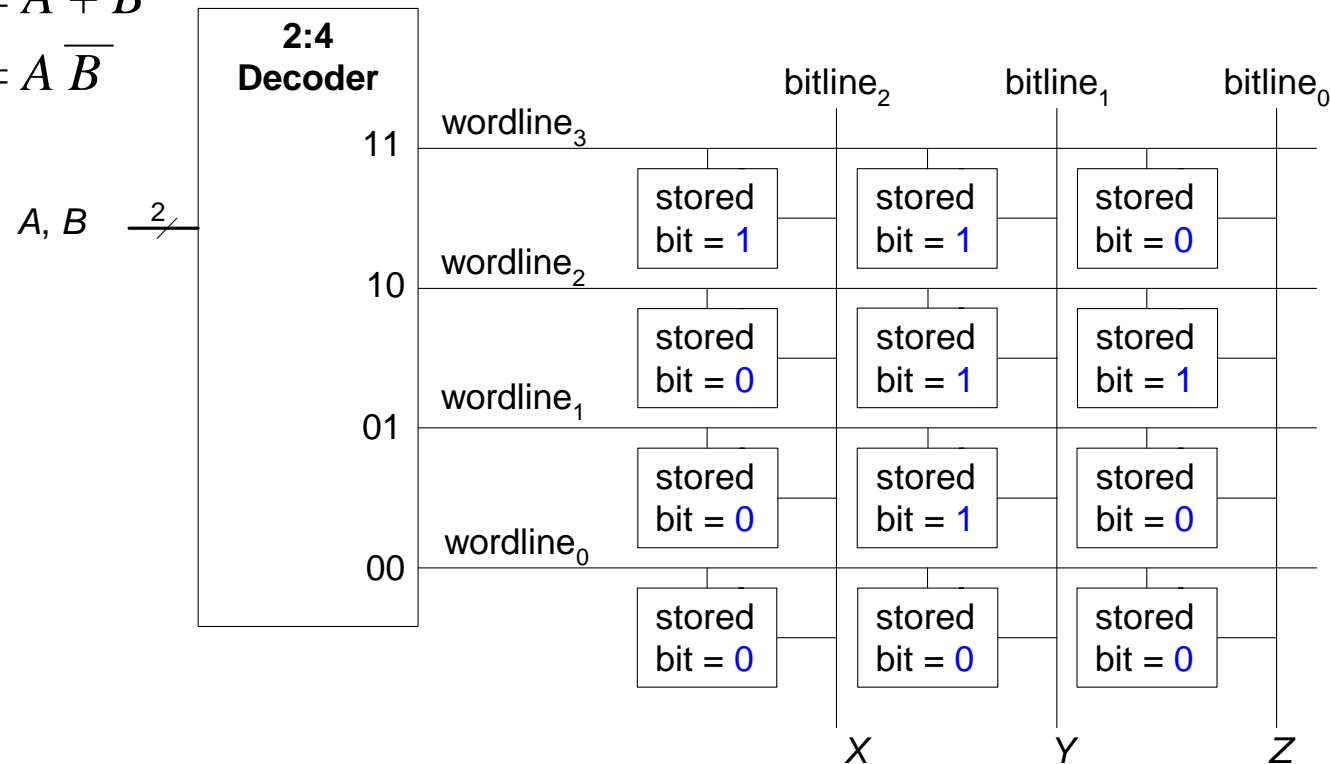
$$Data_2 = A_1 \oplus A_0$$

$$Data_1 = \overline{A_1} + A_0$$

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

Logic with Memory Arrays

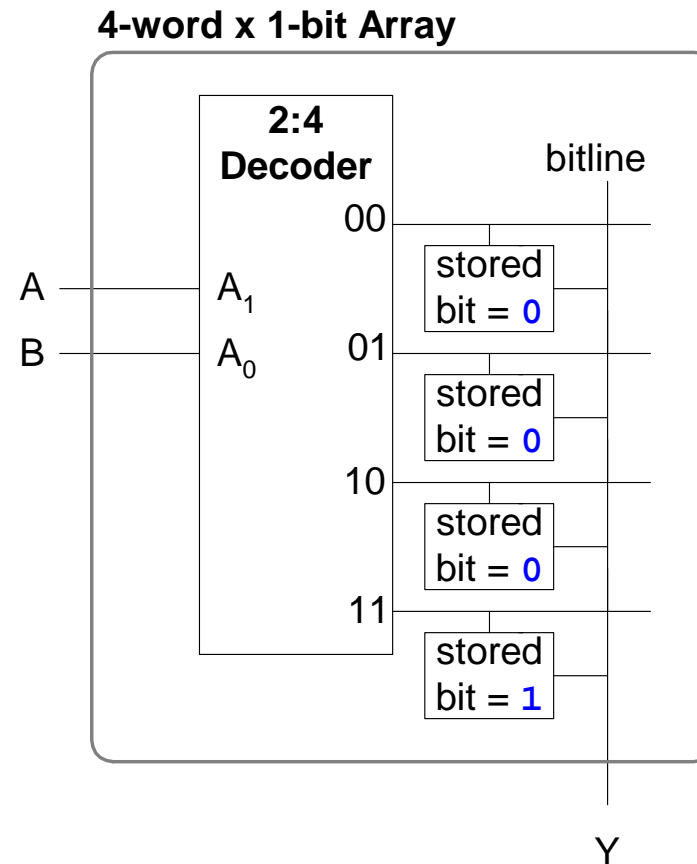
- Implement the following logic functions using a $2^2 \times 3$ -bit memory array:
 - $X = AB$
 - $Y = A + B$
 - $Z = A \overline{B}$



Logic with Memory Arrays

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

Truth Table		
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



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*

