COE211: Digital Logic Design

Memory Devices

Outline

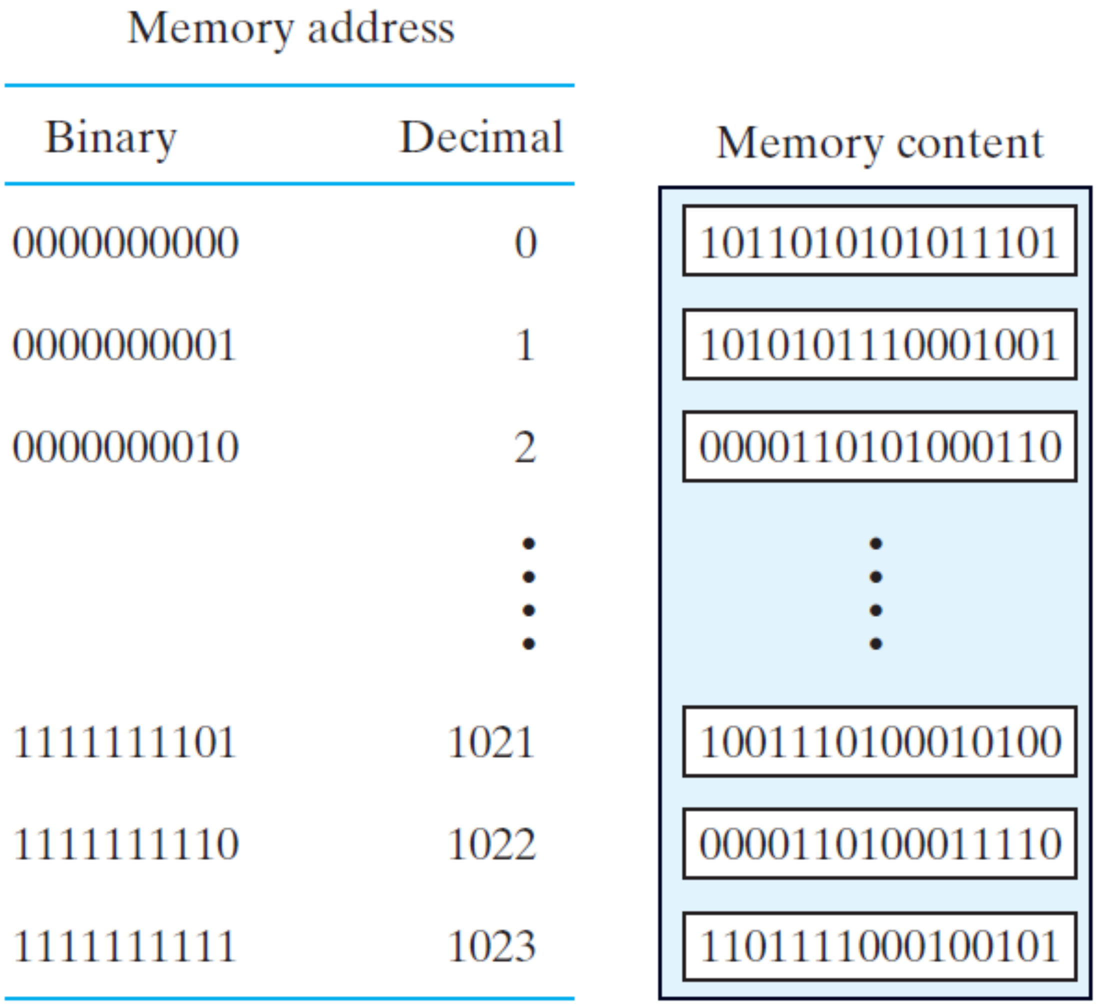
- Random Access Memory (RAM)
  - Read and write operations
  - RAM construction
  - RAM addressing
  - Types of RAM
- Read Only Memory (ROM)
  - Types of ROM
  - Implementing Functions using PROM

Memories

- A memory unit is a device to which binary information is transferred for storage and from which information is retrieved when needed for processing.
- Different memory types:
  - Technology: magnetic, voltage, charge, fuse
  - Volatility: volatile, non-volatile
  - Access sequence: random, sequential
  - Speed: access time and throughput
- Memory access types:
  - Random Access Memory (RAM)
  - Read Only memory (ROM)
- Is memory combinational or sequential logic?
  - RAM: similar to register, based on FFs, but have no clock
  - ROM: combinational (programmable logic)

Memory Layout

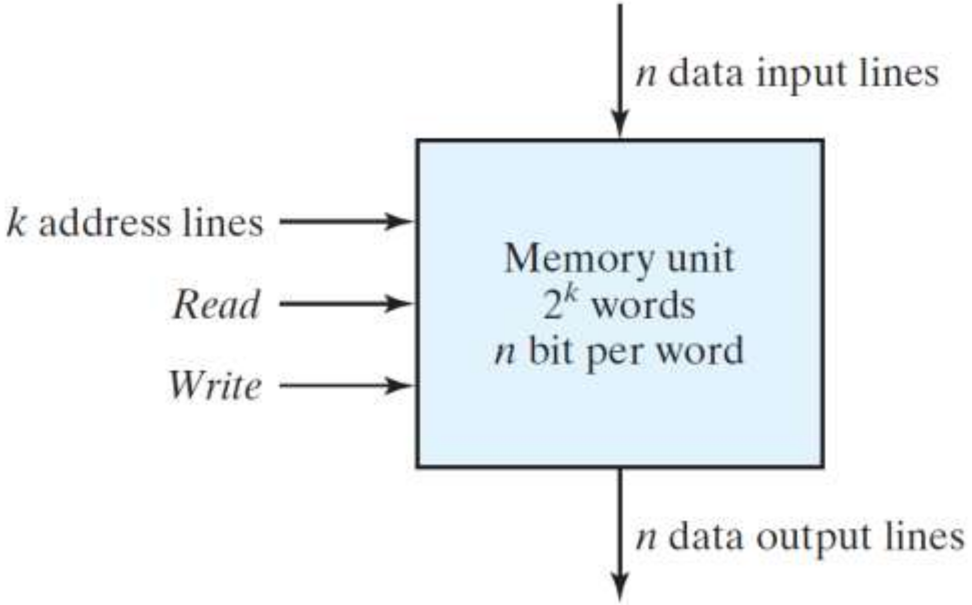
- Example:
  - 1024 16-bit word storage
  - 10 bits identify address
  - 16 bits stored at each location
- Why not make memory bit-wise addressable?
  - How many bits would it have?
  - How many address lines?
  - How big a decoder?
- Remember:
  - k (Kilo) = 2^{10}
  - M (Mega) = 2^{20}
  - G (Giga) = 2^{30}



Memory address		
Binary	Decimal	Memory content
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

Random-access Memory (RAM)

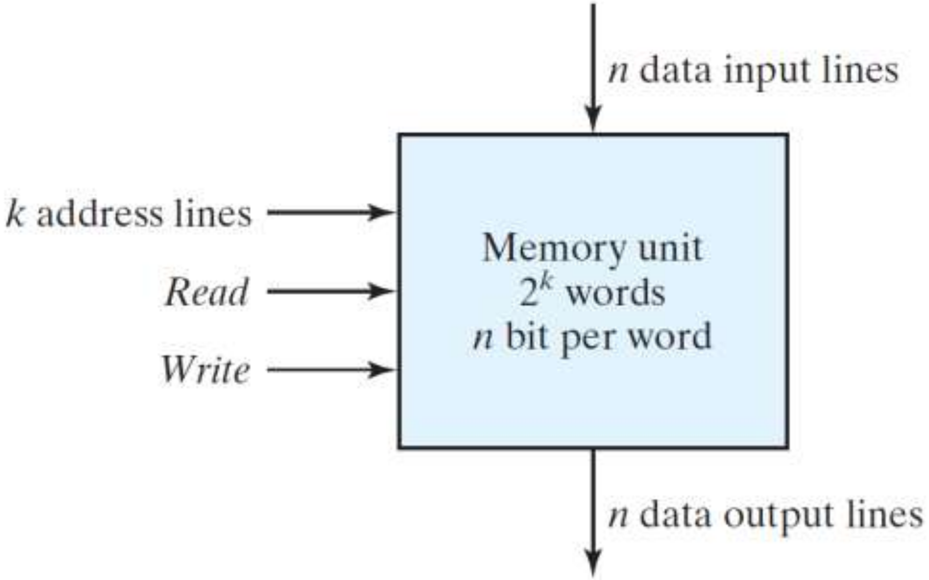
- Memory terminology:
  - Smallest unit is "word" (n bits wide)
  - Identified by "address"
- Block diagram:
  - *k* address lines
  - At most 2^k words storage
  - n data lines
    (input and output)
- What is the total memory size in bits?
  - Size = $2^k \times n$ bits
- Example:
  - 1024-bit memory with 16-bit words
  - How many address lines?



RAM Read and Write Operations

- Control inputs:
  - Address
  - Read
  - Write
- Write operation
  1. Apply address to address lines
  2. Apply data to data input lines
  3. Activate write input
- Read operation
  1. Apply address to address line
  2. Activate read input
  3. Read value from data output lines

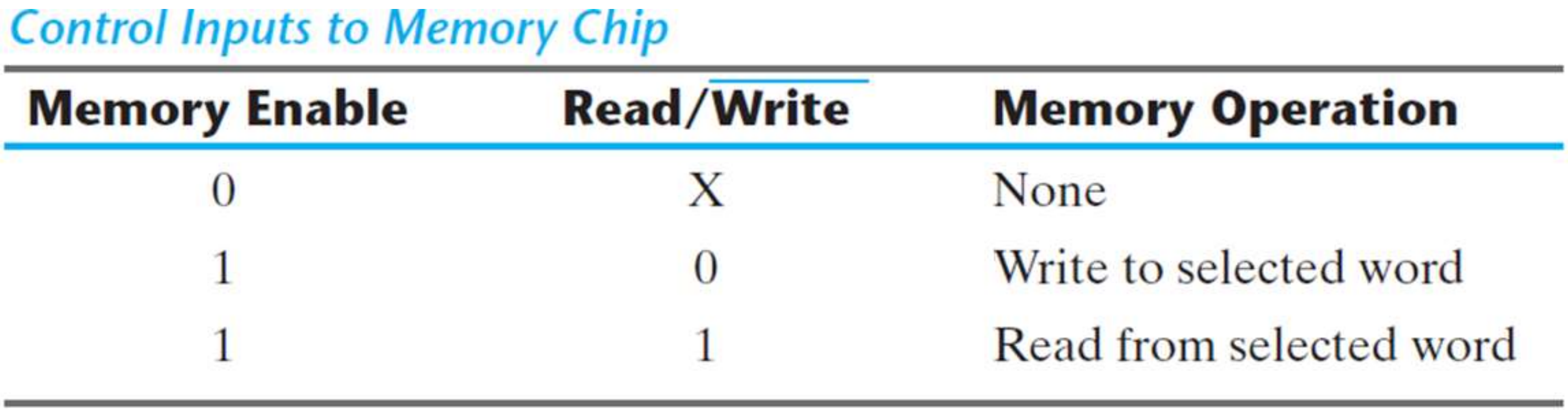
Data to Write



Data to Read

Read and Write Operations (Cont'd)

- Control inputs on commercial memories
  - Memory enable
  - Read and write
- What timing constraints need to be considered?
  - *Access time:* time to select a word and read it
  - *Cycle time:* time to complete a write operation
  - Also: setup and hold time on address lines

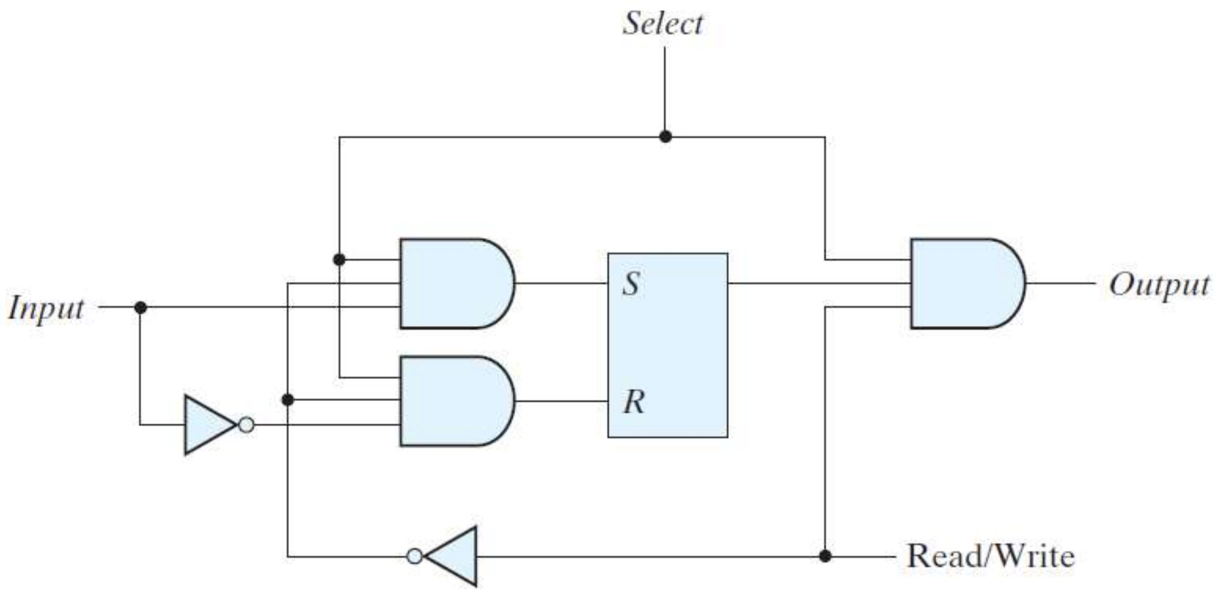


Control Inputs to Memory Chip

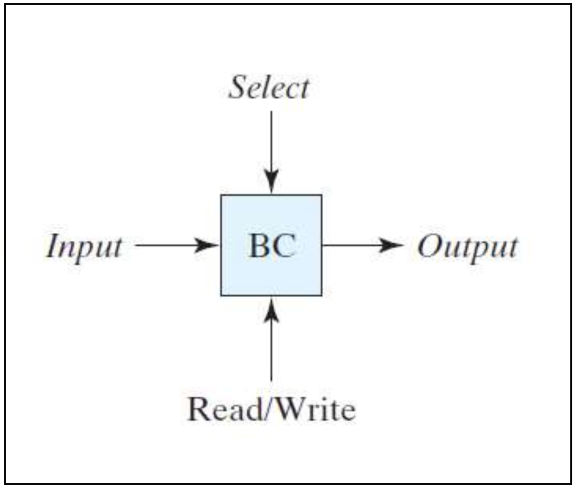
Memory Enable	Read/$\overline{\text{Write}}$	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Memory Cell Design

- Binary memory cell implemented with SR latch: (Note: memory is not clocked)



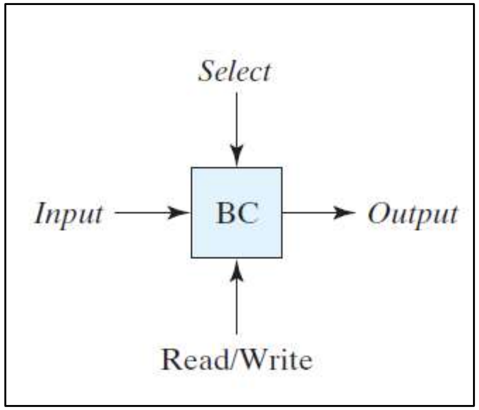
(a) Logic diagram

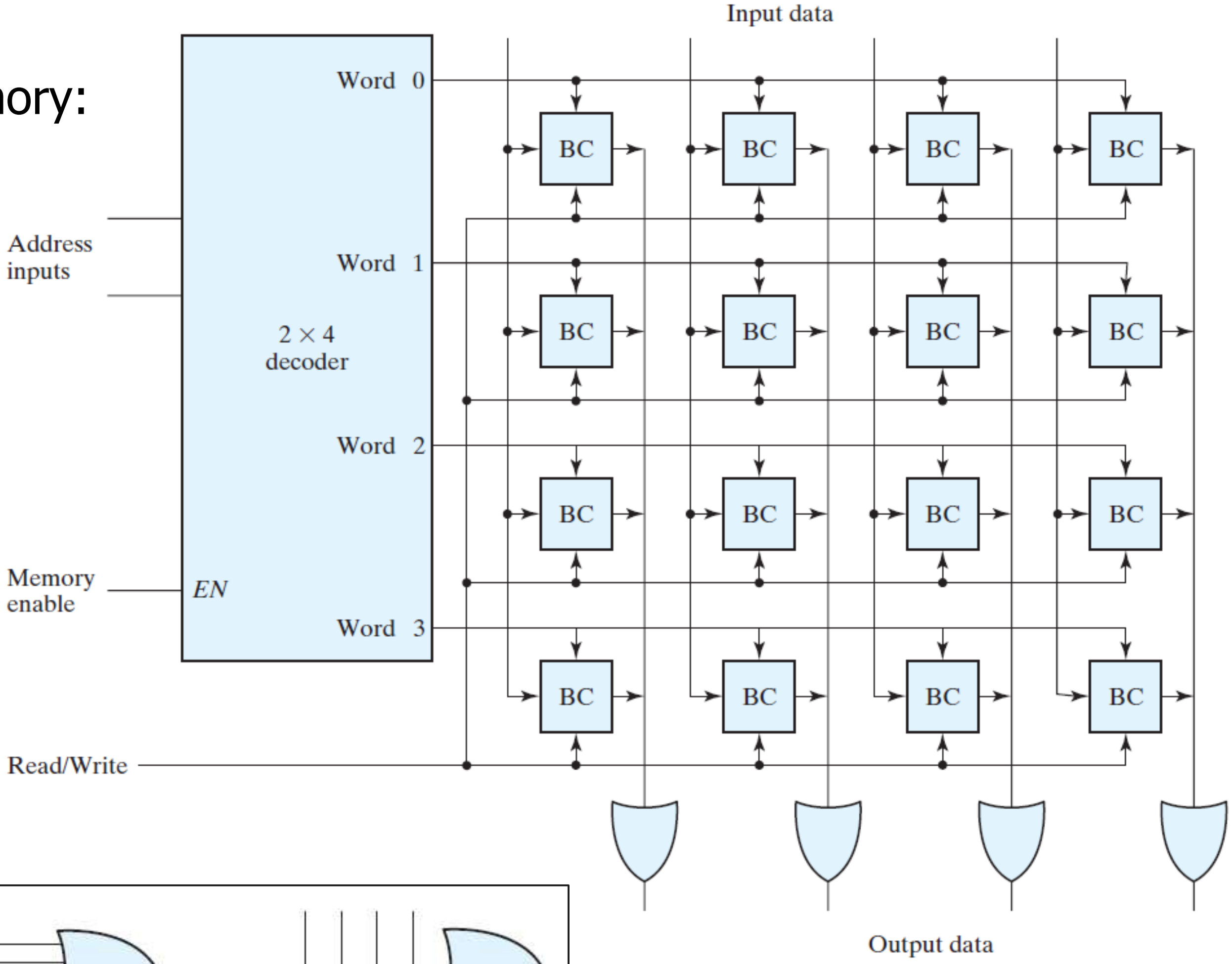


(b) Block diagram

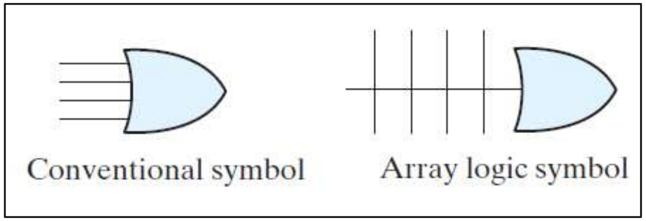
Internal RAM Construction

- Internal layout of memory:
  - One word per "row"
  - Each "binary cell" stores one bit
- Memory cell:





- Decoder:
  - Selects word

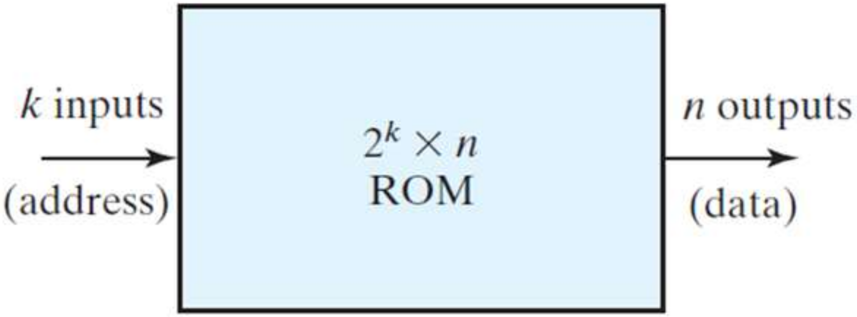


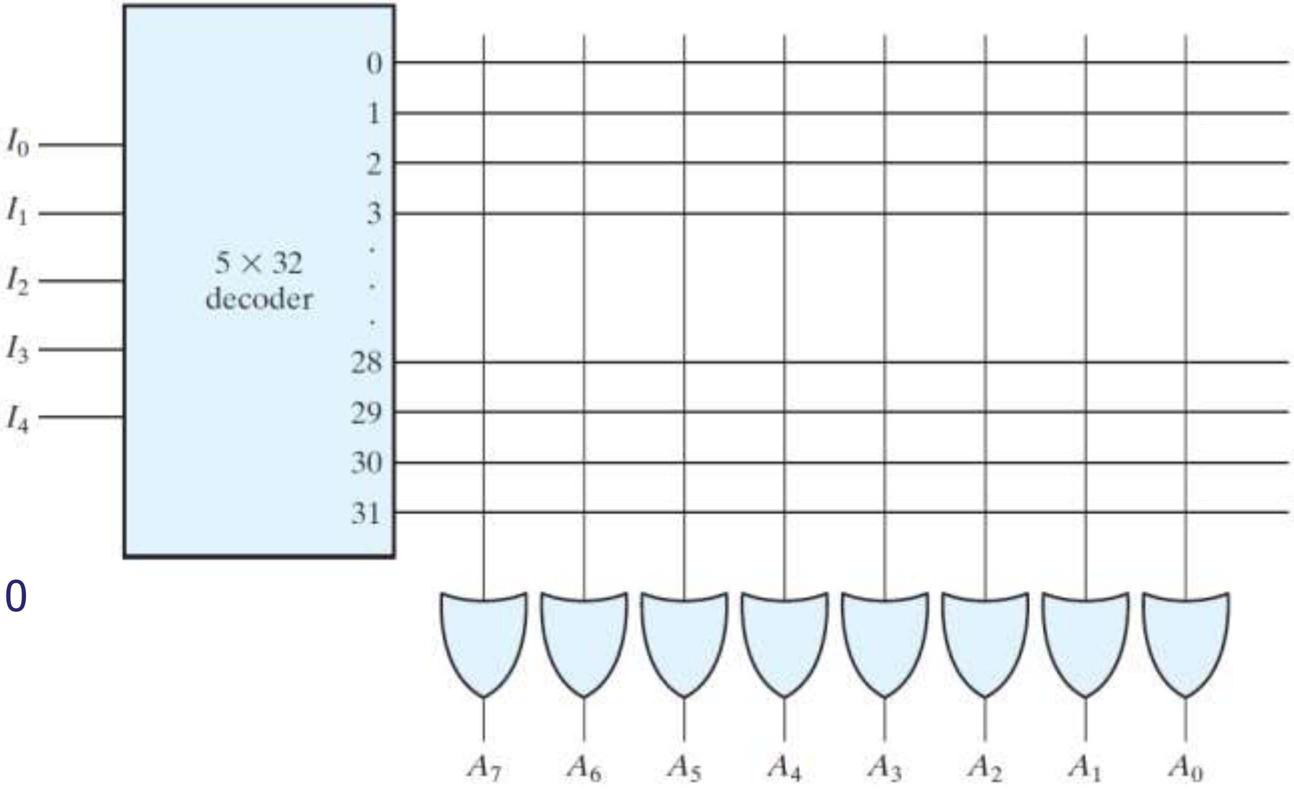
Types of RAM

- Static Random Access Memory (SRAM)
  - Operates like a collection of latches
  - Once a value is written, it is guaranteed to remain in the memory as long as power is applied
  - Generally expensive
  - Used inside processors
- Dynamic Random Access Memory (DRAM)
  - Generally, simpler internal design than SRAM
  - Requires data to be rewritten (refreshed), otherwise data is lost
  - Often hold larger amount of data than SRAM
  - Longer access times than SRAM
  - Used as main memory in computer systems

Read-Only Memory (ROM)

- Storage for memory that does not need to change
- Block diagram:
  - Address
  - Outputs
  - No data inputs



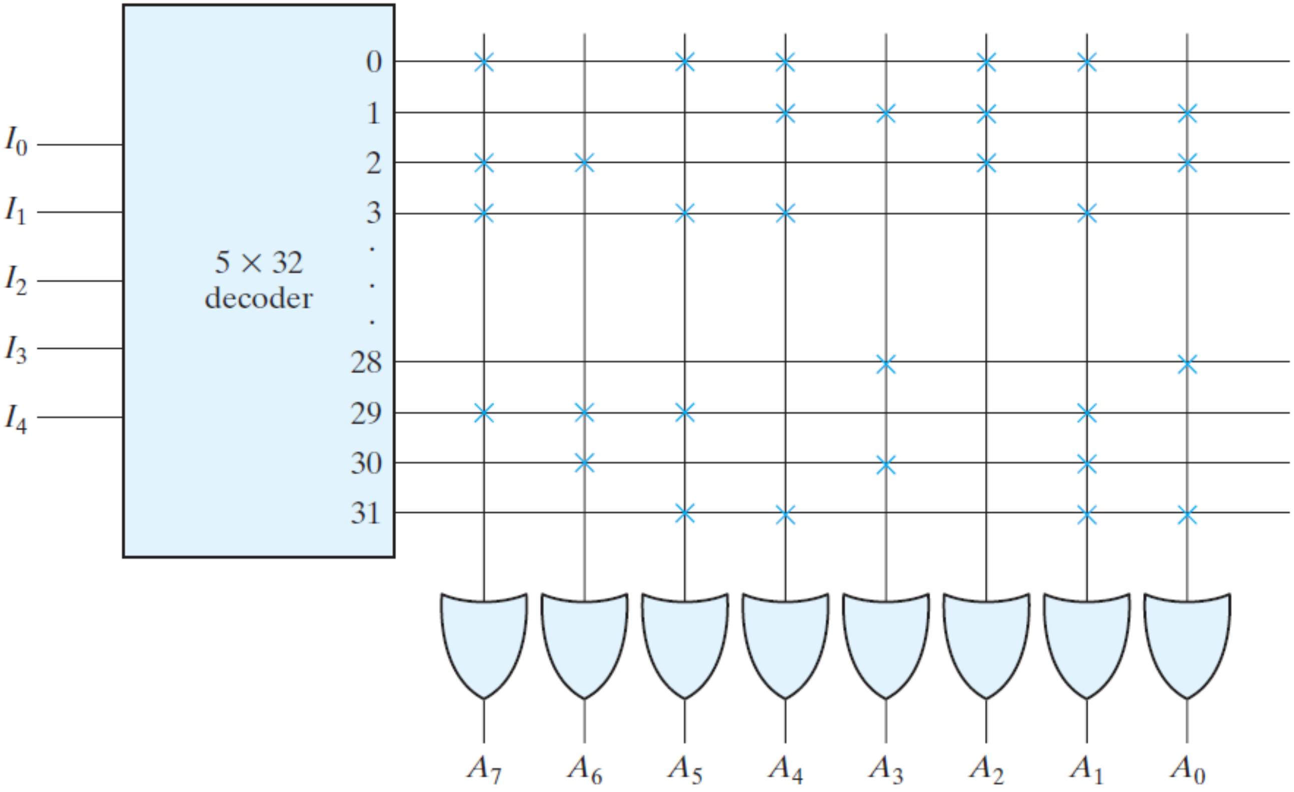


- 32 × 8 ROM with 256 intersections:
  - No connection = 0
  - Connection = 1

ROM Example

- Example of "programmed" ROM:
  - "x" indicates connection
- Programming of ROM
  - Connections implemented as "fuses" (logic 1)
  - Blown fuse (open) indicates 0
  - Higher voltage can blow fuse

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0



Types of ROM

- Mask Programmed ROM
  - During fabrication of ROM
  - Only economical in large quantities
- Programmable ROM (PROM)
  - ROM with all fuses intact
  - High-voltage pulse on special pin can irreversibly blow fuses
  - *Note:* hardware procedure despite "programmable"
- Erasable PROM (EPROM)
  - Connections can be restored
  - Exposure to UV light resets EPROM
- Electrically-erasable PROM (EEPROM)
  - Electric signal can reset connections (no UV required)

Implementing Functions using PROM

- The shown PROM implements the following two Boolean functions:

- *F1(x,y,z)*

  $$= \sum(0,3,5,6)$$

  $$= x'y'z' + x'yz$$

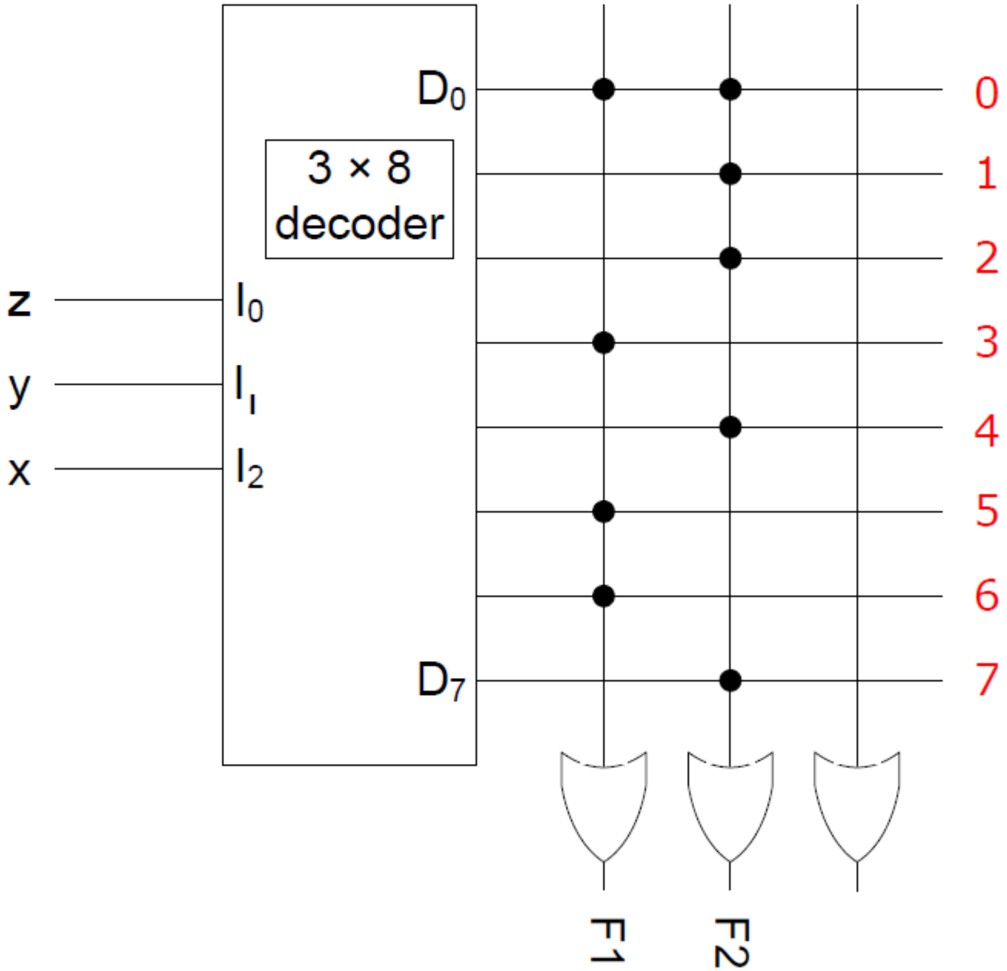
  $$+xy'z + xyz'$$

- *F2(x,y,z)*

  $$= \sum(0,1,2,4,7)$$

  $$= x'y'z' + x'y'z$$

  $$+x'yz' + xy'z' + xyz$$


z
y
x
I0
I1
I2
3 × 8 decoder
D0
D7
0
1
2
3
4
5
6
7
F1
F2