Programmable Logic Devices

CS207 Lecture 14

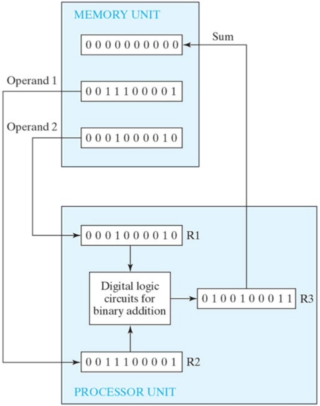
James YU

May 20, 2020

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Memory

- A memory unit is a device to which binary information is transferred for storage and from which information is retrieved when needed.

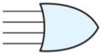


Memory

- A memory unit is a collection of cells capable of storing a large quantities of binary .information
  - Random access memory (RAM)
    - Performs both read and write operations.
  - Read-only memory (ROM)
    - Only read operation, information cannot be altered by writing.
    - Is a programmable logic device (PLD).
    - Information storage in some fashion and embedded within hardware, called *programming* the device.

Memory

- PLD has many types, ROM is one.
  - Others are Programmable Array Logic (PAL), Programmable Logic Array (PLA), and the field-programmable gate array (FPGA).
  - An integrated circuit, internal logic gates connected through electronic paths similar to fuses.
  - May have hundreds to millions of gates, thus requires a special gate symbology:



(a) Conventional symbol

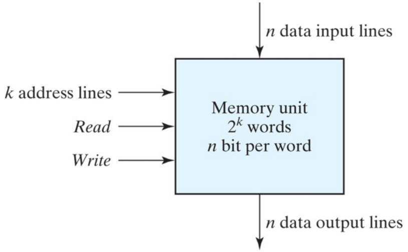
(b) Array logic symbol

Random-access memory

- A memory unit is a collection of storage cells with associated circuits needed.
  - Information can be selectively retrieved from any of its internal locations.
  - Information retrieval time at any random location is the same, therefore *Random-Access* memory.
- Binary information stored are grouped in words.
  - Words are the unit of moving in and out of storage.
  - (Recall) 8 bits = 1 byte.
  - Most computer memory has words multiple of 8 bits/1 byte in length.

Random-access memory

- Communication achieved through:
  - Data input and output lines.
  - Address selection lines.
  - Control lines.
- To have n bits in a word,
  - n lines for input/output
- To have m words,
  - k lines for address, such that $m \leq 2^k$.



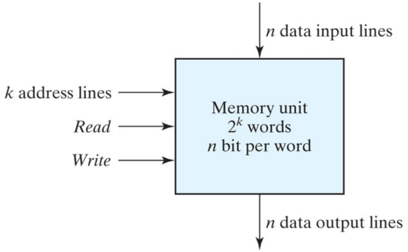
Random-access memory

- Each word in the memory is assigned a unique address from 0 to 2^k-1.
- An internal decoder: k lines to 2^k addresses.
- On the right is 1K words of 16 bits.

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

Random-access memory

- Two operations of RAM: read and write.
- Write
  - Apply binary address to address lines;
  - Apply data bits to data input lines;
  - Activate the write input
- Read
  - Apply binary address to address lines;
  - Activate the read input;



Random-access memory

- Commercial memory components:
    - One input selects the unit.
    - The other determines the operation.

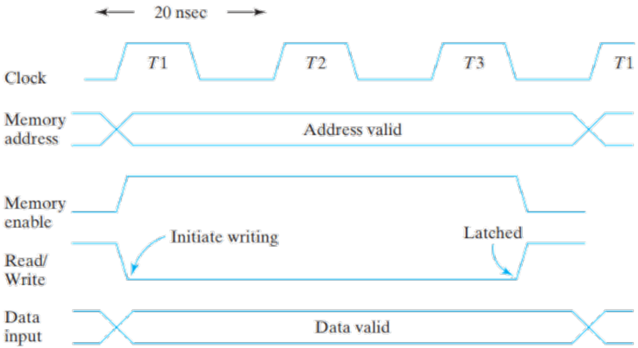
Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Random-access memory

- Memory unit operation is controlled by an external device, e.g., CPU, synchronized by its own clock.
  - Memory does not employ an internal clock.
  - *Access time*: time required to select a word and read it.
  - *Cycle time*: complete a write operation.
  - CPU provide memory control signals to synchronize.
    - Access/Cycle time within a time equal to a fixed number of CPU clock cycles.

Random-access memory

- CPU: 50MHz – one clock cycle = 20ns
- Memory: access time = cycle time = 50ns
  - These times not always equal.

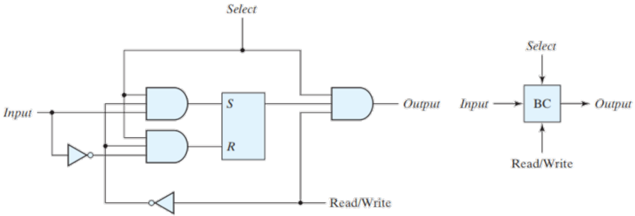


Random-access memory

- Integrated circuit RAM units available in two operating modes:
- Static RAM: essentially latches, available as long as powered.
    - Easier to use, has shorter read/write cycles.
- Dynamic RAM: information stored in the form of electric charges on capacitors, discharge over time.
    - Capacitors periodically recharged by refreshing the memory.
    - Reduced power consumption and larger storage capacity
- Memory units that lose stored information when power is turned off are *volatile*.
- Non-volatile memories retained information with magnetic components.
    - Data represented by direction of magnetization.

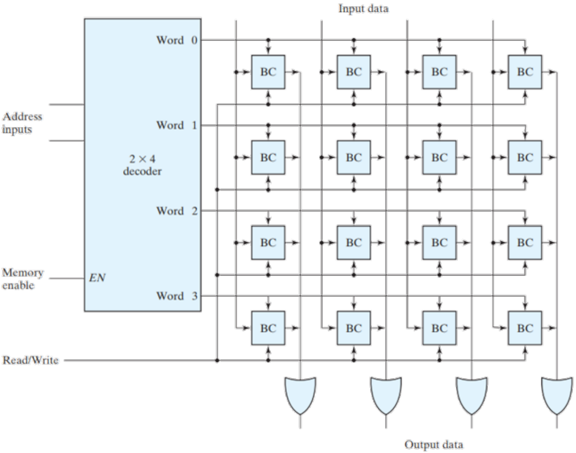
Memory decoding

- To require storage components in a memory unit, a decoding circuit is required:
  - Select memory word specified by input address
- m words with n bits each consist of $m \times n$ binary storage cells as the basic building blocks.
- Modeled by an SR latch with external gates to form a D latch.
  - Actual cells are circuits with 4 to 6 transistors.
  - Convenient to model it with logic symbols



Memory decoding

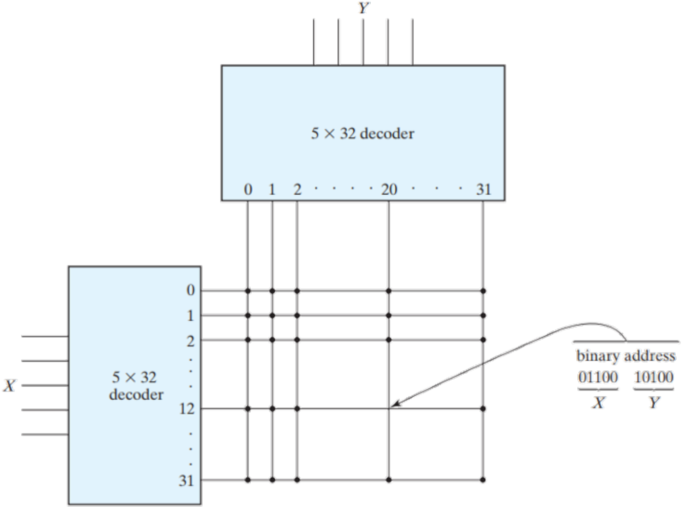
- A binary storage cell must be very small.
  - Pack as many cells as possible in the small IC chip.



Memory decoding

- Commercial RAMs have a capacity of thousands of words, each 1 to 64 bits
- $k - 2^k$ decoder requires 2^k AND gates with k inputs.
  - Can be reduced by employing two decoders: *coincident decoding*.
  - One decoder performs row selection, the other column selection

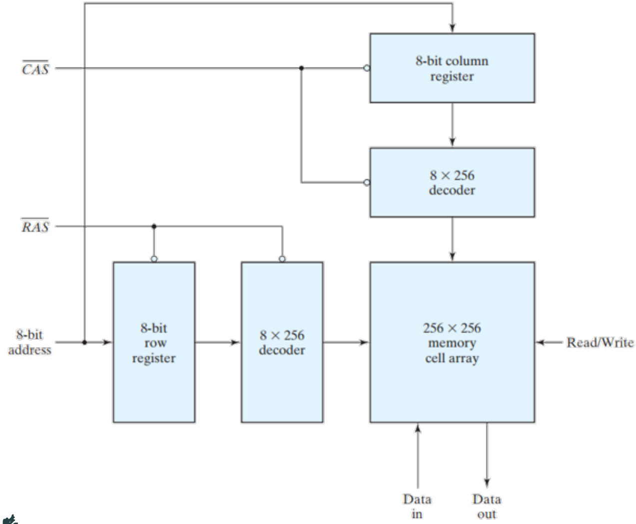
Memory decoding



Memory decoding

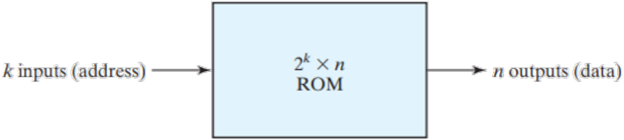
- SRAM: typically 6 transistors
- DRAM: a MOS transistor + a capacitor
  - Four times the density of SRAM.
  - Preferred technology for large memories.
- Address decoding for DRAM is in two-dimensional array, larger memories have multiple arrays.
- Address multiplexing is used to reduce the number of pins in the IC package.
  - The same set of pins used for row and column selection.

Memory decoding



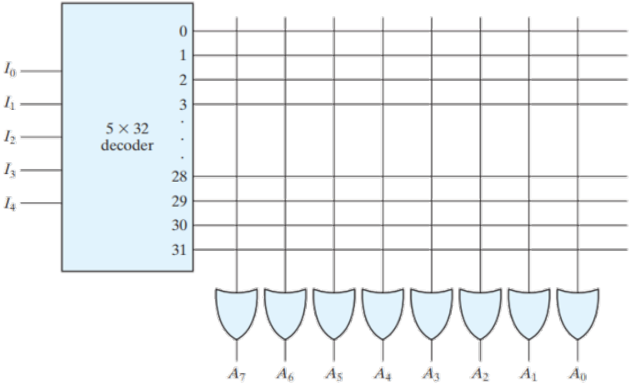
Read-only memory

- ROM is a memory device in which permanent binary information is stored.
- Once the pattern is established, stays within the unit even when power is turned off.



Read-only memory

- Consider a 32×8 ROM:
- Five input lines for address.
- 32 input connections and 8 OR gates = 256 internal connections.



Read-only memory

- These intersections are programmable:
  - Logically equivalent to a switch that can be closed (connected) or open (disconnected).
  - Also called *crosspoints*.
- One of the simplest technology employs a fuse.
  - Normally connects the two points.
  - Opened or “blown” by high-voltage pulse into the fuse.

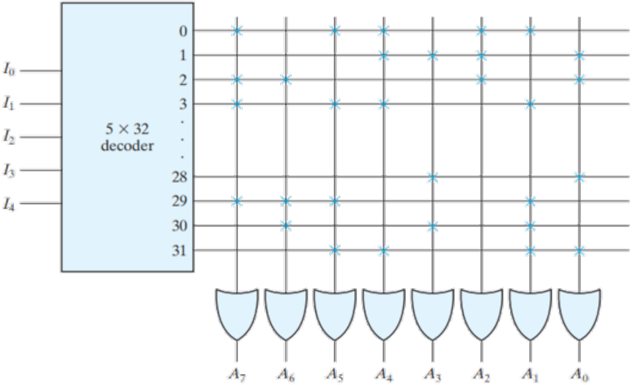
Read-only memory

- Internal binary storage of a ROM specified by a truth table.
  - Each address stores a word of 8 bits.

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
		⋮						⋮				
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

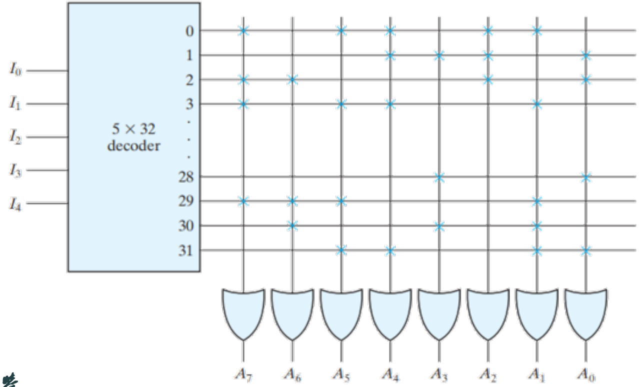
Read-only memory

- Program the ROM: blows fuse links in accordance with the truth table
  - 10110110 in the first row.



Read-only memory

- Combinational circuit
  - (Recall) Decoder: k input = 2^k minterms.
  - By choosing connections for those minterms.
  - For example, eight functions of five binary inputs.



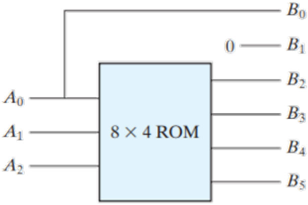
Example

- Design a combinational circuit using a ROM. The circuit accepts a three□bit number and outputs a binary number equal to the square of the input number.
  - $B_0 \equiv A_0$, $B_1 \equiv 0$, no need to generate.

Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

Example

- 3 inputs, 4 outputs: 8×4 ROM.



(a) Block diagram

A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

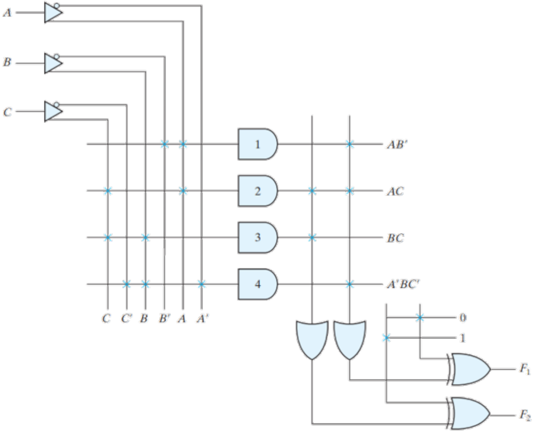
(b) ROM truth table

Read-only memory

- A programmable ROM is a kind of combinational programmable logic device (PLD) with fixed AND array (decoder) and programmable OR array.
- Others are programmable array logic (PAL) with programmable AND array and fixed OR array,
- and programmable logic array (PLA) with programmable AND and OR arrays.
  - Name emerged from different vendors during the development of PLD.

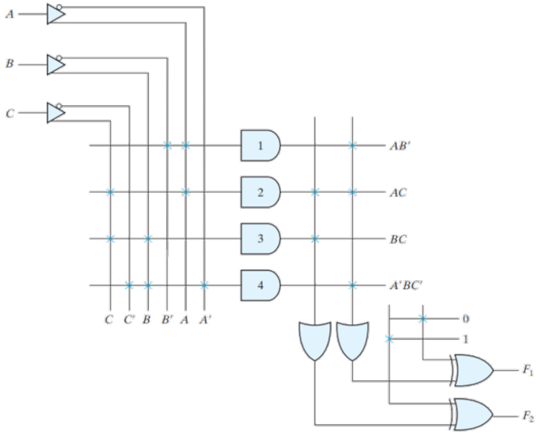
Programmable logic array

- Similar to PROM, but no full decoding of variables, i.e., not all minterms.
- Decoder replaced by an array of AND gates.



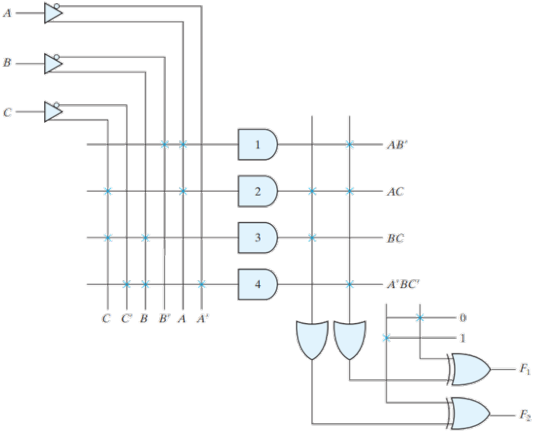
Programmable logic array

- Each input goes through a buffer-inverter combination.
- XOR receive logic `0` or `1`: complement on `1`.



Programmable logic array

- $F_1 = AB' + AC + A'BC'$
- $F_2 = (AC + BC)'$



Programmable logic array

- $F_1 = AB' + AC + A'BC'$
- $F_2 = (AC + BC)'$
- Fuse map can be specified in a tabular form.

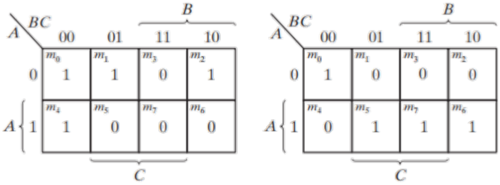
		Inputs			Outputs (T)	Outputs (C)
	Product Term	A	B	C	F_1	F_2
AB'	1	1	0	—	1	—
AC	2	1	—	1	1	1
BC	3	—	1	1	—	1
$A'BC'$	4	0	1	0	1	—

Programmable logic array

- Size of a PLA is specified by the number of inputs, product terms, and outputs.
  - Typically have 16 inputs, 48 product terms, 8 outputs.
- For n inputs, k products, m outputs:
  - n buffer-inverter gates;
  - k AND gates;
  - m OR gates, and m XOR gates;
  - $2n \times k$ connections between inputs and AND array;
  - $k \times m$ connections between AND and OR arrays;
  - m connections associated with XOR gates.

Example

- Implement the following two Boolean functions with a PLA:
  - $F_1(A,B,C) = \sum(0,1,2,4) = (AB + AC + BC)'$.
  - $F_2(A,B,C) = \sum(0,5,6,7) = AB + AC + A'B'C'$.



PLA programming table

	Product term	Inputs			Outputs (C)	Outputs (T)
		A	B	C	F_1	F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$A'B'C'$	4	0	0	0	–	1