

# Module 6 – Memory and Programmable Logic Devices

**CSE-103 Digital Computer Logic**

**Athar Mahboob**

**WWW:** <http://www.atharmahboob.com>

**Email:** [athar@atharmahboob.com](mailto:athar@atharmahboob.com)

# Module Outline

---

- Random Access Memory
- RAM Integrated Circuits
- Programmable Logic Technologies
- ROM
- PLA
- PAL
- CPLD
- FPGA

# Random Access Memory - RAM

---

- Memory is a collection of storage cells together with associated circuits needed to transfer information into and out of these cells.
- Why called Random Access Memory – any cell in the memory can be accessed in the same out of time
- Contrast that with the Hard Disk, Floppy, Tape, CD-ROM which are all sequential access
- Generally information is stored in word size

# Block Diagram of Memory

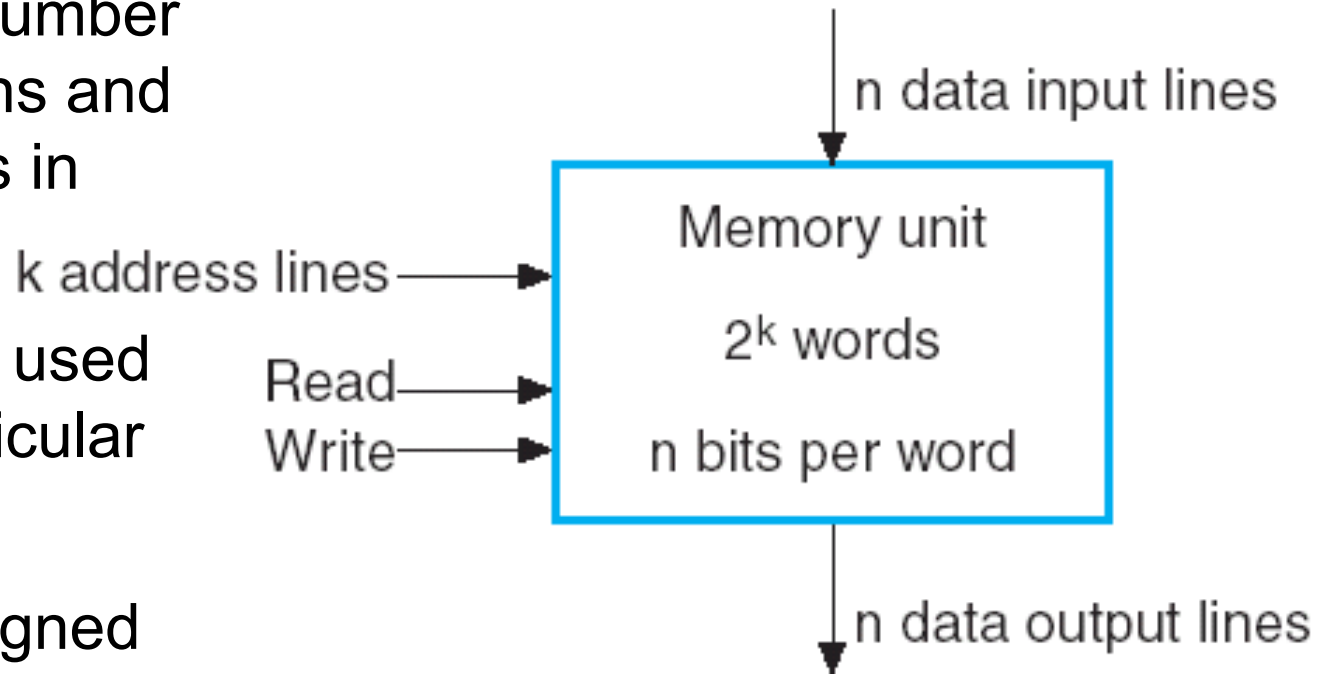
---

The memory unit is specified by the number of words it contains and the number of bits in each word

Address lines are used to select one particular word

Each word is assigned an address

Sizes are measured in k, M, G

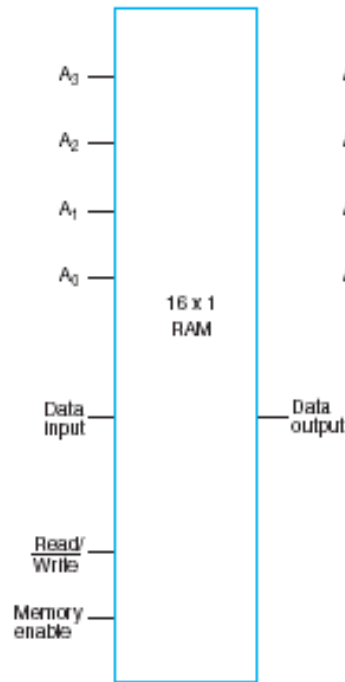


# Contents of a 1024 × 16 Memory

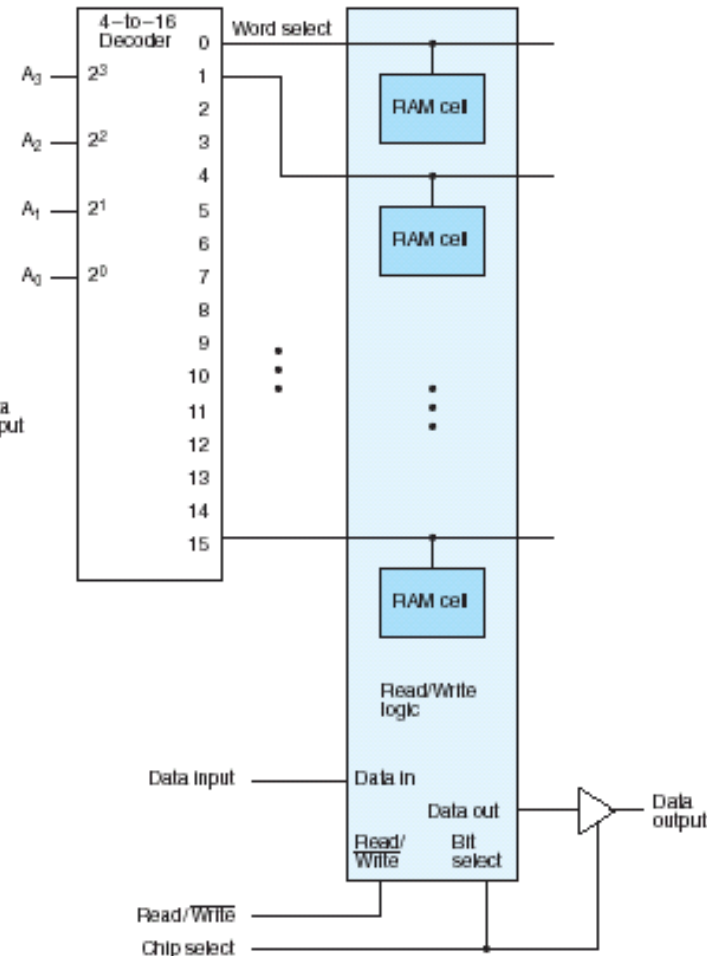
---

<u>Memory address</u>		
<u>Binary</u>	<u>Decimal</u>	Memory contents
0000000000	0	10110101 01011100
0000000001	1	10101011 10001001
0000000010	2	00001101 01000110
	•	•
	•	•
	•	•
	•	•
	•	•
1111111101	1021	10011101 00010101
1111111110	1022	00001101 00011110
1111111111	1023	11011110 00100100

# Memory Chips



(a) Symbol



(b) Block diagram

Chip select CS	Read/ <u>Write</u> R/ <u>W</u>	Memory operation
0	×	None
1	0	Write to selected word
1	1	Read from selected word

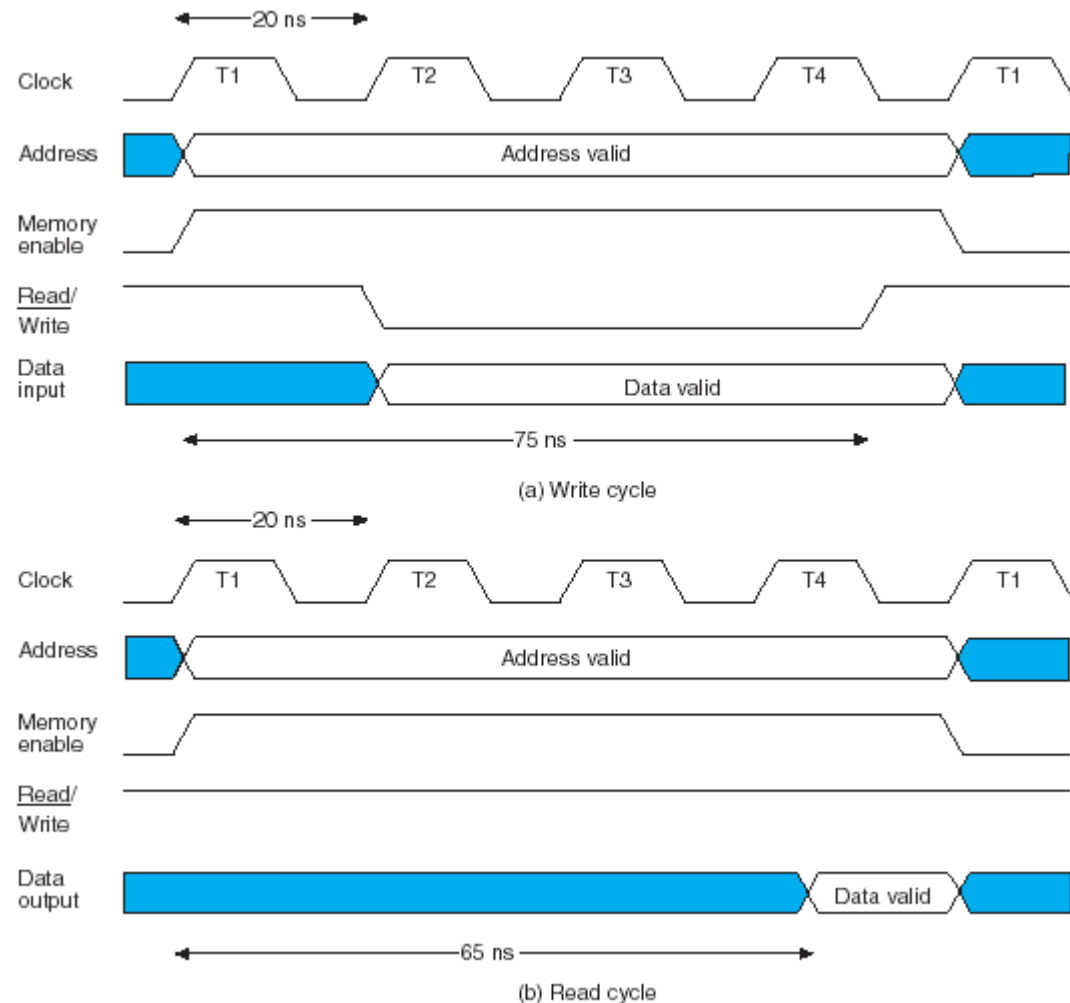
# Memory Cycle Timing Waveforms

## Memory write steps:

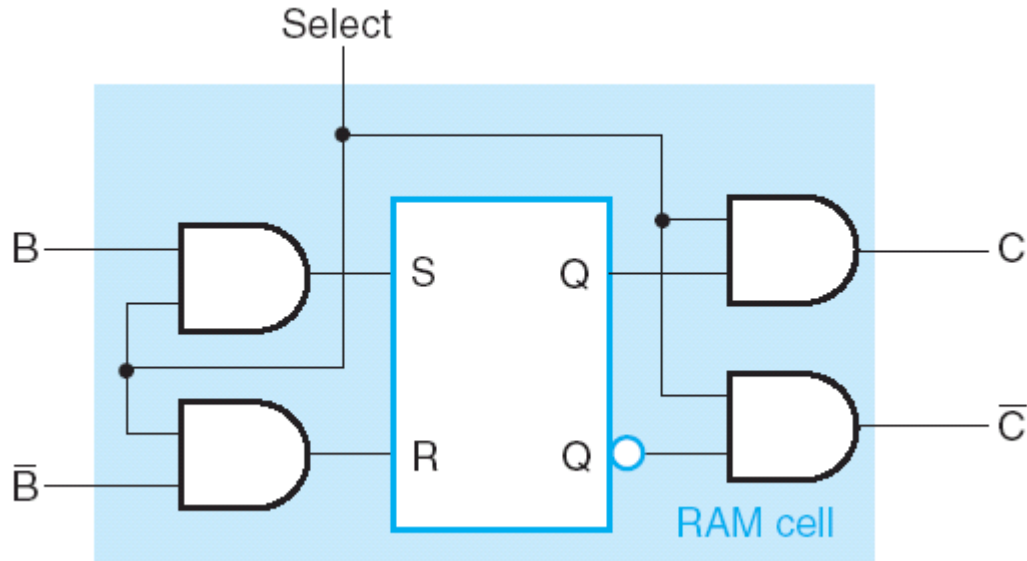
1. Apply the binary address of the desired word to the address lines
2. Apply the data bits to the data input lines
3. Activate the Write input

## Memory read steps:

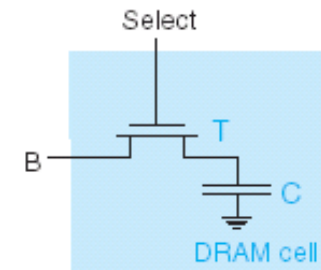
1. Apply the binary address of the desired word to the address lines
2. Activate the Read input



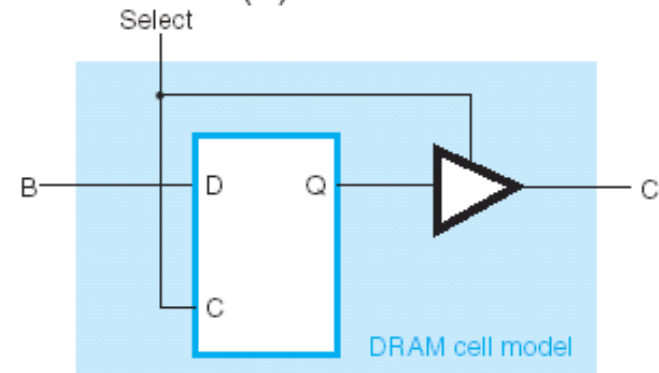
# Static RAM vs. Dynamic RAM



**Static RAM**



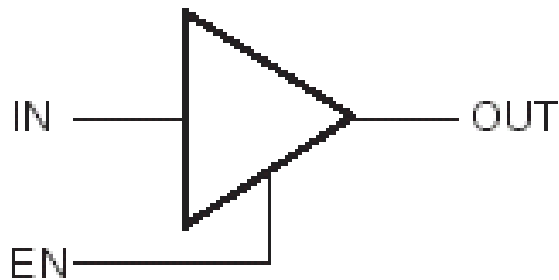
(a)



**Dynamic RAM**

# Tri-State Buffer

---



(a) Logic symbol

EN	IN	OUT
0	X	Hi-Z
1	0	0
1	1	1

(b) Truth table

Fig. 6-8 Three-state Buffer

# Programmable Logic Technologies

- ROM, PROM

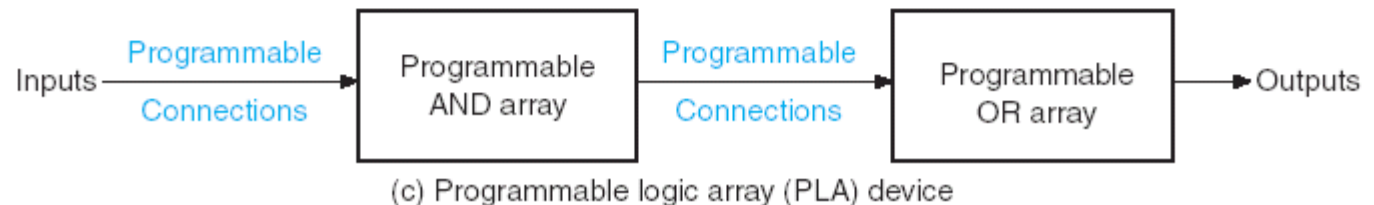
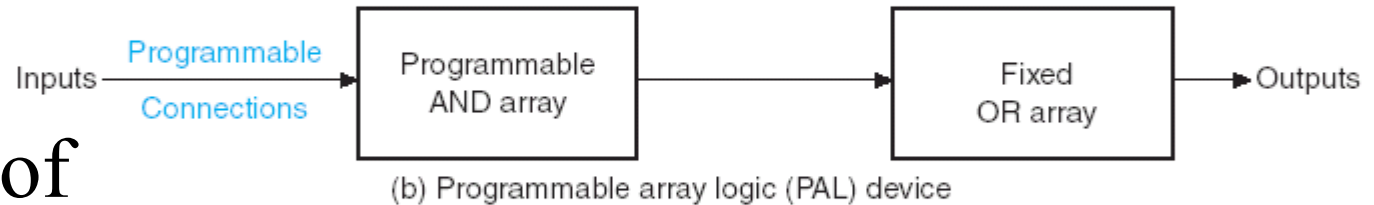
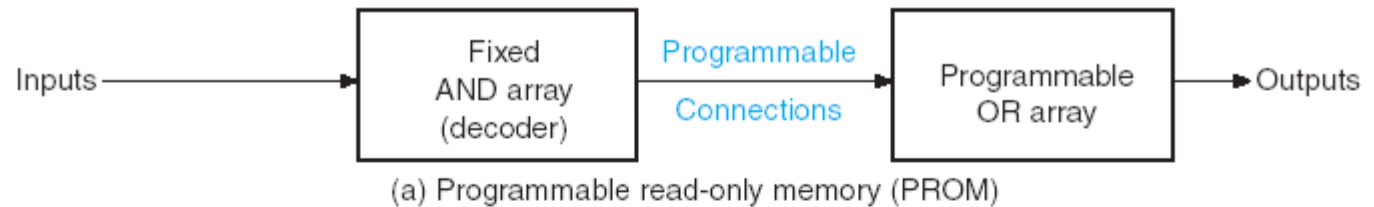
- PAL

- PLA

- CPLD

- FPGA

- Benefits of Programmable Logic



# ROM, PROM

- ROM = Read Only Memory
- Information is stored permanently
- PROM = Programmable Read Only Memory
- EEPROM = ?



A ROM is basically a device that includes a decoder and OR gates within a single unit

A ROM with  $k$ -inputs and  $n$ -outputs can be used to implement up to  $n$  Boolean functions of  $k$  inputs each

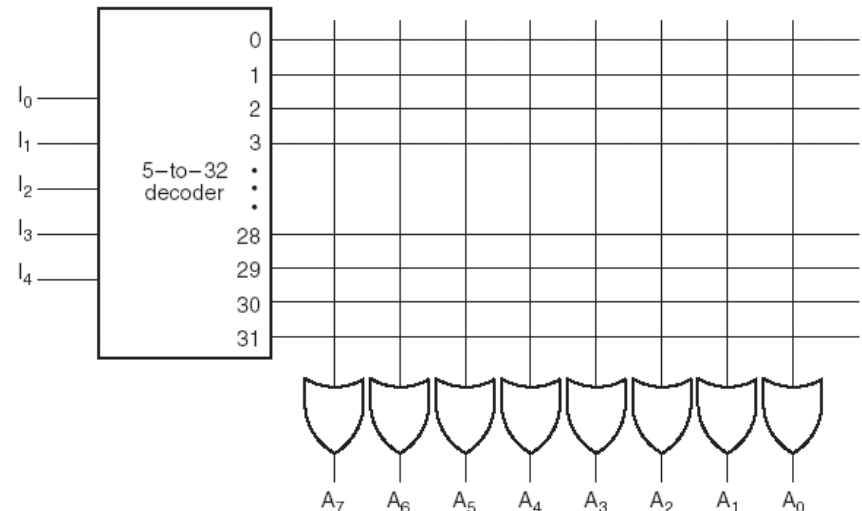


Fig. 6-20 Internal Logic of a 32 x 8 ROM

# Boolean Function Implementation Using ROM

TABLE 6-2  
ROM Truth Table (Partial)

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	0	1	1	1	0	1					
		.										
		.										
		.										
1	1	1	0	0	0	0	0					
1	1	1	0	1	1	1	1					
1	1	1	1	0	0	1	0					
1	1	1	1	1	0	0	1					

When a combinational circuit is implemented through a ROM then it is not necessary to design the logic – just the Truth Table is sufficient and the model of the ROM IC

ROM can be used in applications requiring a moderate number of inputs and outputs – code converters, complex arithmetic operations such as division, multiplication, sine, cosine, etc.

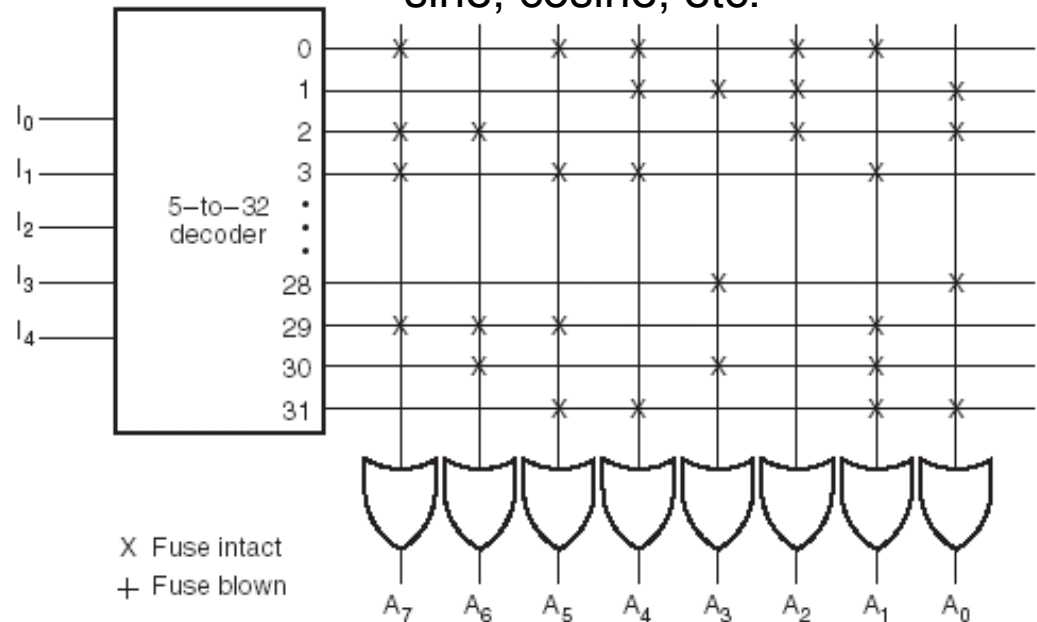
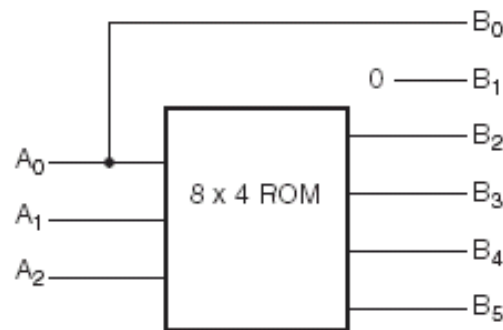


Fig. 6-21 Programming the ROM According to Table 6-2

# Another ROM Example

□ **TABLE 6-3**  
**Truth Table for Circuit of Example 6-1**

Inputs			Outputs						Decimal
A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	
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



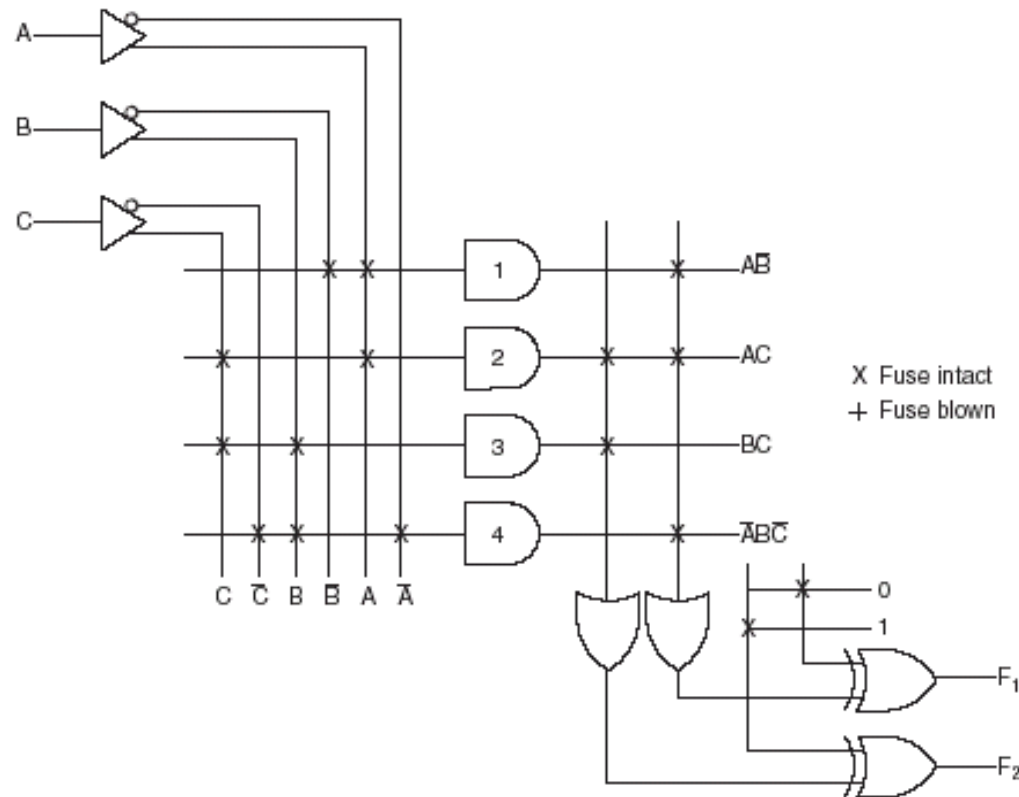
(a) Block diagram

A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	B <sub>5</sub>	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>
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

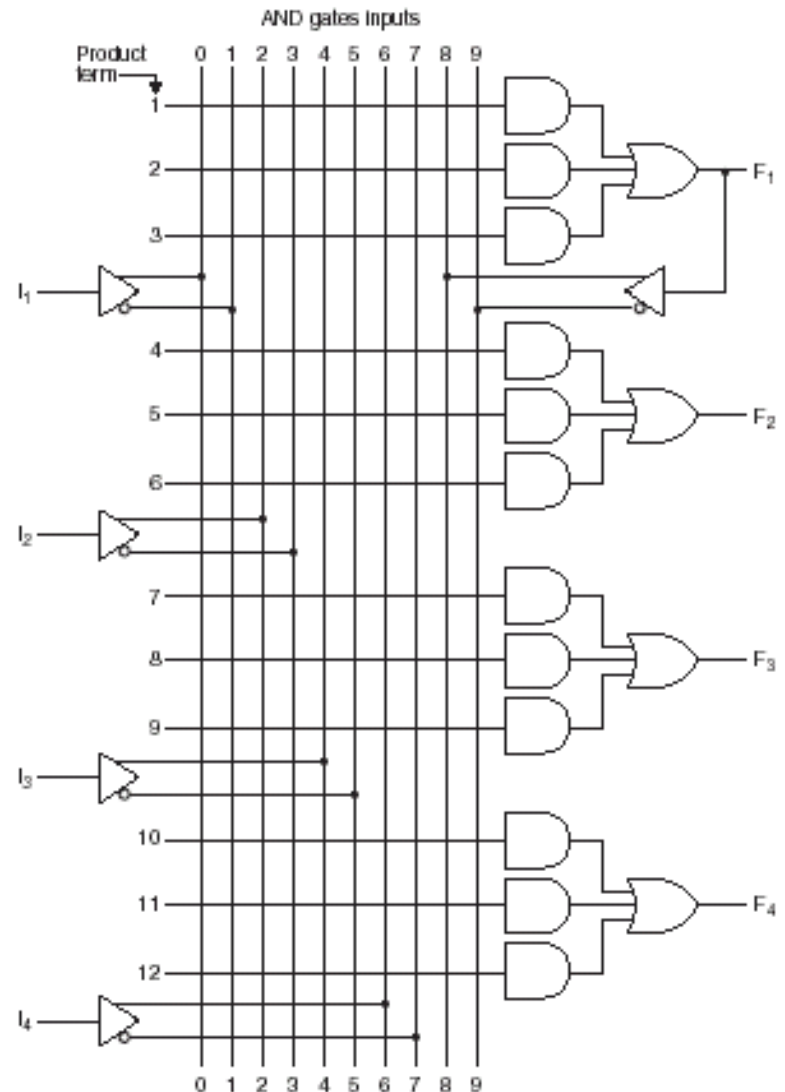
# PLA

- Programmable Logic Array
- Consists of an array of AND gate followed by an array of OR gates
- AND gates generate product terms
- OR gates sum the product terms so we get SOP implementation



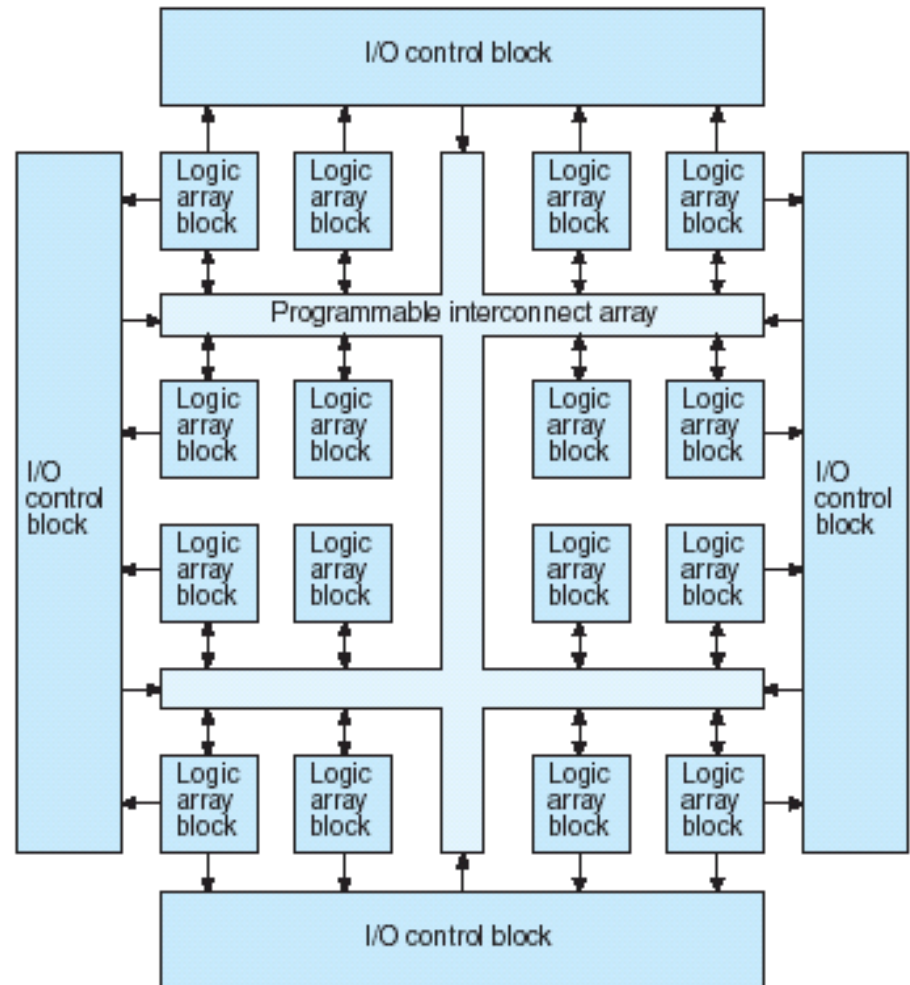
# PAL

- Programmable Array Logic
- The OR array is fixed and not programmable
- The AND array is programmable
- The design with PAL devices is facilitated by use of computer aided techniques



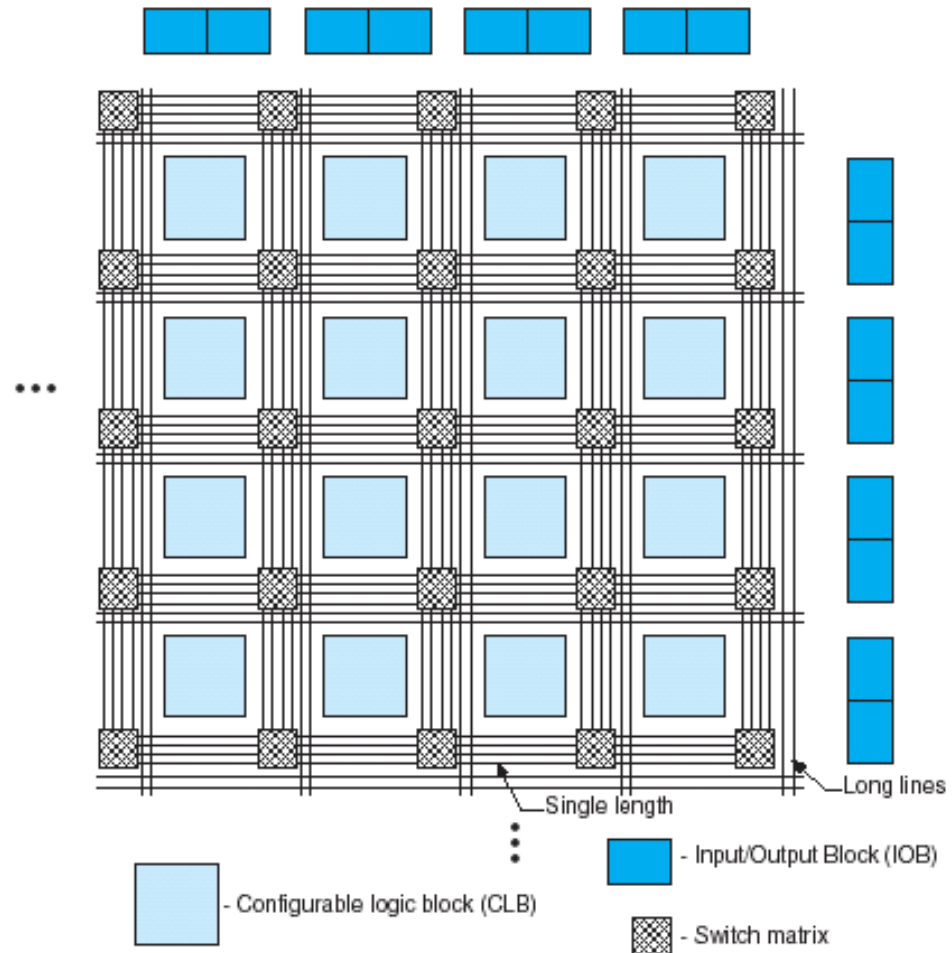
# CPLD

- Complex Programmable Logic Device
- Used for VLSI design
- Several Manufacturers - Altera, Xilinx
- Substantial amounts of uncommitted combinational logic
- Pre-implemented flip-flops
- Programmable interconnections between these two

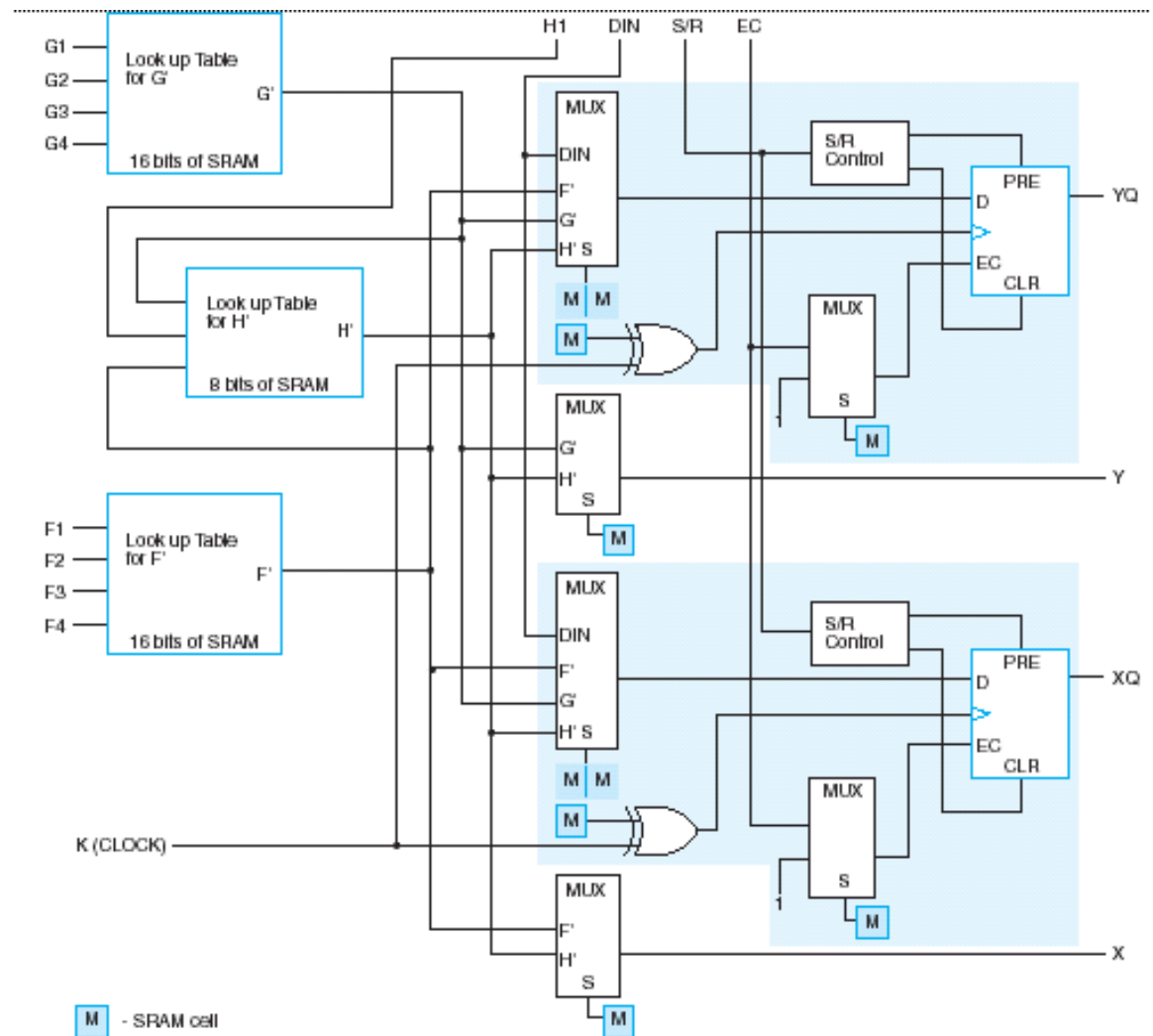


# FPGA

- Field Programmable Gate Array
- Consists of CLBs and IOBs along with Switch matrix
- Like CPLDs have:
- Substantial amounts of uncommitted combinational logic
- Pre-implemented flip-flops
- Programmable interconnections between these two



# Configurable Logic Block - CLB



# Benefits of Programmable Logic

---

- Reconfigurable Hardware – hardware that can be reprogrammed instantly (generally a software attribute)
- Quick development and debugging – faster product time to market
- Opens hardware design and development to those without fabrication facilities
- Very large scale designs can now be implemented on a single high density programmable logic chip
- Use of Intellectual Property (IP) means design reuse