Beyond SSI Logic Integrated Circuits

Combinational Logic Components

Learning Objectives

After this lecture, you should be able to…….

- **Distinguish between logic gates and components**
- **Describe and apply the listed functions:**
  - **Multiplexer**
  - **Demultiplexer**
  - **Decoder**
  - **Priority Encoder**
  - **Programmable Logic Array**
- **Design a ripple carry adder using the different components described**

Digital Components

High level digital circuit designs are normally made using collections of logic gates referred to as components, rather than using individual logic gates. The majority function can be viewed as a component.

Levels of integration (numbers of gates) in an integrated circuit (IC):

Small scale integration (SSI): 10-100 gates.

Medium scale integration (MSI): 100 to 1000 gates.

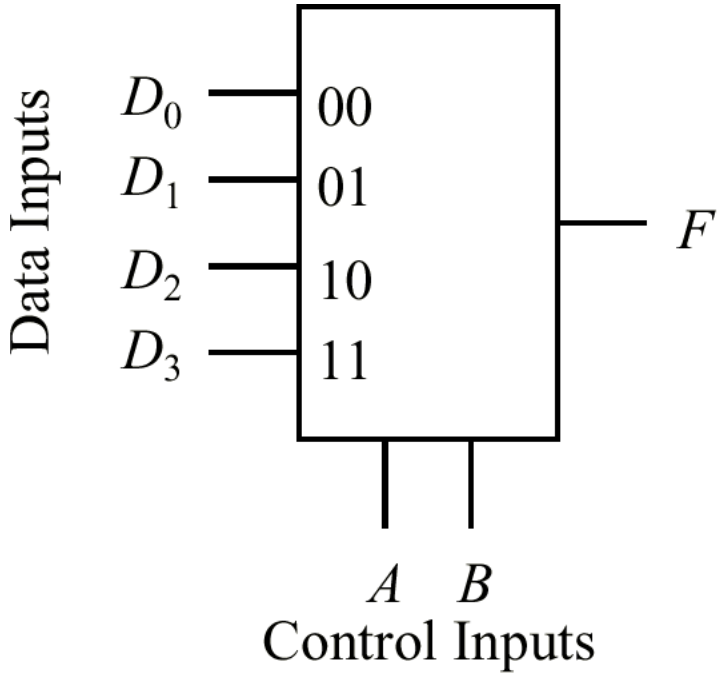
Large scale integration (LSI): 1000-10,000 logic gates.

Very large scale integration (VLSI): 10,000-upward.

These levels are approximate, but the distinctions are useful in comparing the relative complexity of circuits.

Let us consider several useful MSI components:

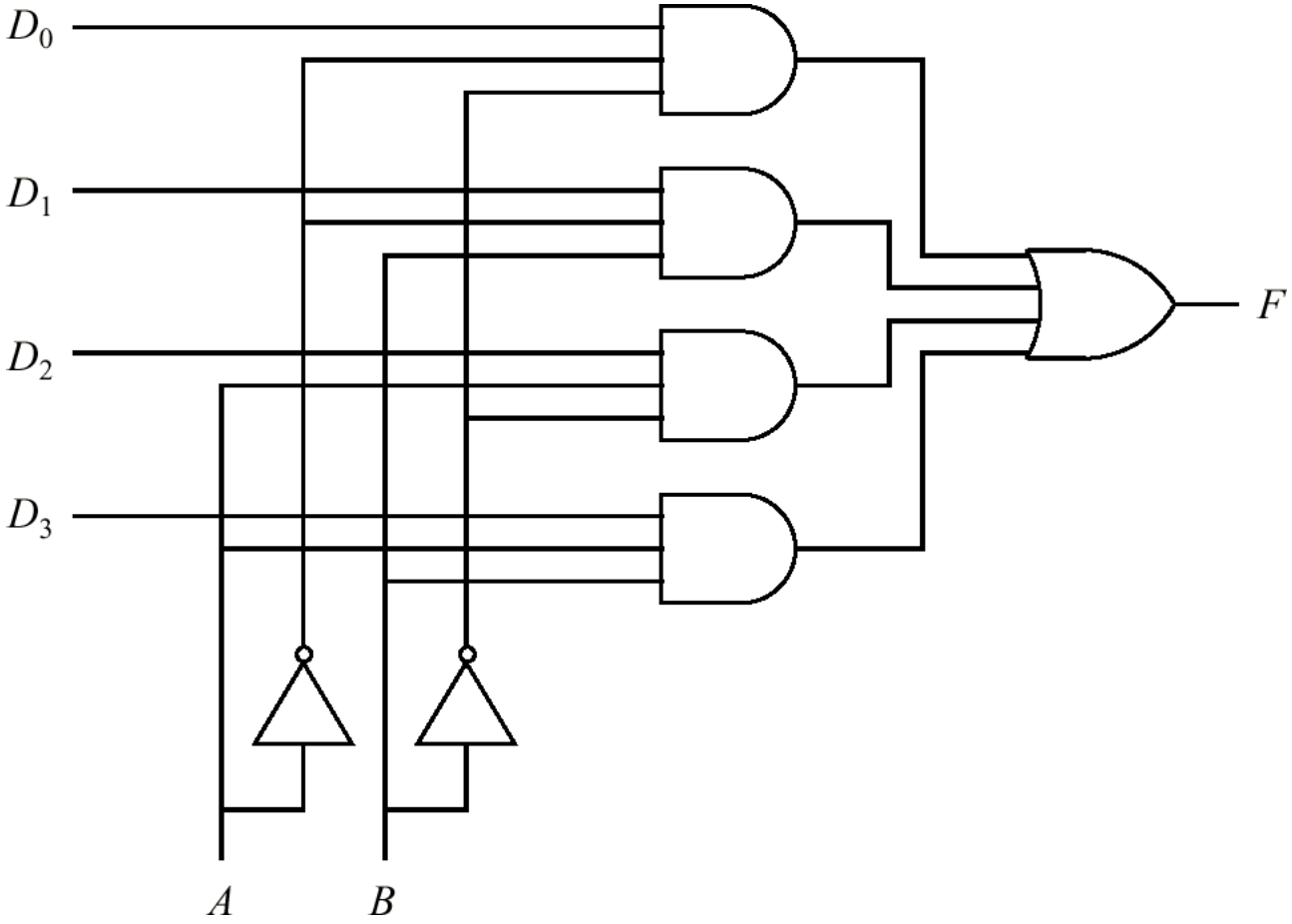
A Simple Multiplexer



A	B	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

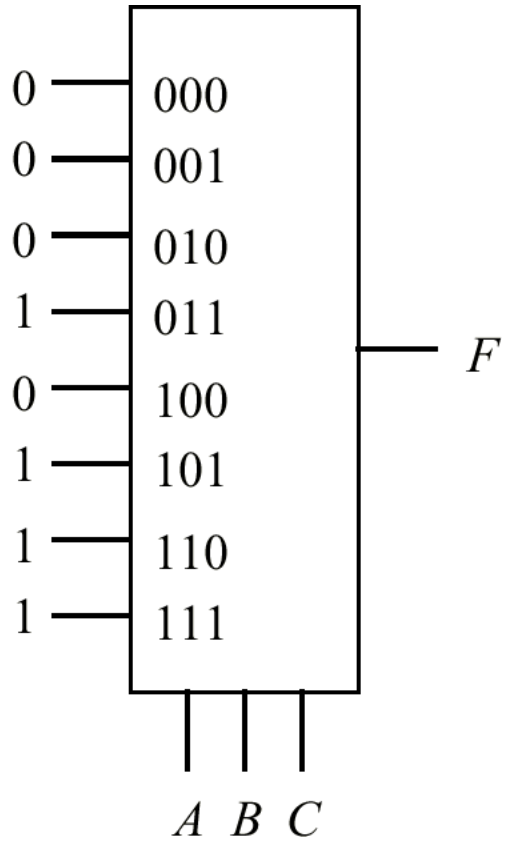
$$F = \overline{A}\,\overline{B}\,D_0 + \overline{A}\,B\,D_1 + A\,\overline{B}\,D_2 + A\,B\,D_3$$

The Multiplexer



Implementing the Majority Function with an 8-1 Mux

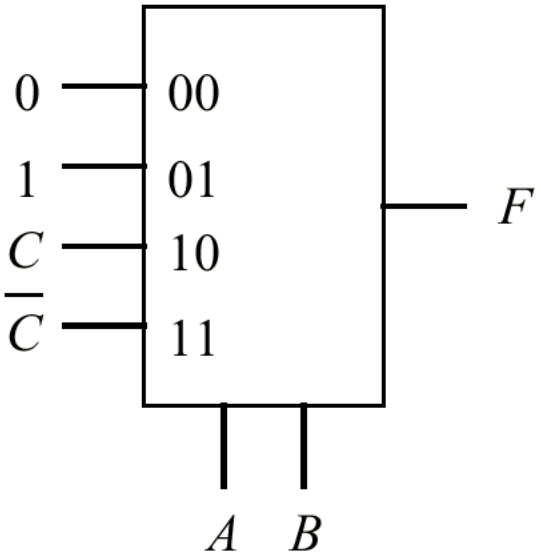
A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Principle: Use the mux select to pick out the selected minterms of the function.

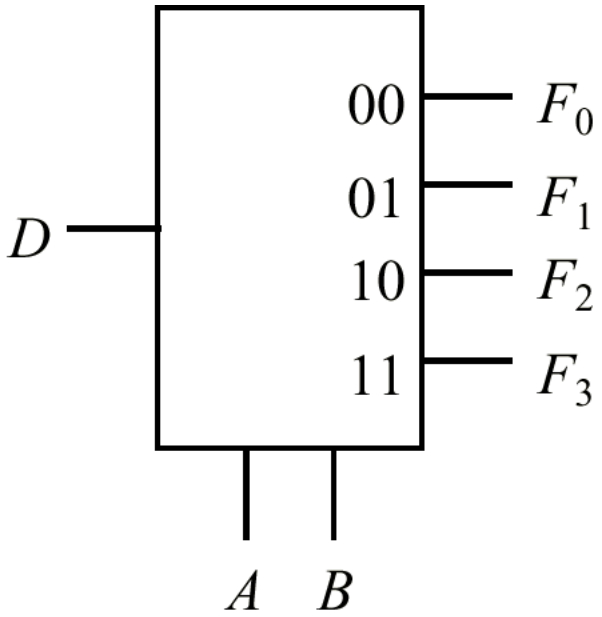
More Efficiency: Using a 4-1 Mux to Implement the Majority Function

A	B	C	F	
0	0	0	0	0
0	0	1	0	
0	1	0	1	1
0	1	1	1	
1	0	0	0	C
1	0	1	1	
1	1	0	1	$\overline{C}$
1	1	1	0	



Principle: Use the A and B inputs to select a pair of minterms. The value applied to the MUX input is selected from {0, 1, C, C} to pick the desired behavior of the minterm pair.

The Demultiplexer (DEMUX)



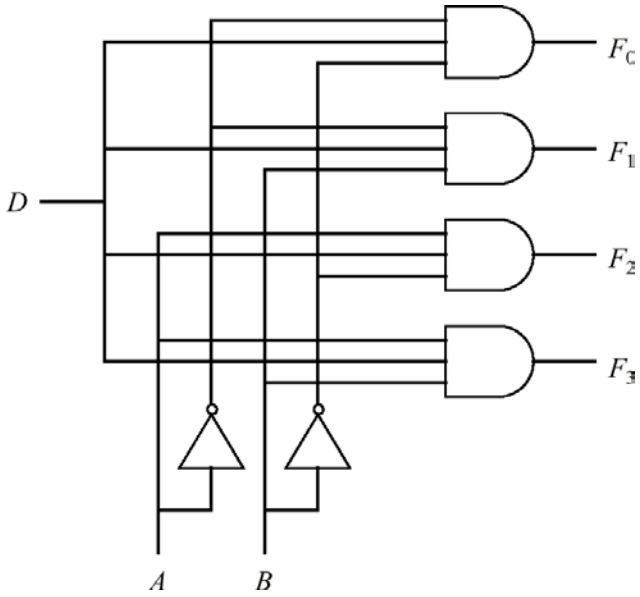
D	A	B	F_0	F_1	F_2	F_3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

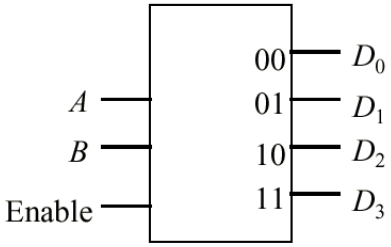
$$F_0 = D\overline{A}\,\overline{B} \qquad F_2 = DA\overline{B}$$

$$F_1 = D\overline{A}B \qquad F_3 = DAB$$

The Demultiplexer is a Decoder with an Enable Input

Compare to Fig A.29



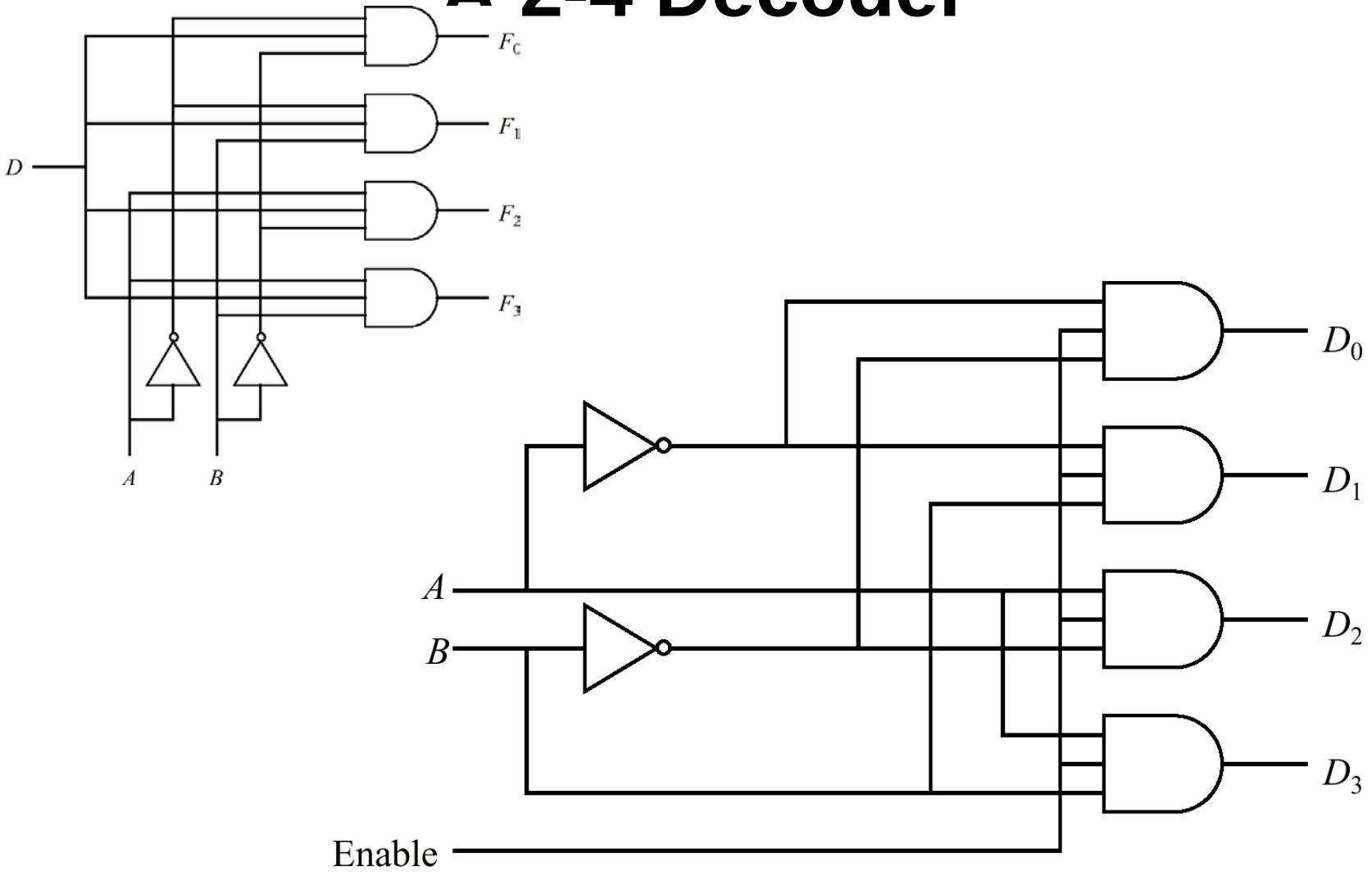


Enable = 1						
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

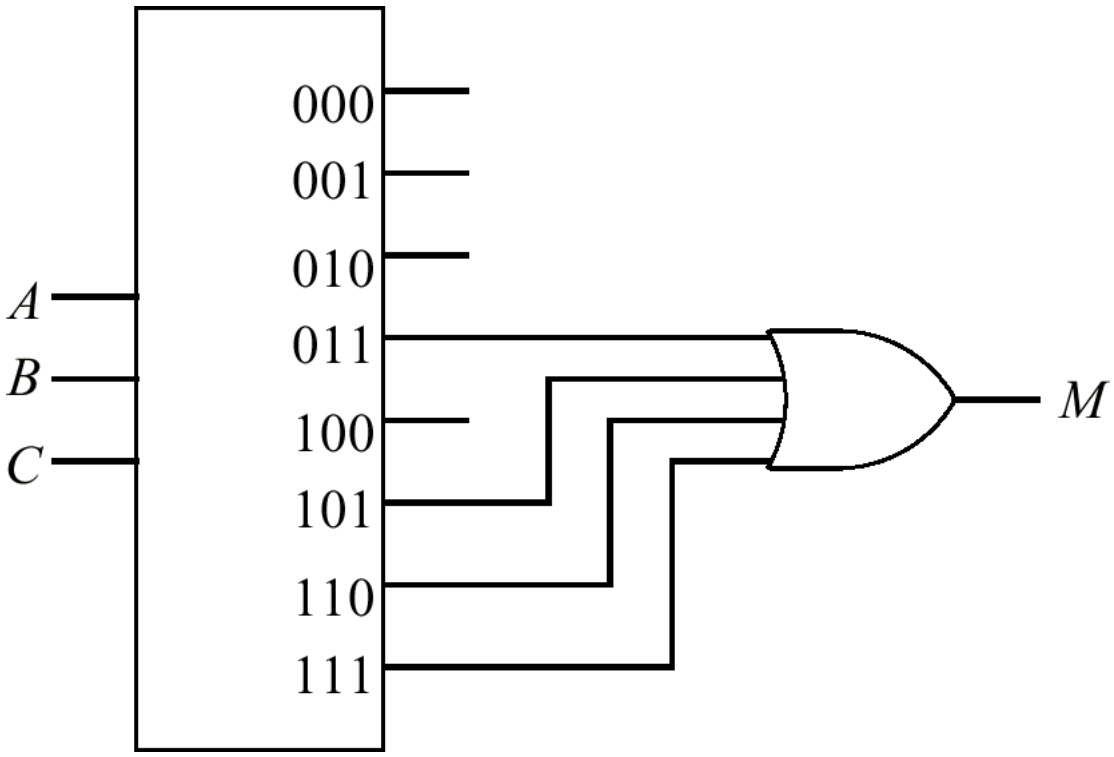
Enable = 0						
A	B	D_0	D_1	D_2	D_3
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	0

$$D_0 = \overline{A}\,\overline{B} \qquad D_1 = \overline{A}\,B \qquad D_2 = A\,\overline{B} \qquad D_3 = A\,B$$

A 2-4 Decoder



Using a Decoder to Implement the Majority Function



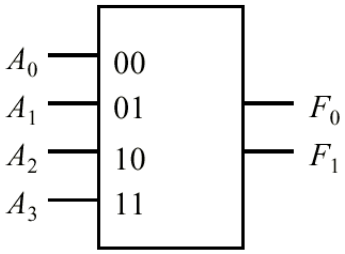
The Priority Encoder

An encoder translates a set of inputs into a binary encoding,

Can be thought of as the converse of a decoder.

A priority encoder imposes an order on the inputs.

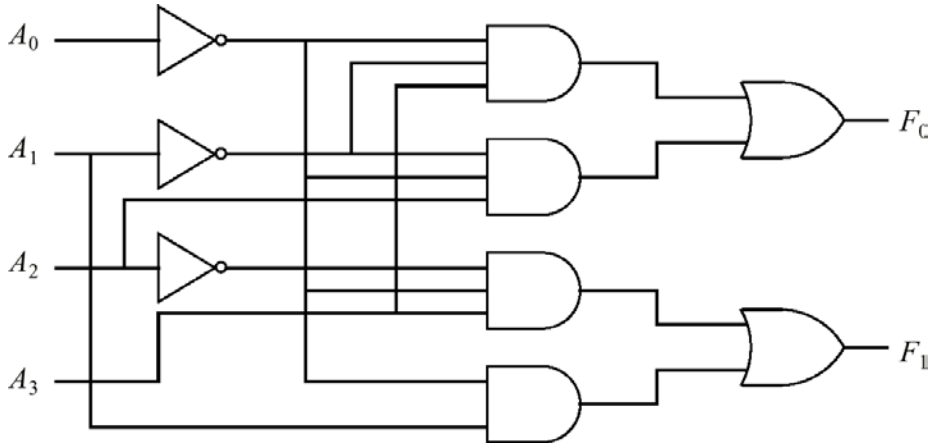
A_i has a higher priority than A_{i+1}



$$F_0 = \overline{A}_0 \overline{A}_1 A_3 + \overline{A}_0 \overline{A}_1 A_2$$

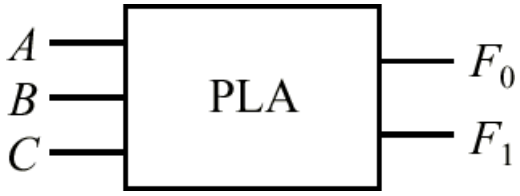
$$F_1 = \overline{A}_0 \overline{A}_2 A_3 + \overline{A}_0 A_1$$

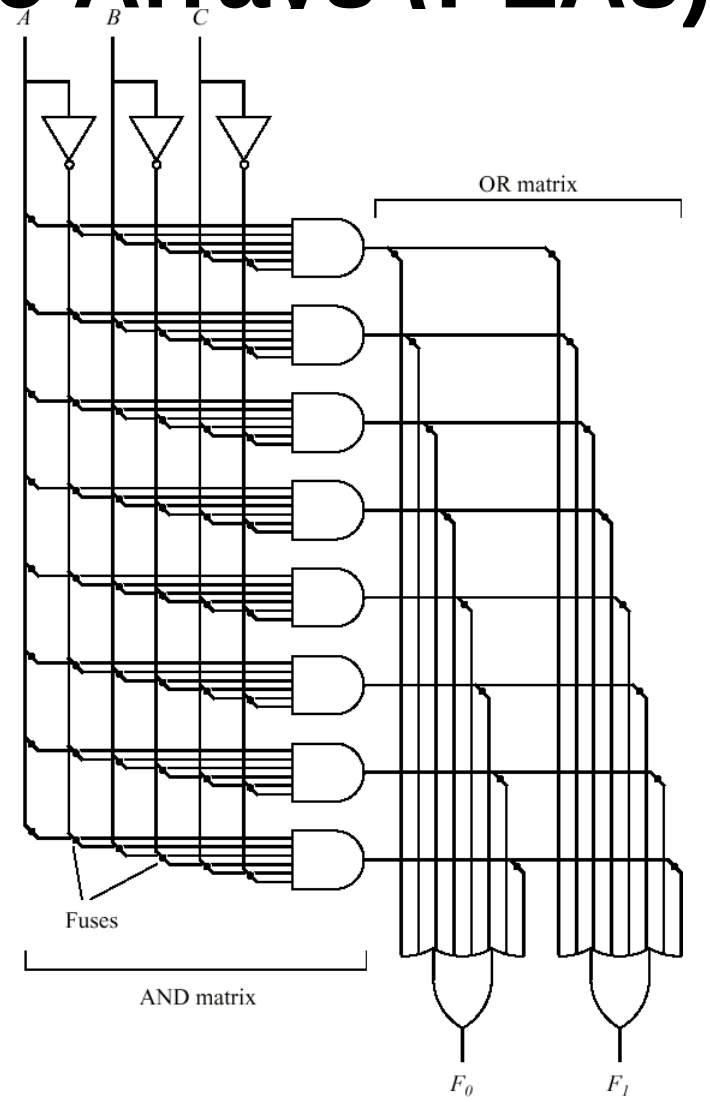
A_0	A_1	A_2	A_3	F_0	F_1
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



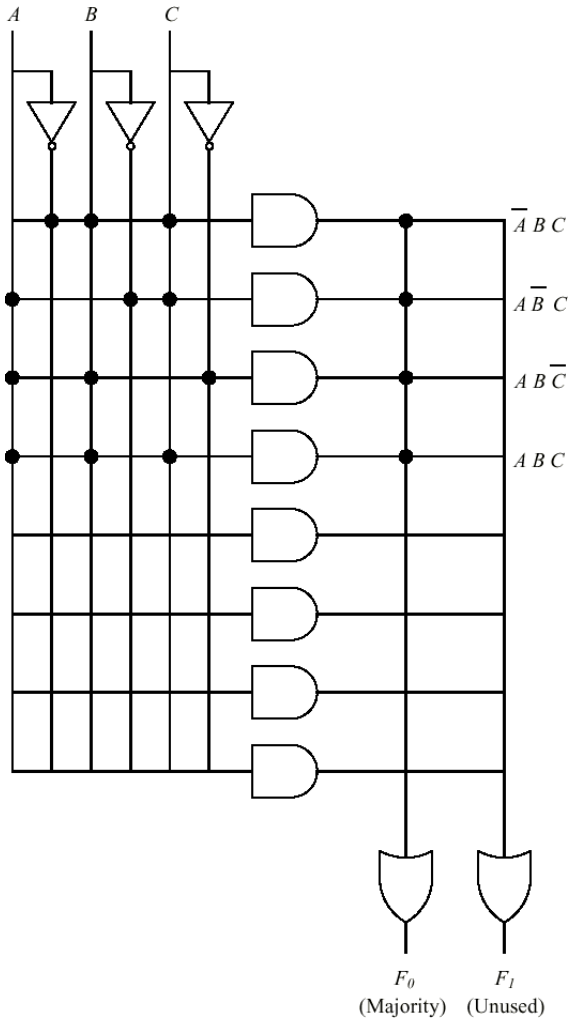
Programmable Logic Arravs (PLAs)

A PLA is a customizable AND matrix followed by a customizable OR matrix:





Using a PLA to Implement the Majority Function



Using PLAs to Implement an Adder

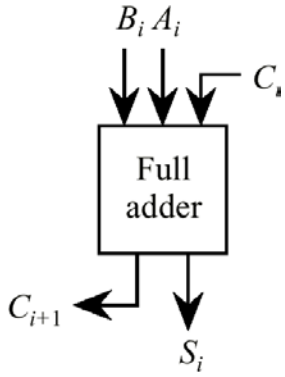
Carry In →	0	0	0	0	1	1	1
Operand A →	0	0	1	1	0	0	1
Operand B →	+ 0	+ 1	+ 0	+ 1	+ 0	+ 1	+ 0
	0 0	0 1	0 1	1 0	0 1	1 0	1 0

Carry Out — Sum

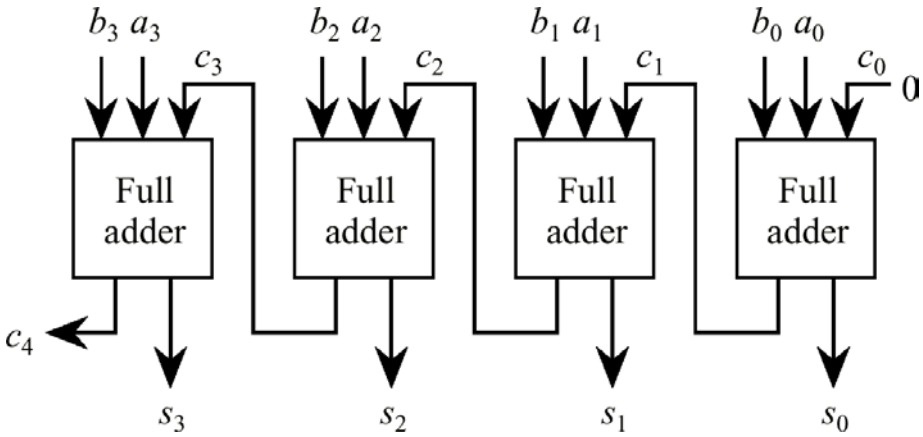
Example:

Carry	1	0	0	0
Operand A	0	1	0	0
Operand B +	0	1	1	0
Sum	1	0	1	0

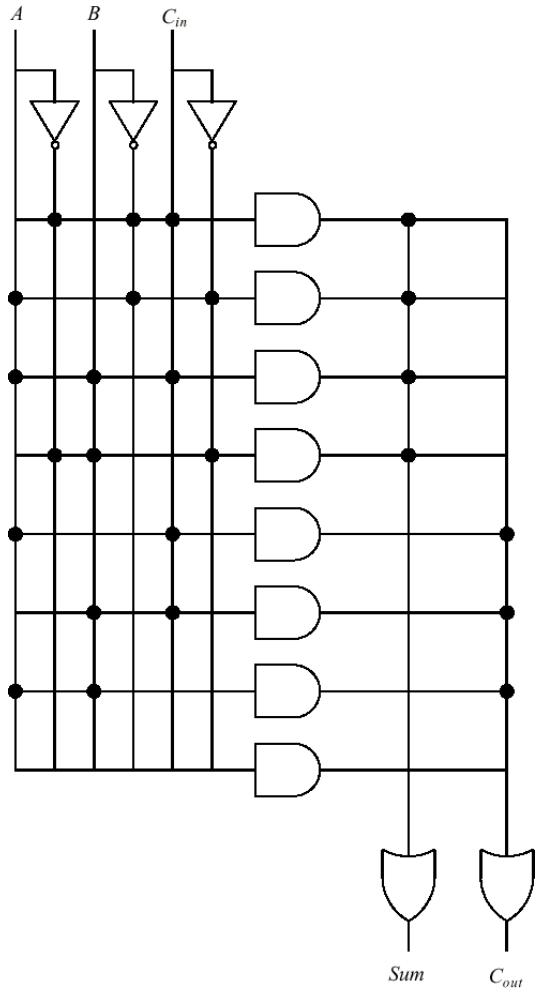
A_i	B_i	C_i	S_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



A Multi-Bit Ripple-Carry Adder



PLA Realization of a Full Adder



Objectives Completed

- Differentiated logic gates from components
- Described the function of the Multiplexer (MUX), Demultiplexer (DEMUX), Decoder, Priority Encoder, Programmable Logic Array
- Designed a ripple carry adder using the different components described

Next time

- **Making smaller circuits with logic reduction techniques**
- **Characterizing propagation delay in gates**
- **OR gate and MUX decomposition methods of logic reduction**

SSI : small scale integrated circuit