ALU Basics

Digital Logic Design (CC131)

BSCS 2nd Semester Spring-2026

By

Syed Zulfiqar Ali Shah,

Associate Professor (CS) / HoD

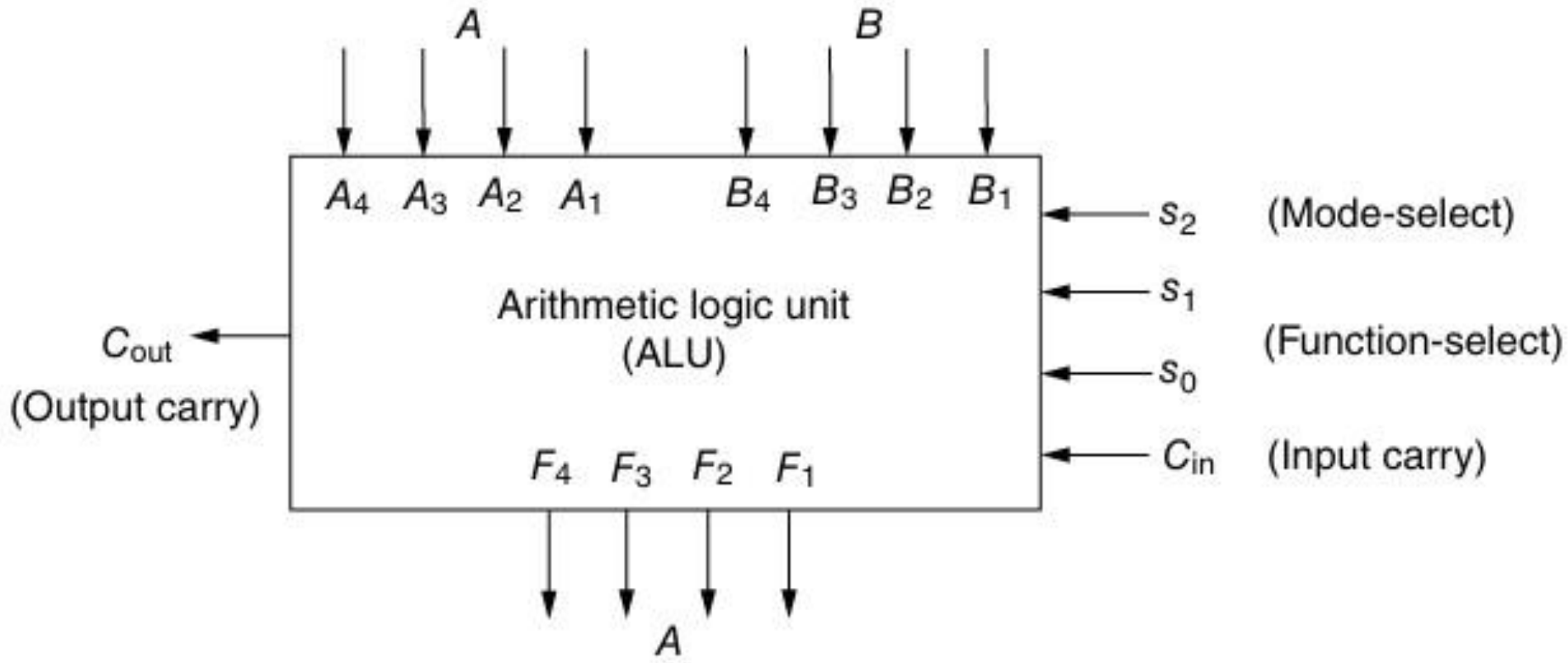
Govt. College of Management Sciences, Kohat

www.linkedin.com/in/syedzulfiqaralishah

ARITHMETIC LOGIC UNIT

- An arithmetic logic unit (ALU) is a multi-operation, combinational-logic digital circuit.
- It can perform a set of basic arithmetic operations and a set of logic operations.
- The ALU has a number of selection lines to select a particular operation in the unit.
- The selection lines are decoded within the ALU so that k selection variables can specify up to 2^k distinct operations.

ARITHMETIC LOGIC UNIT



Block diagram of a 4-bit ALU

ARITHMETIC LOGIC UNIT

Consider the 4-bit ALU Block Diagram.

- The four data inputs from A are combined with the four inputs from B to generate an operation at the F outputs.
- The mode-select input $\mathbf{s_2}$ distinguishes between arithmetic and logic operations. (0 = arithmetic , 1 = logic)
- The two function-select inputs $\mathbf{s_1}$ and $\mathbf{s_0}$ specify the particular arithmetic or logic operation to be generated.
- With three selection variables, it is possible to specify four arithmetic operations (with s_2 in one state) and four logic operations (with s_2 in the other state).
- The input and output carries have meaning only during an arithmetic operation.

ARITHMETIC LOGIC UNIT

The design of a typical ALU will be carried out in three stages.

- First, the design of the arithmetic section will be undertaken.
- Second, the design of the logic section will be considered.
- Finally, the arithmetic section will be modified so that it can perform both arithmetic and logic operations.

Design of Arithmetic Circuit

Design of Arithmetic Circuit

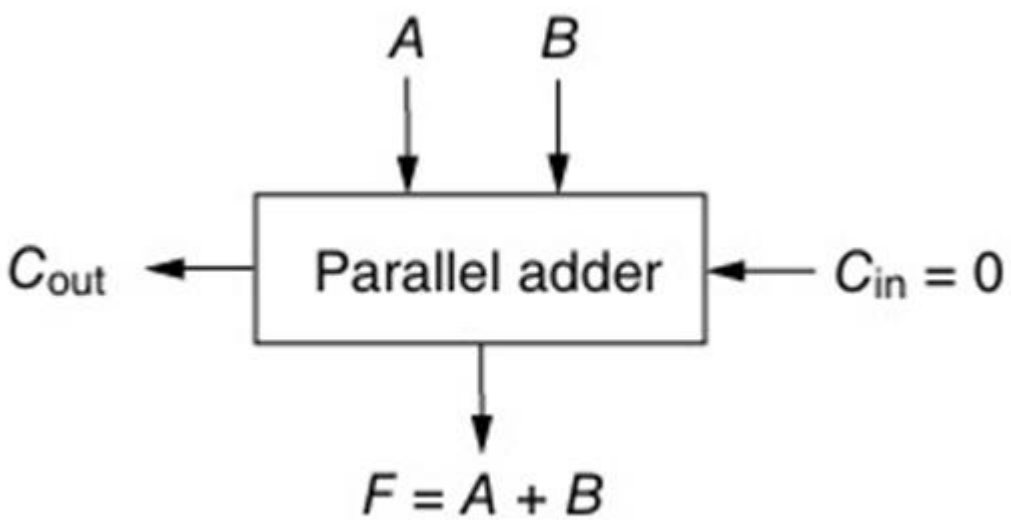
- The basic component of the arithmetic section of an ALU is a parallel adder.

- By controlling the data inputs (A & B , C_{in}) to the parallel adder, it is possible to obtain different types of arithmetic operations.

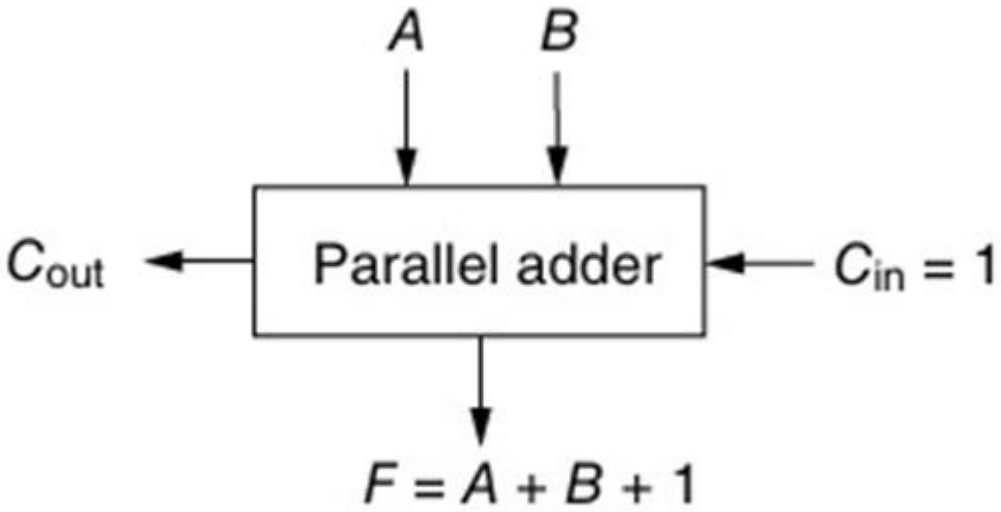
- 8 basic arithmetic operations include;

1. Addition	5. Transfer A (without Carry)
2. Addition with Carry	6. Increment A
3. A plus 1's Complement of B	7. Decrement A
4. Subtraction	8. Transfer A (with Carry)

Basic Operations

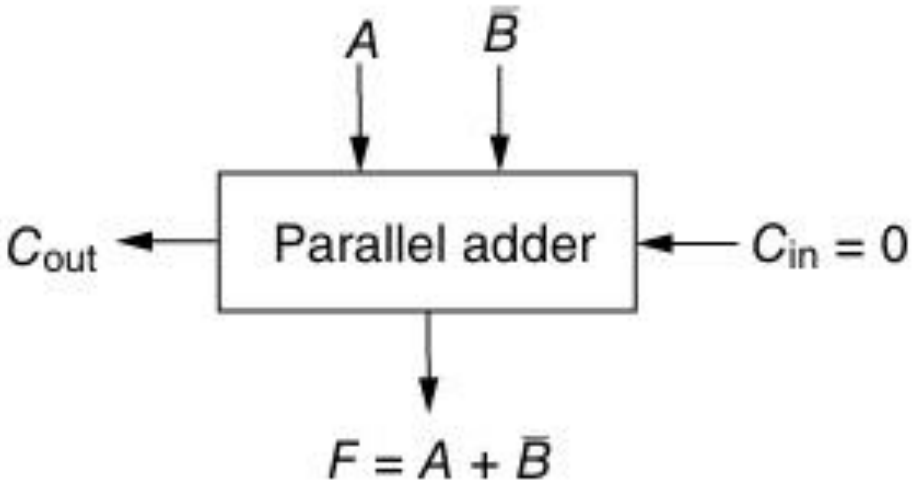


(a) Addition

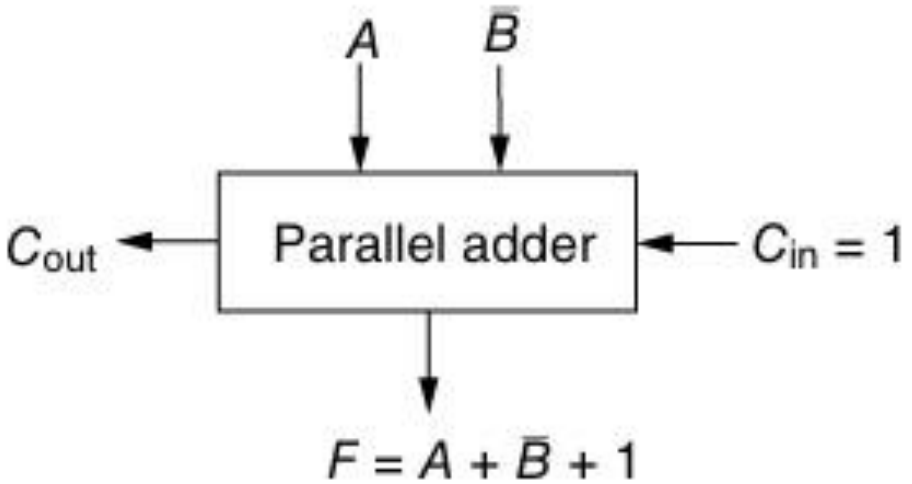


(b) Addition with carry

Basic Operations

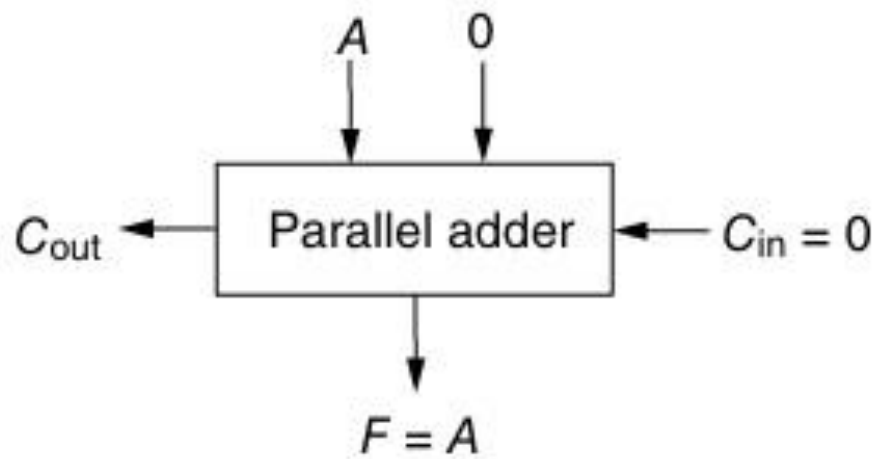


(c) A plus 1's complement of B

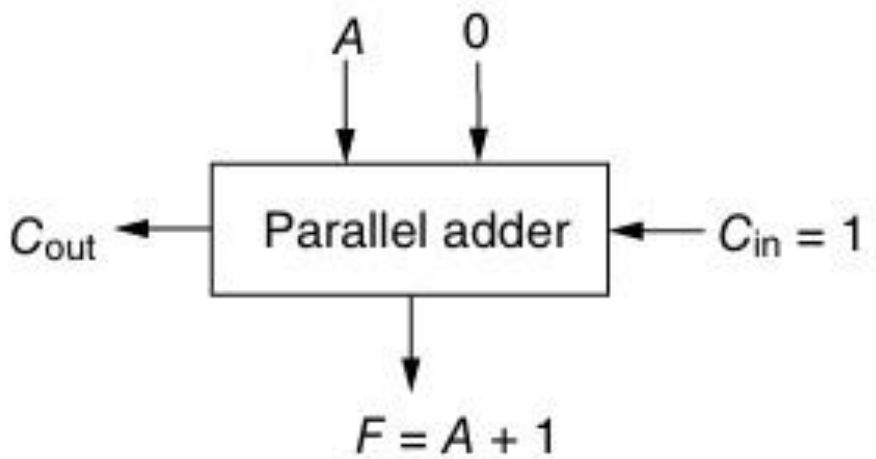


(d) Subtraction

Basic Operations

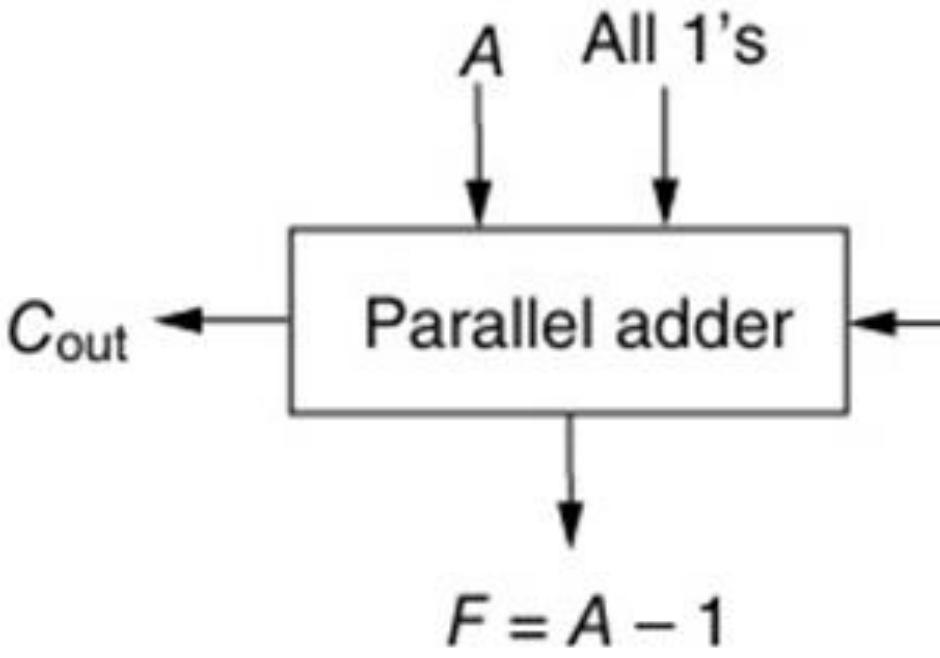


(e) Transfer A



(f) Increment A

Basic Operations



(g) Decrement A

Example (4-bit): Decrement A

Let:

$$A = 0110_2 = 6$$

Add all 1's: 1111 = 15

Then

$$\begin{array}{r} 0110 \\ +\ 1111 \\ \hline 1\ \ 0101 \end{array}$$

Arithmetic circuit keeps only the lower 4 bits:

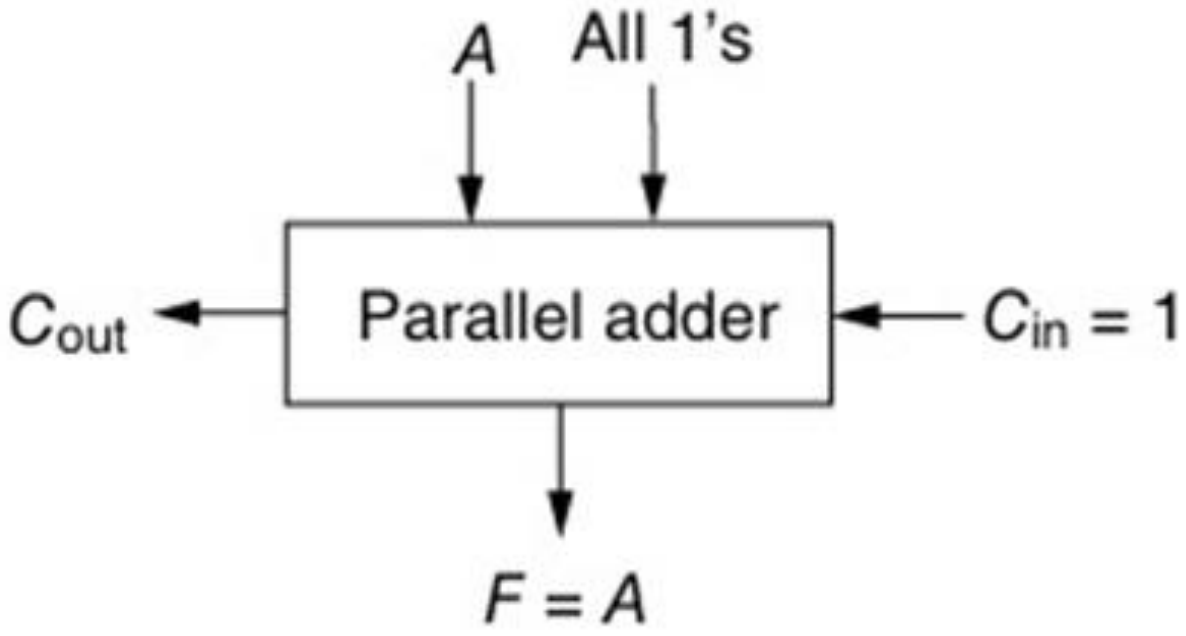
$$\mathbf{0101_2 = 5}$$

Thus:

$$6 - 1 = 5$$

So the operation is **Decrement A**.

Basic Operations



(h) Transfer *A*

Example (4-bit): Transfer A

Let:
A = 0110_2

Then:

$$\begin{array}{rr} & 0110 \\ + & 1111 \\ + & \quad 1 \\ \hline 1 & 0110 \end{array}$$

Discard the carry-out:

$$\mathbf{0110}_2$$

Result:

F = A

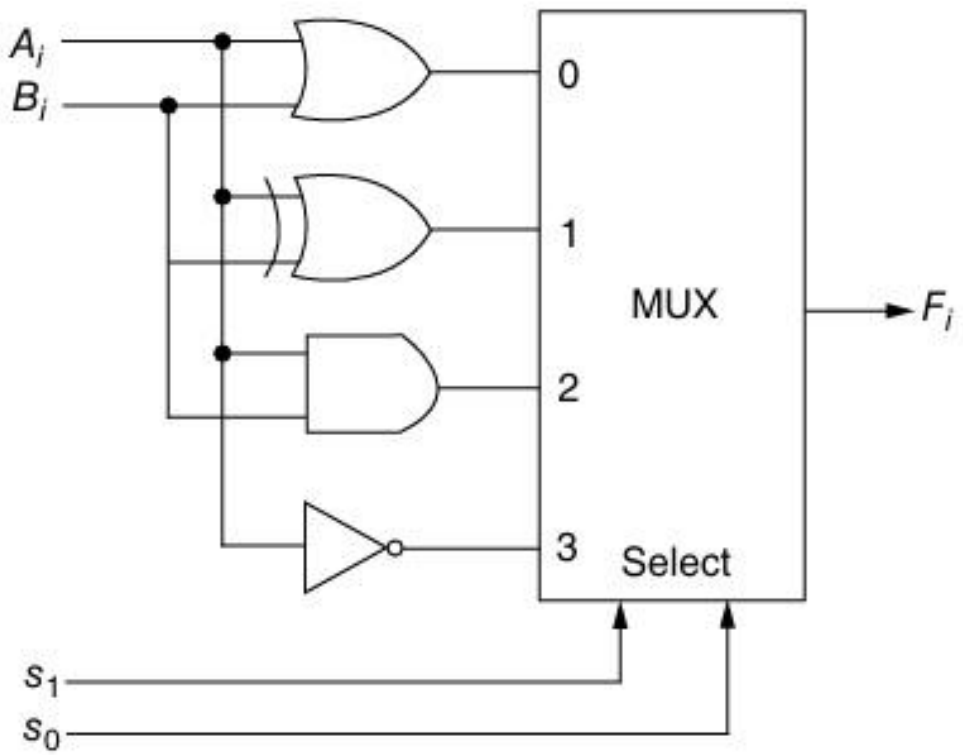
Again, the output is unchanged.

Function table for the arithmetic circuit

Function select			Y equals	Output equals	Function
s_1	s_0	C_{in}			
0	0	0	0	$F = A$	Transfer A
0	0	1	0	$F = A + 1$	Increment A
0	1	0	B	$F = A + B$	Add B to A
0	1	1	B	$F = A + B + 1$	Add B to A plus 1
1	0	0	$\overline{B}$	$F = A + \overline{B}$	Add 1's complement of B to A
1	0	1	$\overline{B}$	$F = A + \overline{B} + 1$	Add 2's complement of B to A
1	1	0	All 1's	$F = A - 1$	Decrement A
1	1	1	All 1's	$F = A$	Transfer A

DESIGN OF LOGIC CIRCUIT

LOGIC UNIT



Logic diagram

s_1	s_0	Output	Operation
0	0	$F_i = A_i + B_i$	OR
0	1	$F_i = A_i \oplus B_i$	XOR
1	0	$F_i = A_i B_i$	AND
1	1	$F_i = A'_i$	NOT

Function table

LOGIC UNIT

- The logic unit performs **OR, XOR, AND, and NOT** operations.
- All four operations are generated simultaneously.
- A **MUX** selects the desired operation.
- Two selection lines (s_1,s_0) determine which operation appears at the output.
- Repeating the stage for all bits creates an **n-bit logic unit**.

Logic Operations

Inputs

For each bit position i, the logic unit receives:

- Ai = bit from operand A
- Bi = bit from operand B

Example:

A = 1010
B = 1100

The logic unit works on each pair of bits separately:

A_3 B_3
A_2 B_2
A_1 B_1
A_0 B_0

Logic Operations

Generation of Four Logic Operations

The circuit simultaneously generates four possible outputs:

1. OR Operation

$$F_i = A_i + B_i$$

Example:

1 OR 0 = 1

0 OR 0 = 0

Logic Operations

2. XOR Operation

$$F_i = A_i \oplus B_i$$

Output is 1 when inputs are different.

Example:

1 XOR 0 = 1

1 XOR 1 = 0

Logic Operations

3. AND Operation

$$Fi = Ai\ Bi$$

Output is 1 only when both inputs are 1.

Example:

1 AND 1 = 1
1 AND 0 = 0

Logic Operations

4. NOT Operation

$$Fi = \overline{A}_i$$

Complements (inverts) A.

Example:

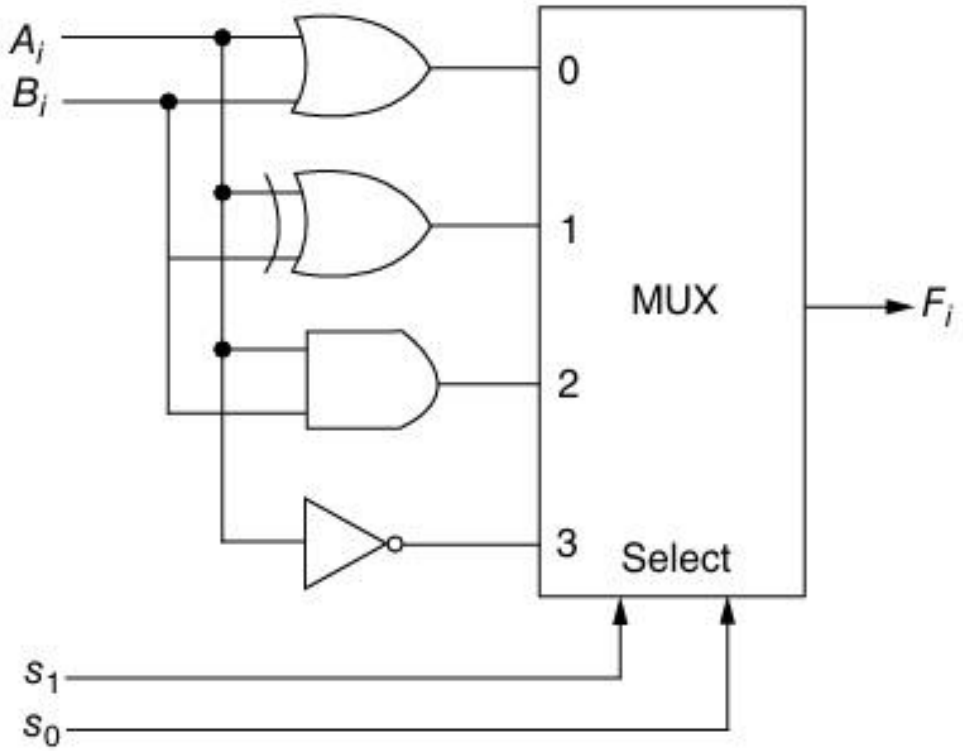
A = 1 → NOT A = 0
A = 0 → NOT A = 1

Logic Operations

Multiplexer (MUX) Chooses One Operation

- All four results are available at the same time.
- A **4-to-1 Multiplexer (MUX)** selects which result goes to the output.
- Two selection lines are used:
  - s_1
  - s_0

Logic Operations



Logic diagram

s_1	s_0	Output	Operation
0	0	$F_i = A_i + B_i$	OR
0	1	$F_i = A_i \oplus B_i$	XOR
1	0	$F_i = A_i B_i$	AND
1	1	$F_i = A'_i$	NOT

Function table

One stage of Logic circuit

Logic Operations

Example

Suppose: A = 1010
B = 1100

If s_1s_0 = 00 (OR)

1010
1100

1110

Output = **1110**

If s_1s_0 = 01 (XOR)

1010
1100

0110

Output = **0110**

Logic Operations

Example

A = 1010

B = 1100

If s_1s_0 = 10 (AND)

1010
1100

1000

Output = **1000**

If s_1s_0 = 11 (NOT A)

A = 1010

Output = **0101**

SUMMARY

An ALU combines an Arithmetic Circuit and a Logic Unit into a single hardware block, enabling a processor to perform mathematical calculations and logical decisions using the same input data.

Thank You