

UNIT 9 ALU

Presented by

**Er.Harendra Bikram Shah
(Lecturer)**

OUTLINES

- Introduction
- ALU and its Design
- status Register
- Shifter
- Processor Unit
- Accumulator

ALU and Its Design

- ✓ An arithmetic logic unit (ALU) is a major component of the central processing unit of a computer system that does all arithmetic and logic operations that need to be done on instruction.

This unit consists of two subsections namely, Arithmetic Section and Logic Section

Arithmetic Section

- ✓ Function of arithmetic section is to perform arithmetic operations like addition, subtraction, multiplication, and division. All complex operations are done by making repetitive use of the above operations.

Logic Section

- ✓ Function of logic section is to perform logic operations such as comparing, selecting, matching, and merging of data.

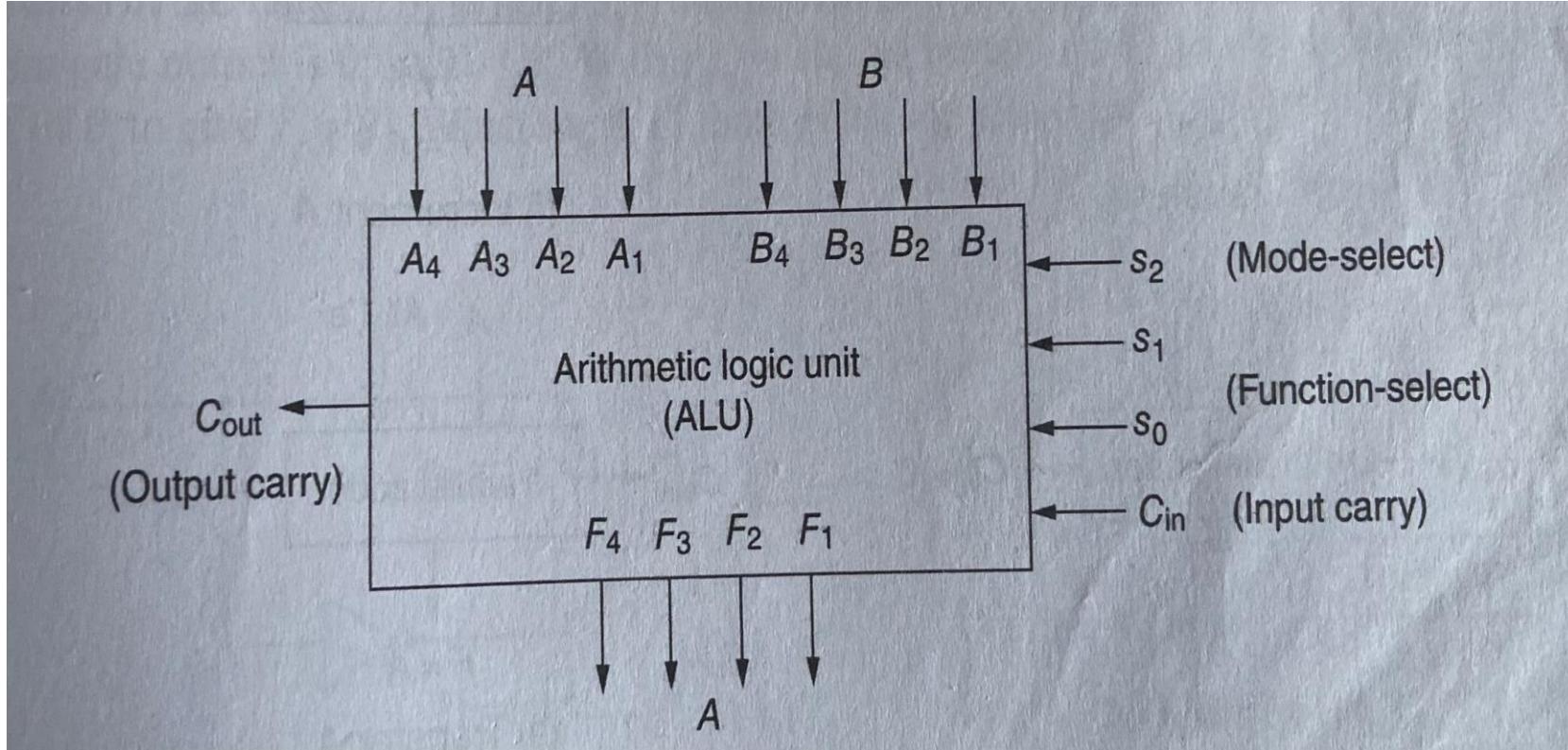
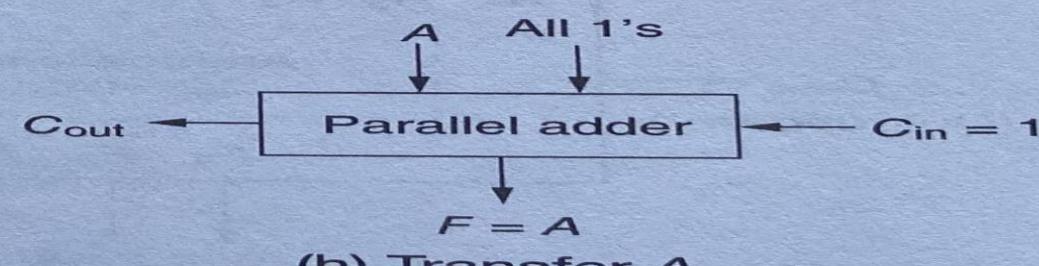
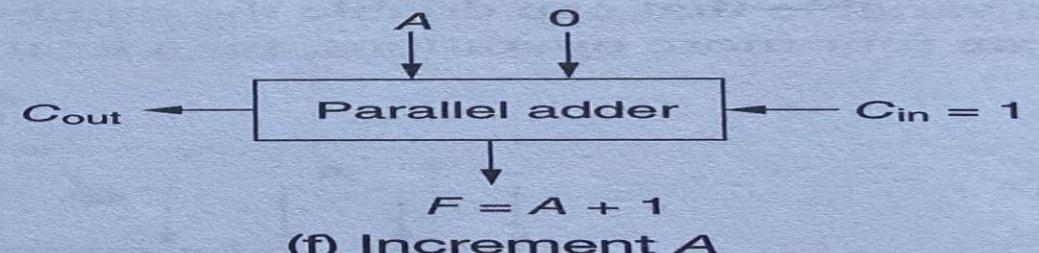
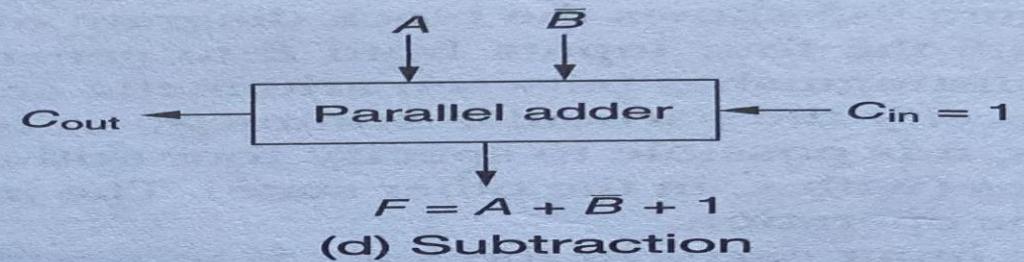
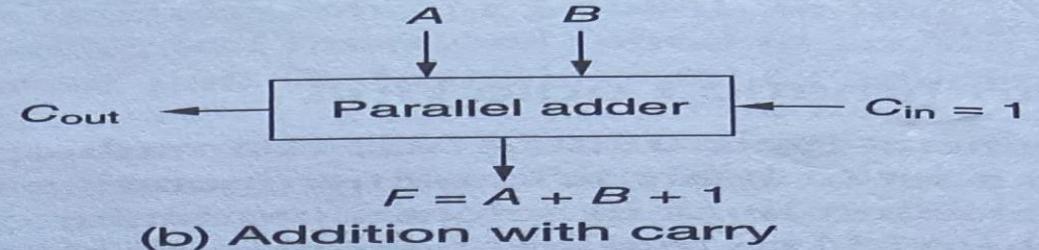
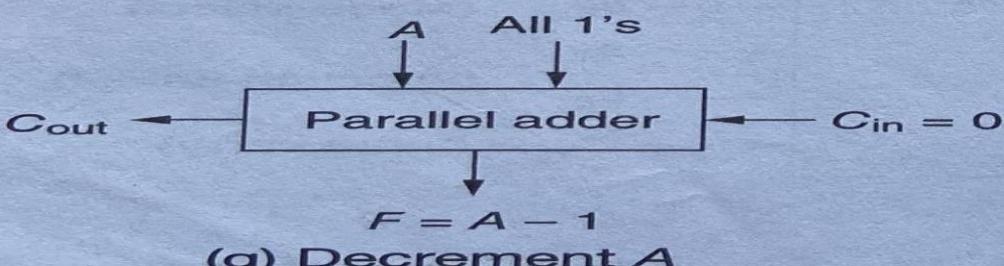
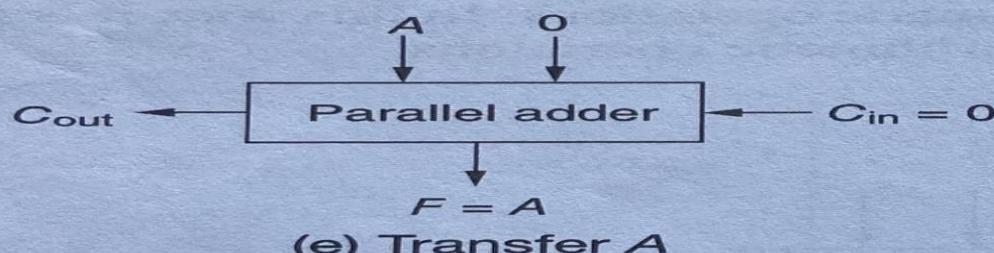
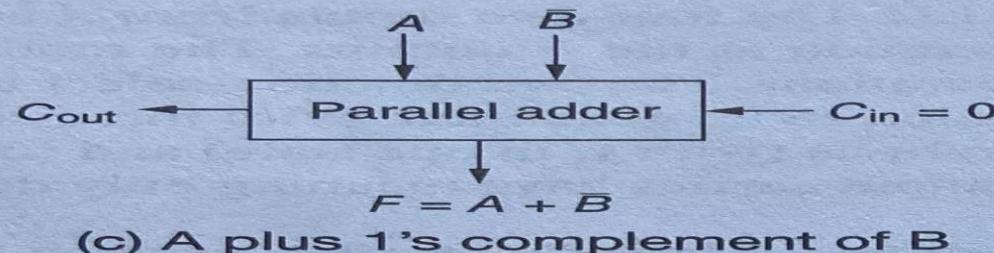
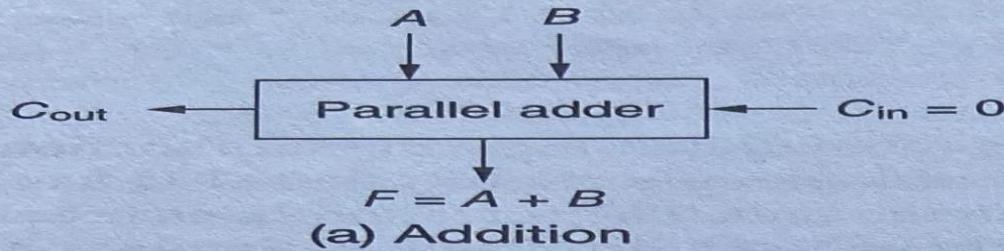


Fig: Block Diagram of 4-bit ALU

- ✓ The above figure shows the block diagram of 4-bit ALU.
- ✓ The four data inputs from A are combined with the four inputs of B to generate the operation at the F outputs.
- ✓ The mode select input s_2 distinguishes between arithmetic or logical operations at the F outputs.
- ✓ The two function – select S_1, S_0 specify the particular arithmetic or logical operation to be generated.
- ✓ With three variables selection variables it is possible to specify four arithmetic operation (with S_2 in one state) and a four logic operations(S_2 in other state).
- ✓ The input output carries have meaning only during an arithmetic operation
- ✓ In this way there are 8 arithmetic operations are generated .

Design of Arithmetic Circuit operation

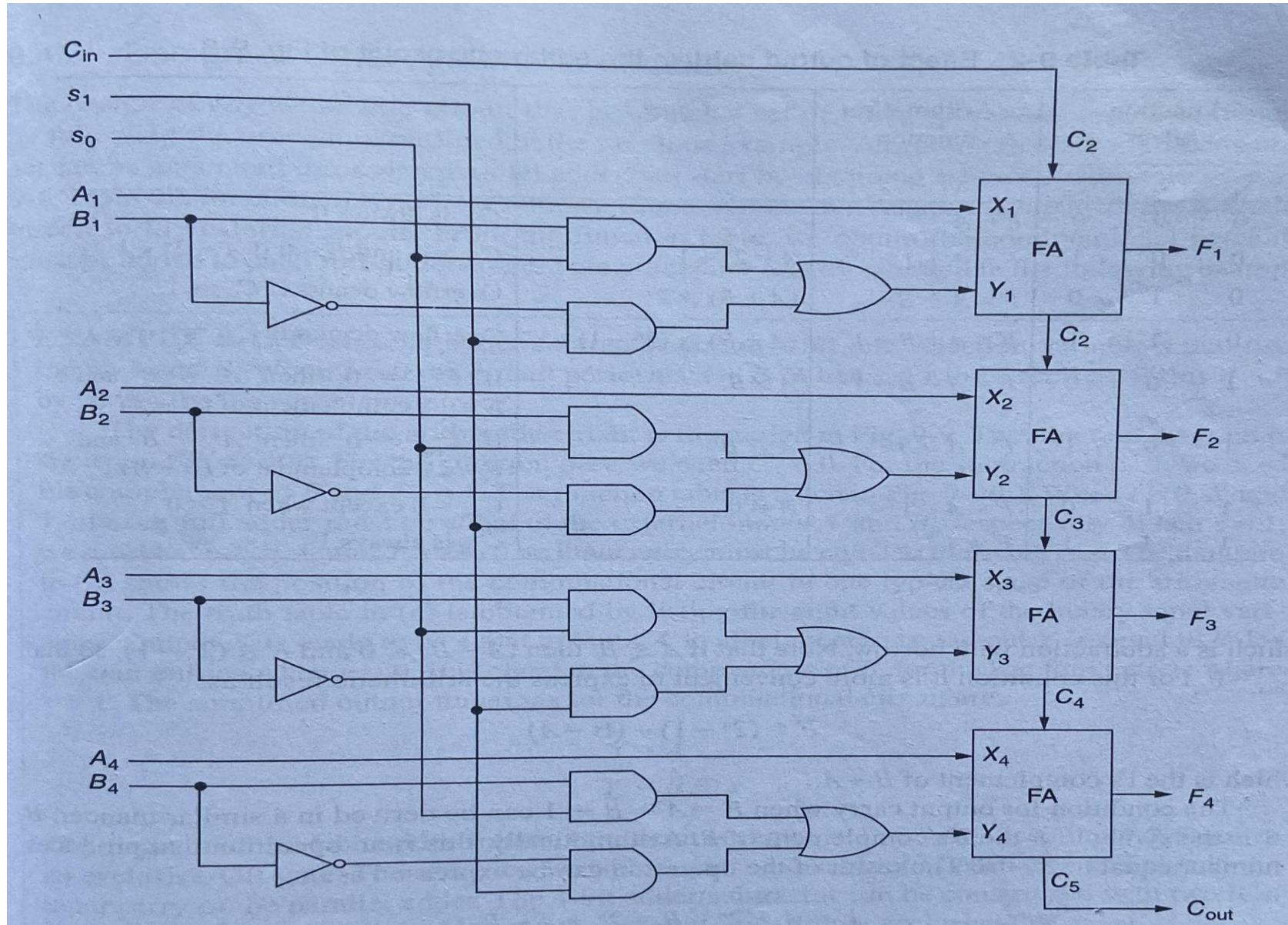
- ✓ The following figure demonstrate the 8 arithmetic operations



Function Table of 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	\bar{B}	$F = A + \bar{B}$	Add 1's complement of B to A
1	0	1	\bar{B}	$F = A + \bar{B} + 1$	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

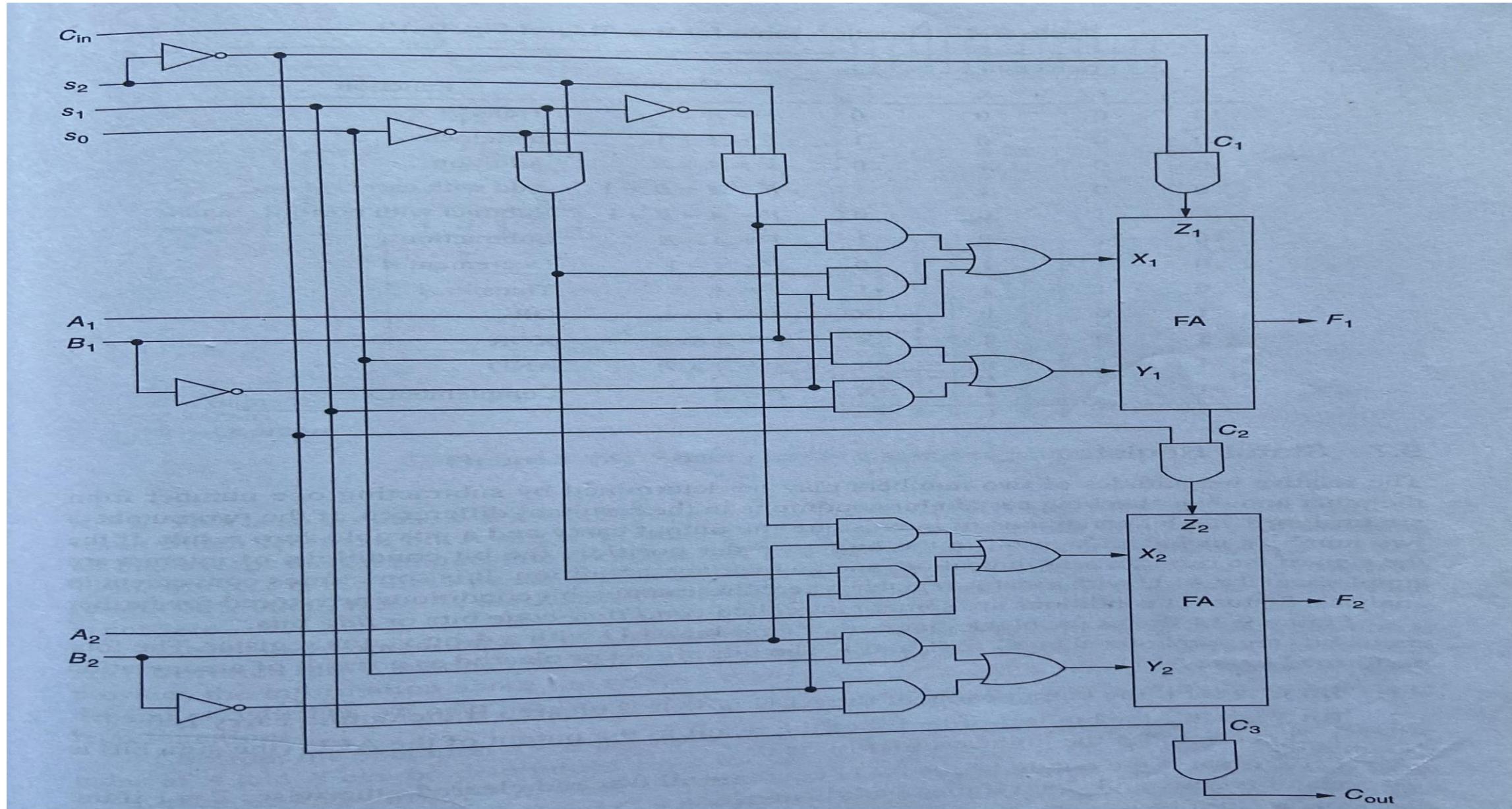
Logic Diagram of Arithmetic Circuit



Functional Table of ALU

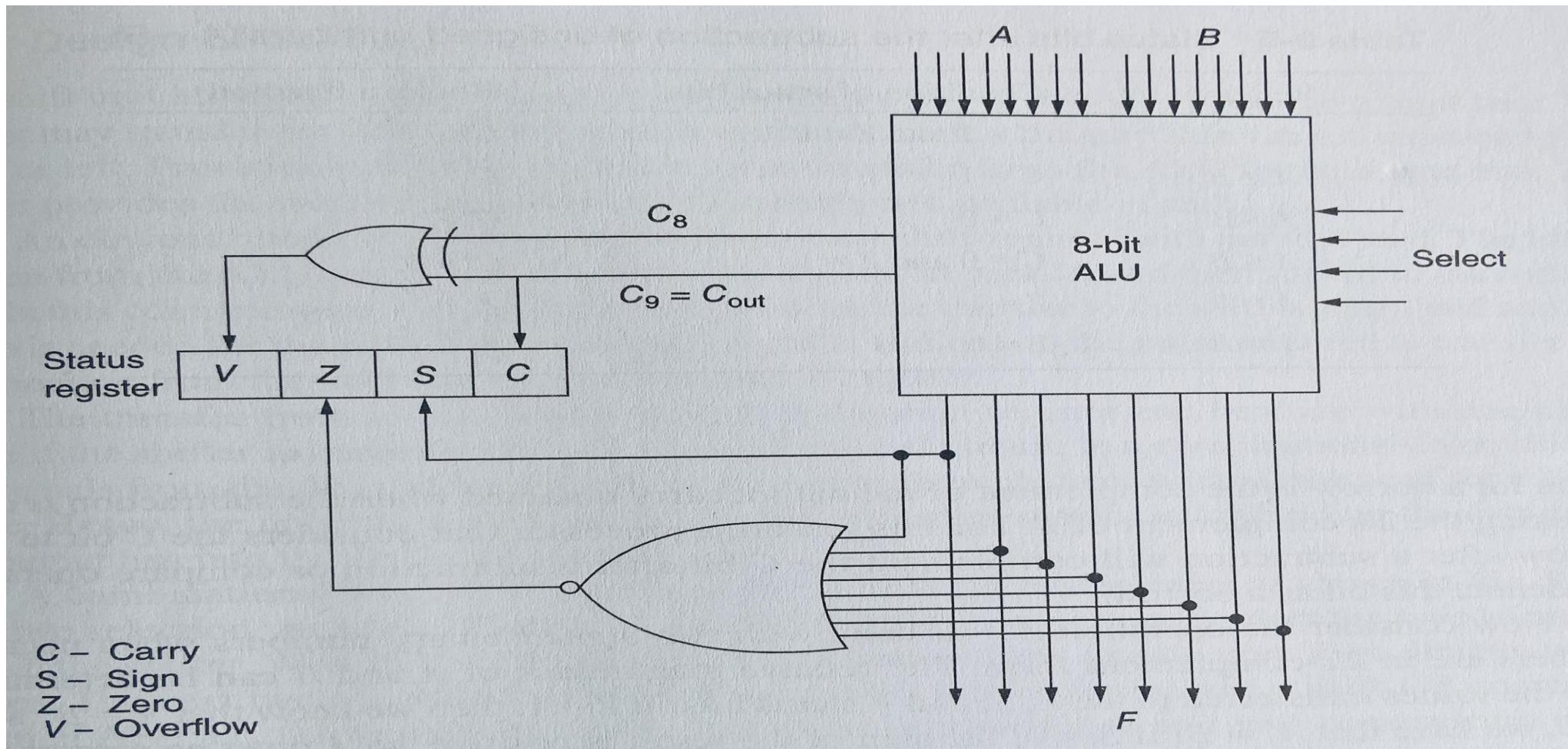
Selection				Output	Function
s_2	s_1	s_0	C_{in}		
0	0	0	0	$F = A$	Transfer A
0	0	0	1	$F = A + 1$	Increment A
0	0	1	0	$F = A + B$	Addition
0	0	1	1	$F = A + B + 1$	Add with carry
0	1	0	0	$F = A - B - 1$	Subtract with borrow
0	1	0	1	$F = A - B$	Subtraction
0	1	1	0	$F = A - 1$	Decrement A
0	1	1	1	$F = A$	Transfer A
1	0	0	X	$F = A \vee B$	OR
1	0	1	X	$F = A \oplus B$	XOR
1	1	0	X	$F = A \wedge B$	AND
1	1	1	X	$F = \bar{A}$	Complement A

Logic Diagram of ALU



Status Register

- ✓ The Status Register reflects the results obtained in the ALU.
- ✓ They are stored in the status register for further analysis.
- ✓ Following figure shows the 8-bit ALU with 4-bit Status Register ,where C is carry bit, S is sign bit Z is Zero bit and V is Overflow bit.



Working of Status Register

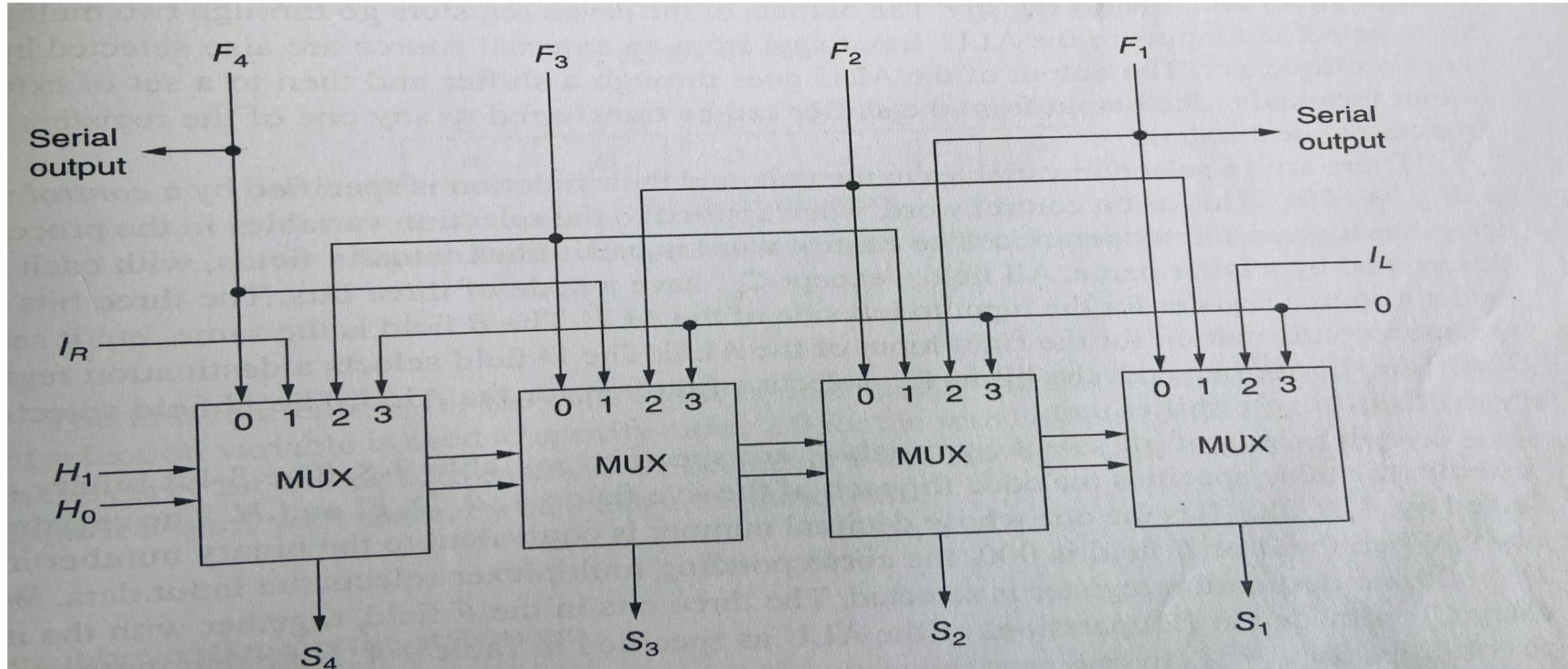
1. Bit C is set if the output carry of the ALU is 1. It is cleared if the output carry is 0.
2. Bit S is set if the highest-order bit of the result in the output of the ALU (the sign bit) is 1. It is cleared if the highest-order bit is 0.
3. Bit Z is set if the output of the ALU contains all 0's, and cleared otherwise. $Z = 1$ if the result is zero, and $Z = 0$ if the result is nonzero.
4. Bit V is set if the exclusive-OR of carries C_8 and C_9 is 1, and cleared otherwise. This is the condition for overflow when the numbers are in sign-2's-complement representation (see Section 8-6). For the 8-bit ALU, V is set if the result is greater than 127 or less than -128.

Relation	Condition of status bits	Boolean function
$A > B$	$C = 1$ and $Z = 0$	CZ'
$A \geq B$	$C = 1$	C
$A < B$	$C = 0$	C'
$A \leq B$	$C = 0$ and $Z = 1$	$C' + Z$
$A = B$	$Z = 1$	Z
$A \neq B$	$Z = 0$	Z'

Functional Table of Status Register

Shifter

- ✓ The Shifter are the unit attached to a processor that transfers the output of the ALU to the output bus.
- ✓ They shift the information without a shift or shifting the information right or left.
- ✓ Following diagram shows the block diagram of the 4-bit Combinational –Logic Shifter



Working

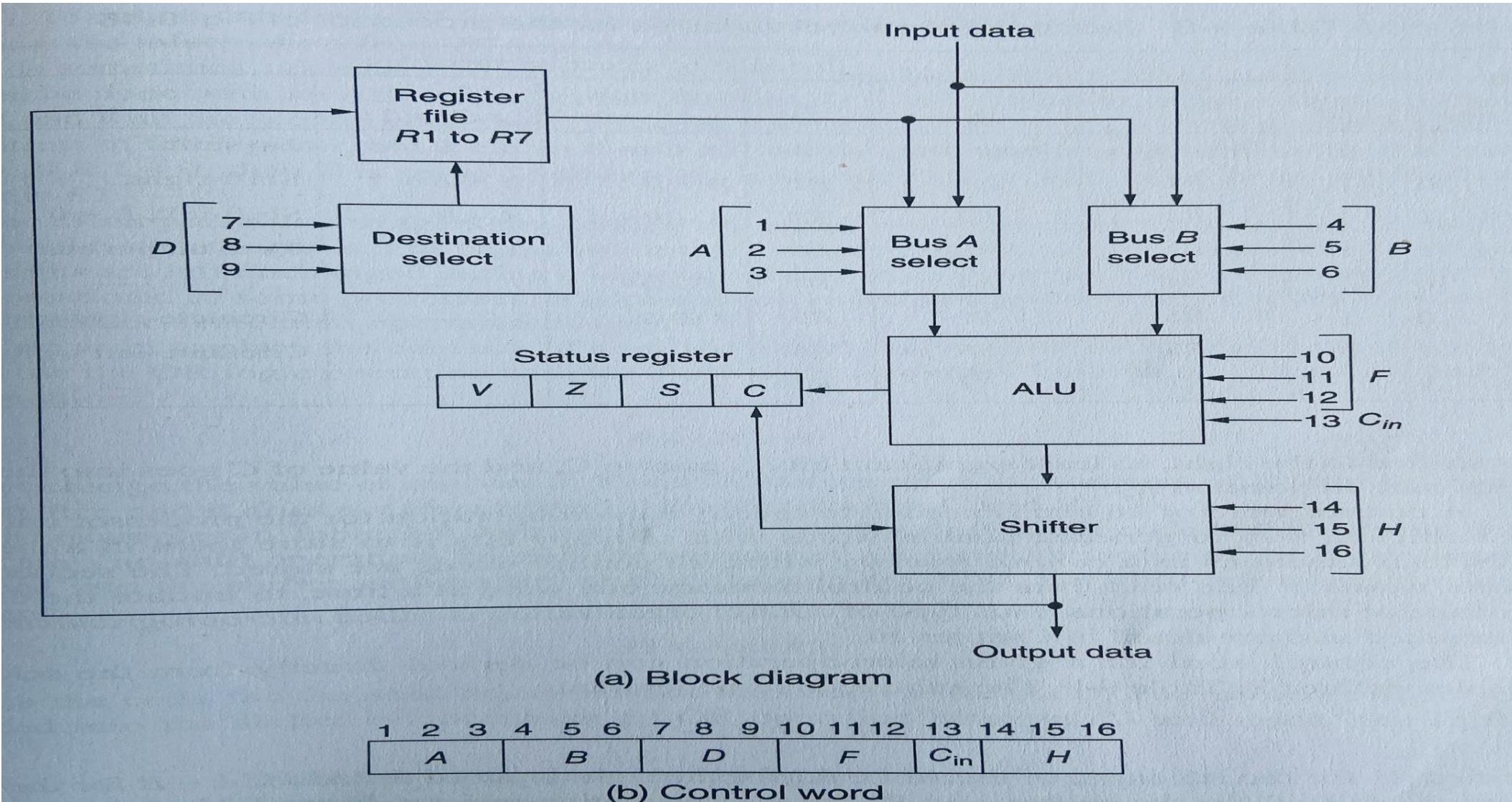
- ✓ A shifter is a bi directional shift register with the parallel load .
- ✓ The information from the ALU is transferred to the register in parallel and then shifted to the right or left.
- ✓ Here only one clock pulse is needed in the processor system for loading the data from the output bus into destination register.
- ❖ A combinational logic shifter can be constructed with the multiplexer as shown above .
- ❖ The Two select variable H1 and H0 , is applied to all four multiplexer and selects the type of operation in the shifter.
- ❖ When H1H0 =00,no shift is executed and the signals from F go directly to the S lines.
- ❖ When H1H0=1,the multiplexer selects the input attached to 0 and as a consequences the S outputs are also equal to 0,blocking the transfer of information from ALU to the output bus.
- ❖ Here IL and IR serves as the serial inputs for the last and first stage during right or left shift.

H_1	H_0	Operation	Function
0	0	$S \leftarrow F$	Transfer F to S (no shift)
0	1	$S \leftarrow \text{shr } F$	Shift-right F into S
I	0	$S \leftarrow \text{shl } F$	Shift-left F into S
1	I	$S \leftarrow 0$	Transfer 0's into S

Functional Table of Shifter

Processor Unit

✓

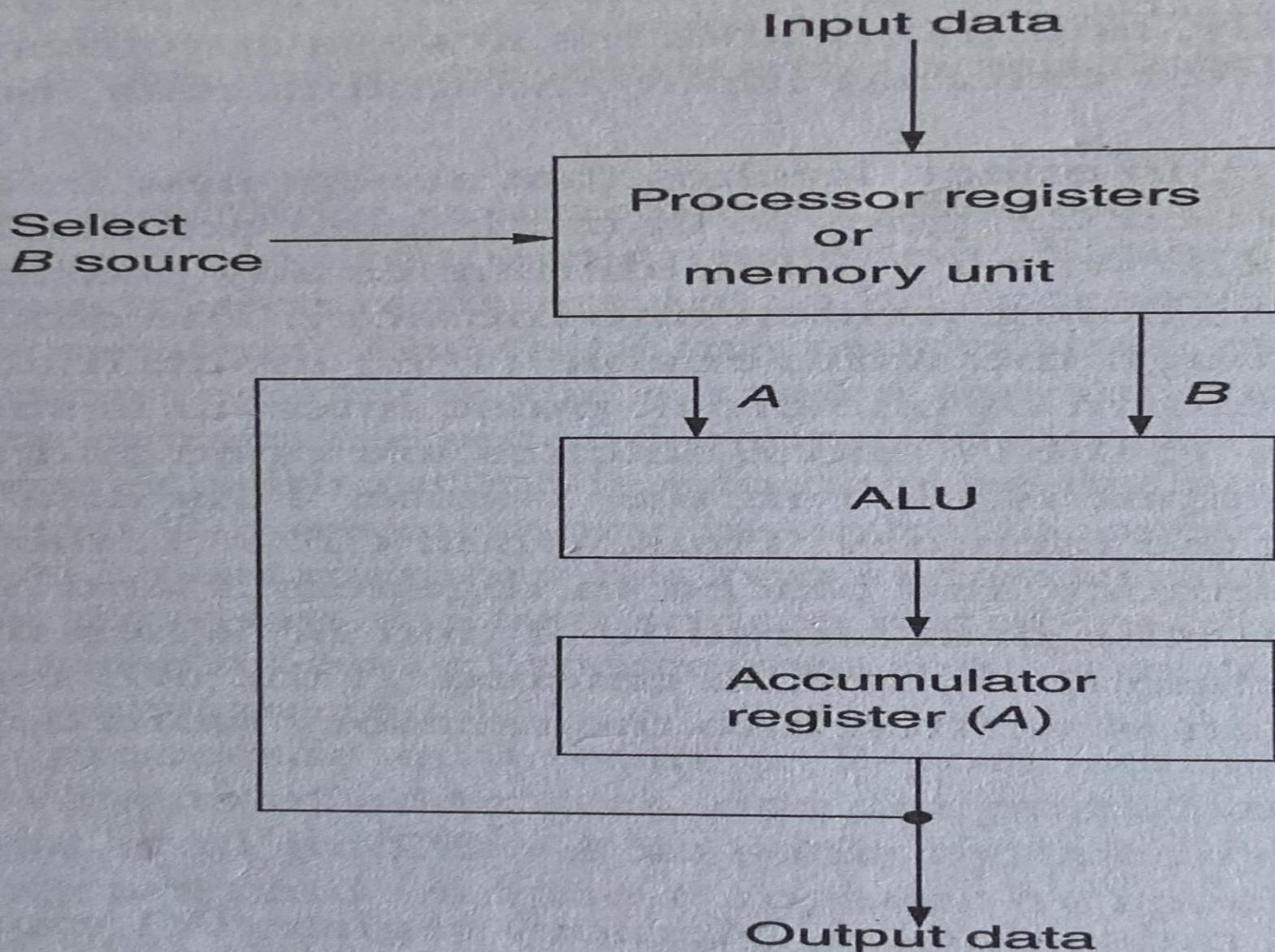


Block Diagram of Processor Unit

Binary code	Function of selection variables					
	A	B	D	F with $C_{in} = 0$	F with $C_{in} = 1$	H
0 0 0	Input data	Input data	None	$A, C \leftarrow 0$	$A + 1$	No shift
0 0 1	R_1	R_1	R_1	$A + B$	$A + B + 1$	Shift-right, $I_R = 0$
0 1 0	R_2	R_2	R_2	$A - B - 1$	$A - B$	Shift-left, $I_L = 0$
0 1 1	R_3	R_3	R_3	$A - 1$	$A, C \leftarrow 1$	0's to output bus
1 0 0	R_4	R_4	R_4	$A \vee B$	—	—
1 0 1	R_5	R_5	R_5	$A \oplus B$	—	Circulate-right with C
1 1 0	R_6	R_6	R_6	$A \wedge B$	—	Circulate-left with C
1 1 1	R_7	R_7	R_7	\bar{A}	—	—

Functional Table of Processor Unit

Accumulator Register



Accumulator Register

- ✓ The accumulator register is another type of central processing unit register that is widely used for storing the logic or intermediate results. It is denoted or represented by "A" or "AC"
- ✓ The accumulator register has a very important role as if it is not there then all the intermediate results need to be stored in the main memory that can increase the overhead on the memory.
- ✓ It is because then unnecessary read and writes operations will be increased.
- ✓ The accumulator register can easily store the intermediate results.
- ✓ The accessing speed of the accumulator register is much faster compared to the main memory.
- ✓ In many modern systems, there are various types of accumulators that can be used to store the intermediate results.
- ✓ However, more the use of accumulators more complex will be the design.
- ✓ For example, in the operation "3 + 4 + 5," the accumulator would hold the value 3, then the value 7, then the value 12.
- ✓ The benefit of an accumulator is that it does not need to be explicitly referenced, which conserves data in the operation statement.

Microoperations at Accumulator Register:

1. $A \leftarrow 0$ i.e. A is cleared first
2. $A \leftarrow A + R1$ i.e. Transfer R1 to A
3. $A \leftarrow A + R2$ i.e. Add R2 to A

RAM

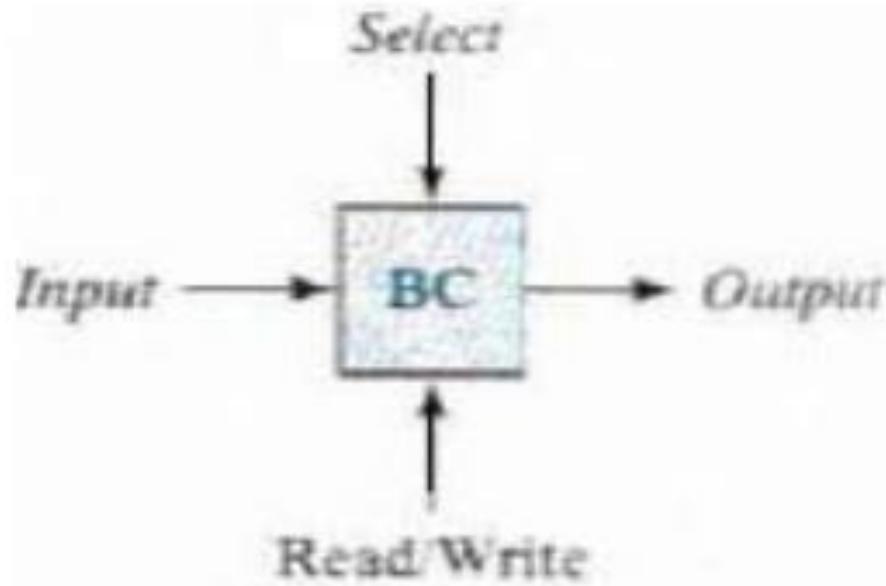


Fig: Block diagram of a memory cell

- ✓ When the Read/Write Input value is 0 then the write mode is selected and when Read/Write Input value is 1 then Read mode is selected

RAM

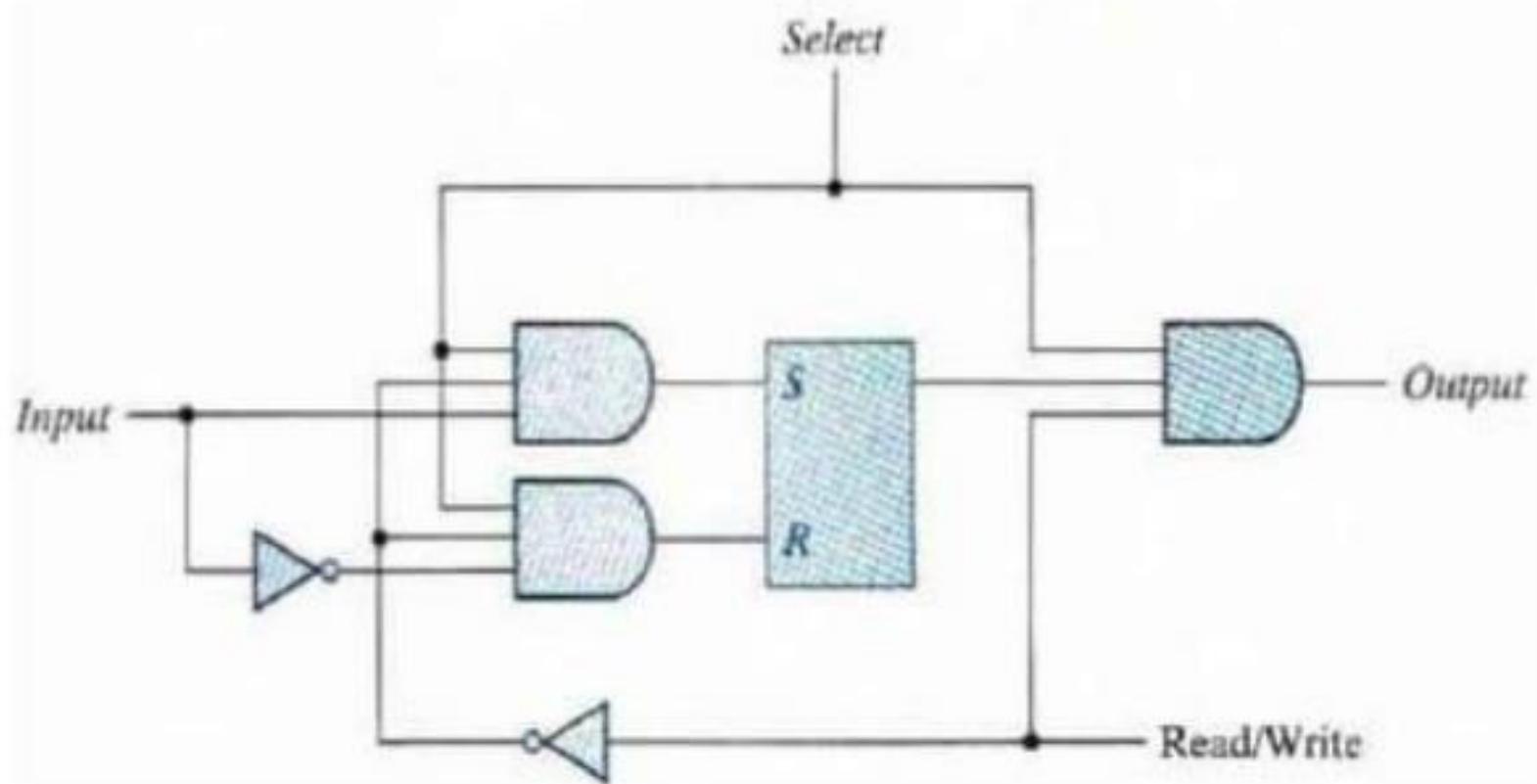


Fig: Logic diagram of a memory cell

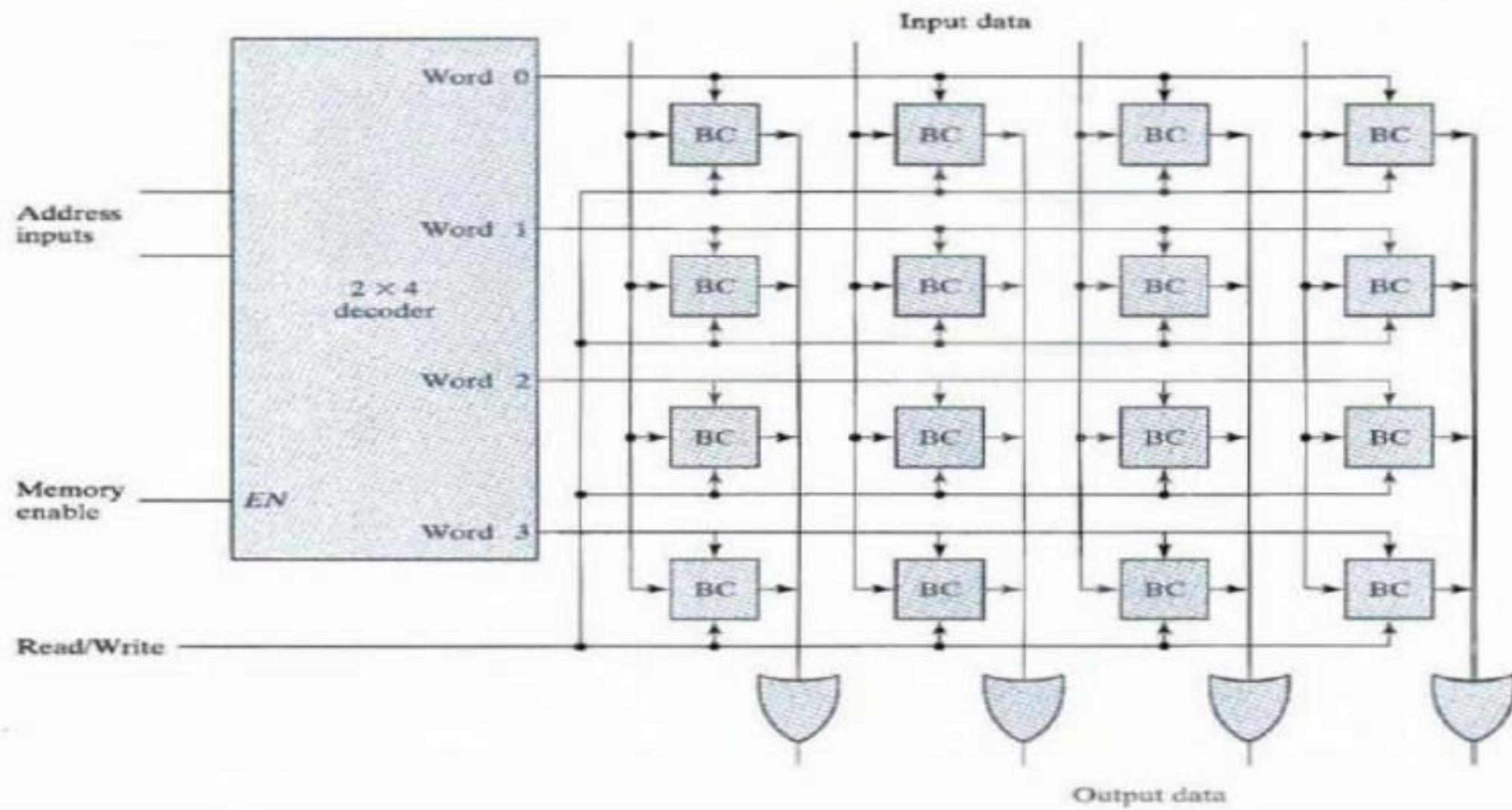


Fig: Diagram of a 4×4 RAM

END of UNIT 9