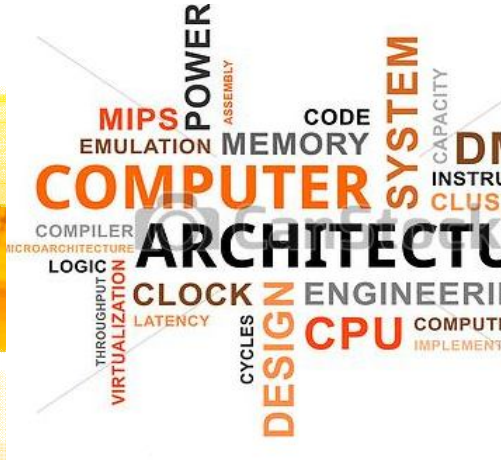
UNIT II
ARITHMETIC OPERATIONS

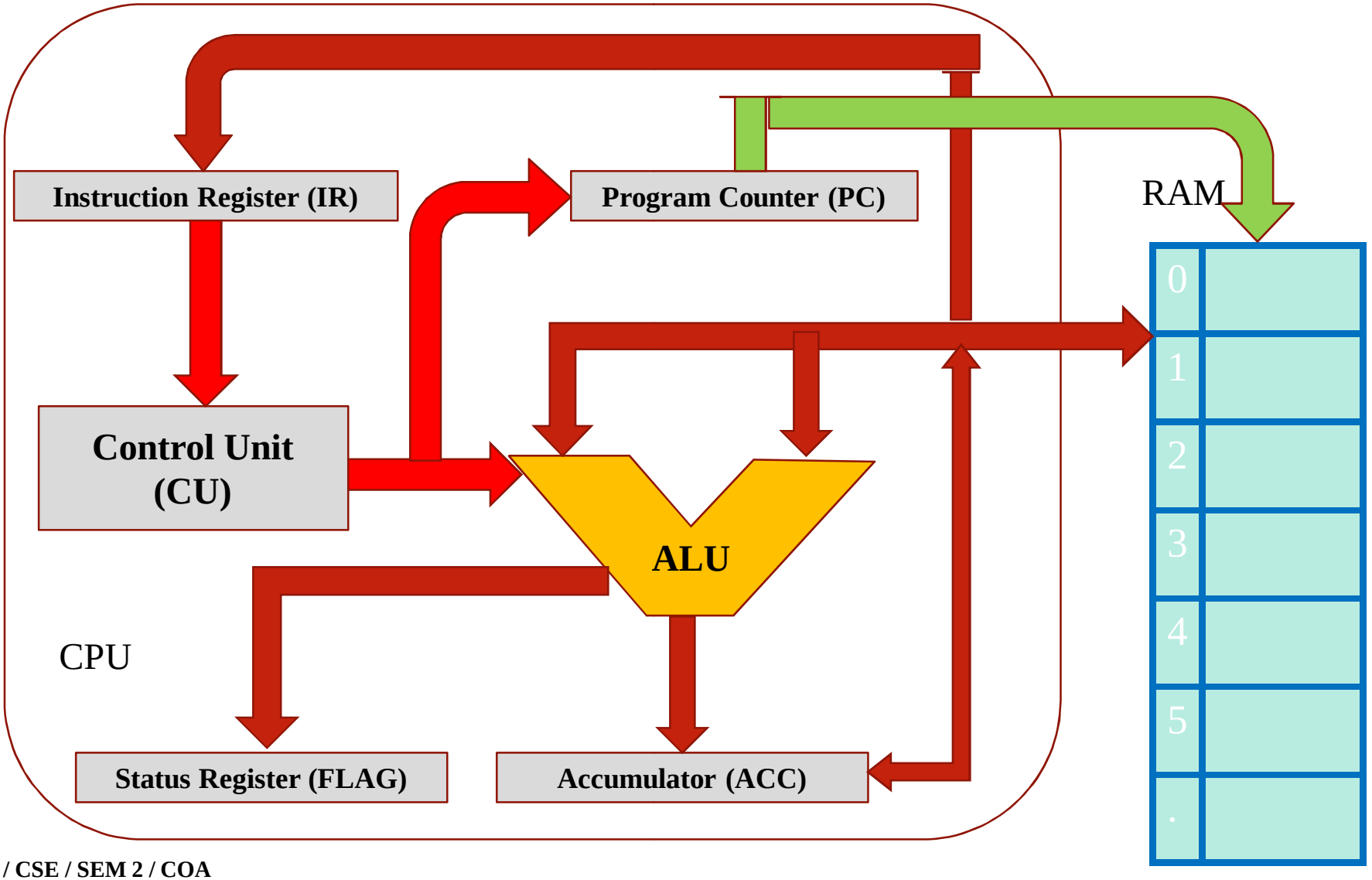
Addition and subtraction of signed numbers – Design of fast adders – Multiplication of positive numbers - Signed operand multiplication- fast multiplication – Integer division – Floating point numbers and operations



Recall Unit I

- Functional units
- Basic operational concepts
- Bus Structures
- Performance
- Memory locations and addresses
- Memory operations
- Instruction and Instruction sequencing
- Addressing modes
- Assembly language

Recall Unit I

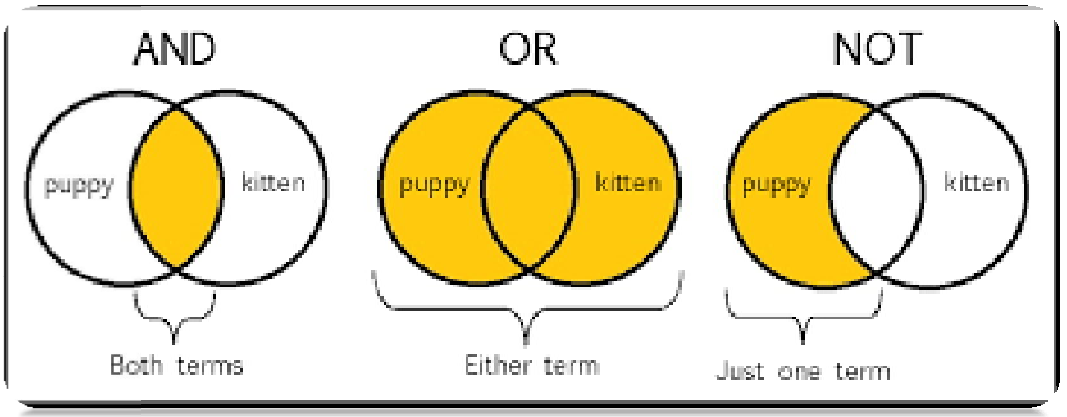


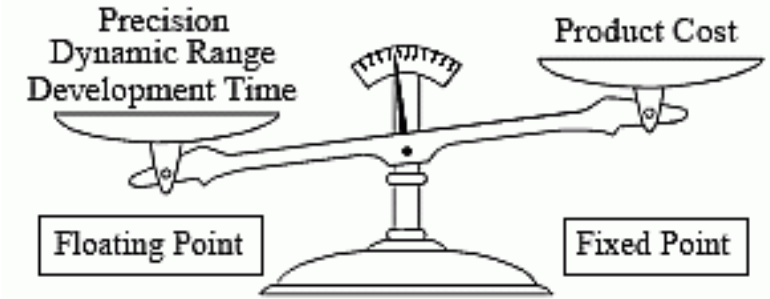
Introduction

Basic Arithmetic Operation

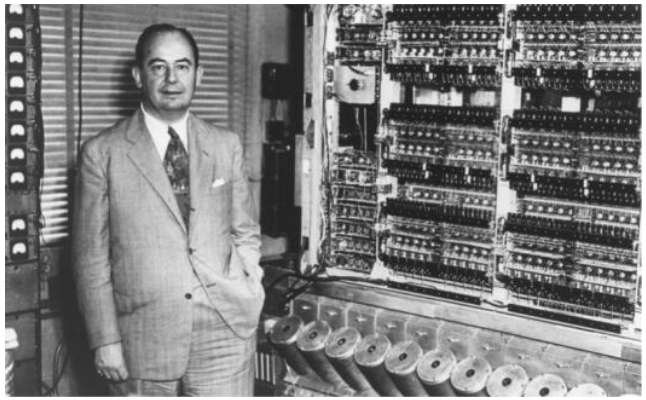
Representation

Logical Operation





What's ALU?

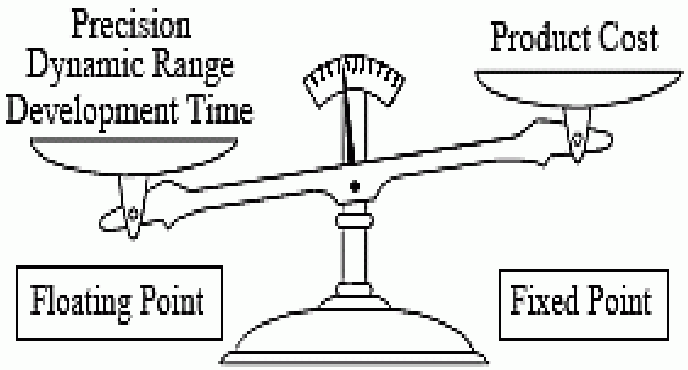


- Stands for Arithmetic and Logic Unit
- Performs Arithmetic (Add, Sub, . . .) and Logical (AND, OR, NOT) operations.
- John Von Neumann proposed the ALU in 1945 when he was working on EDVAC (**E**lectronic **D**iscrete **V**ariable **A**utomatic **C**omputer)

Arithmetic and Logical Unit

Circuit Design

Operations



Combinational Logic Circuit

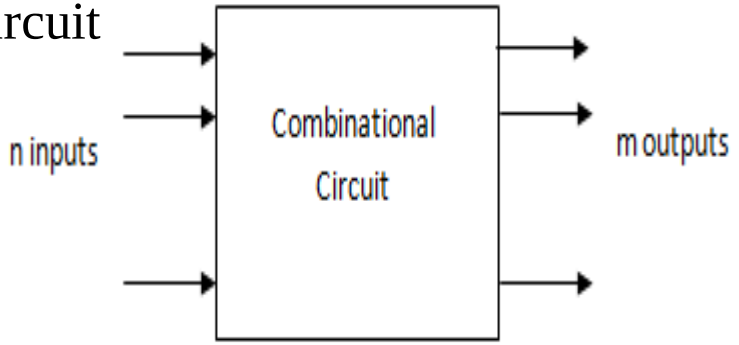
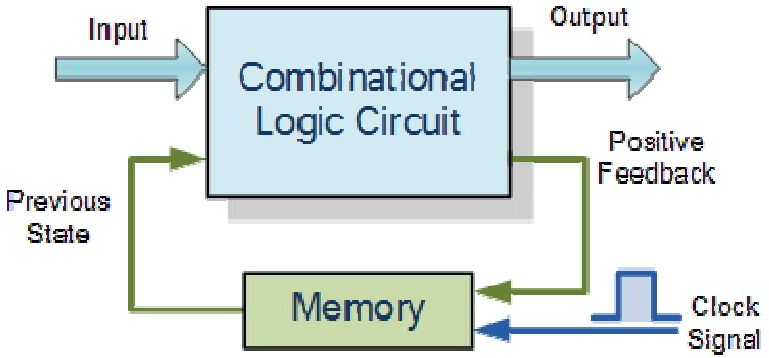
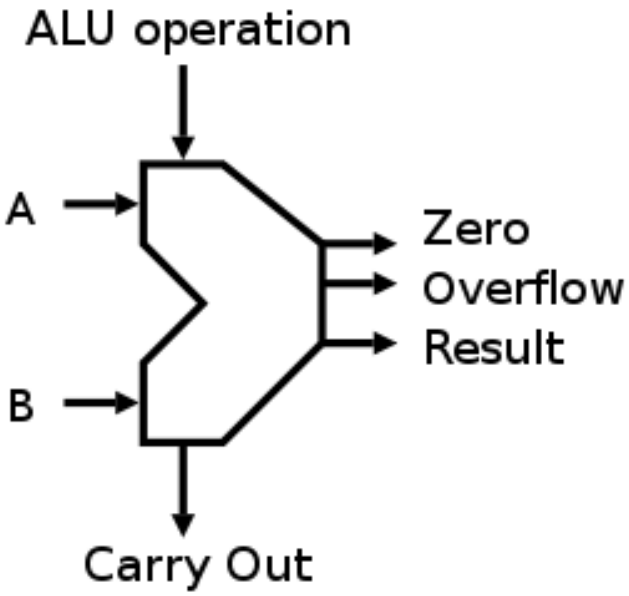


Diagram-1

Sequential Logic Circuit



Typical Schematic Symbol of an ALU



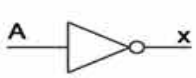
Basic Hardware Components

AND Gates , OR Gates, Inverters & Multiplexers

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

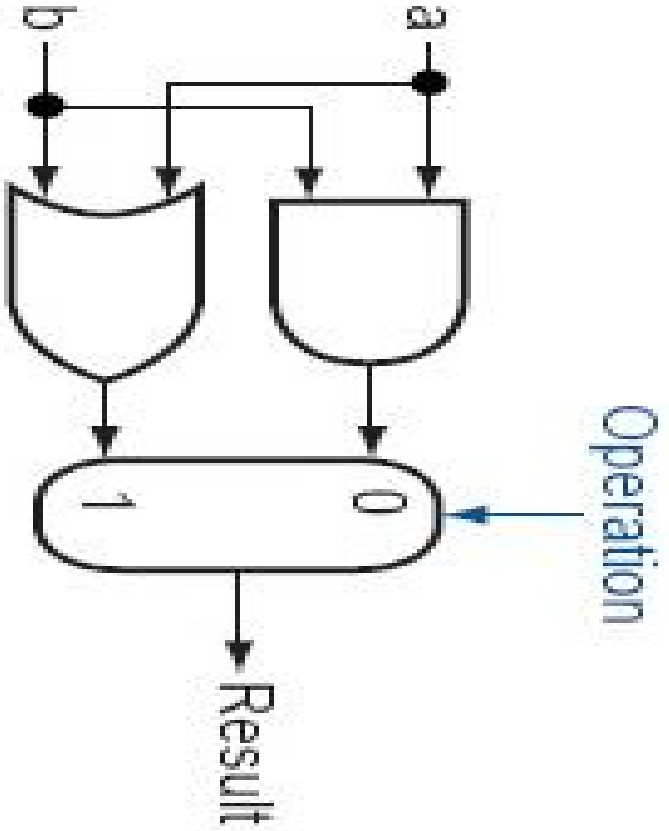
1 Bit ALU

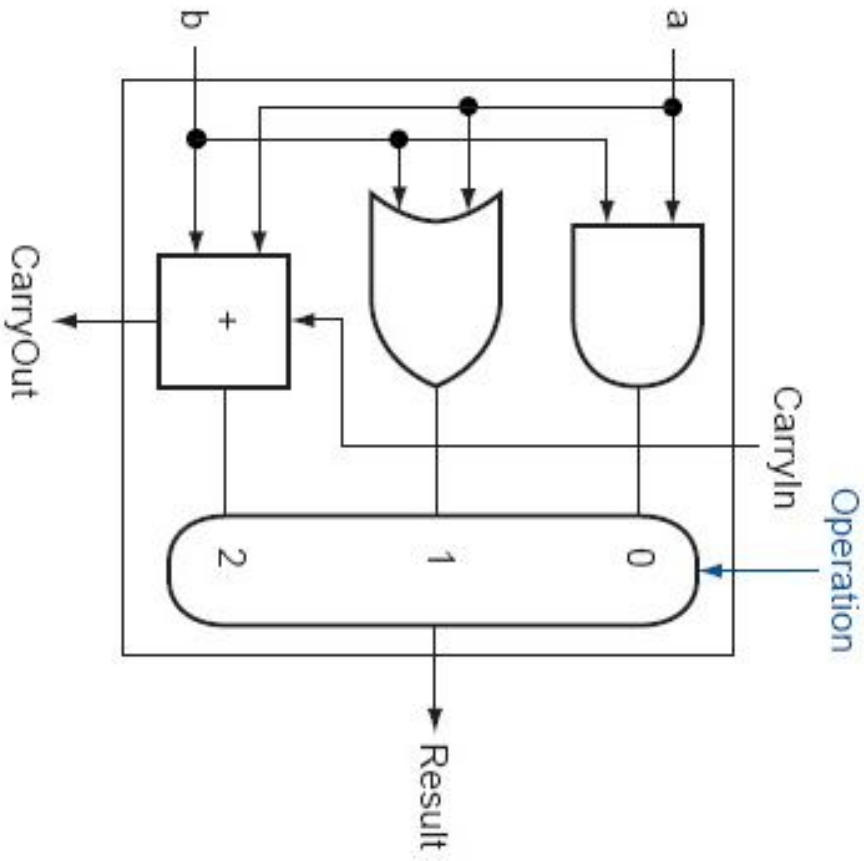
Logic Gates

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$
Symbol							
Truth Table	A X 0 1 1 0	B A X 0 0 0 0 1 0 1 0 0 1 1 1	B A X 0 0 1 0 1 1 1 0 1 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 1	B A X 0 0 1 0 1 0 1 0 0 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 0	B A X 0 0 1 0 1 0 1 0 0 1 1 1

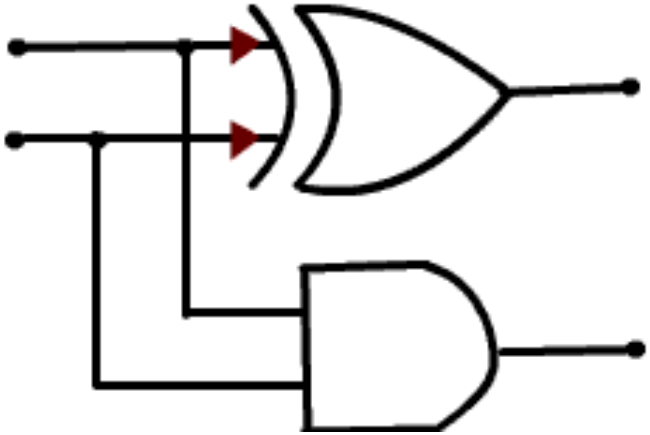
1 Bit ALU

Data line and control Line



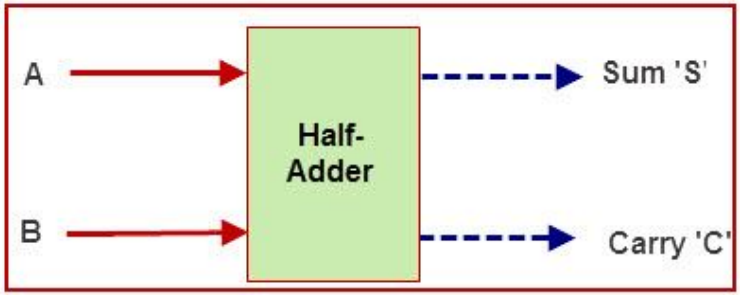


Logic specification for a stage of Binary Addition



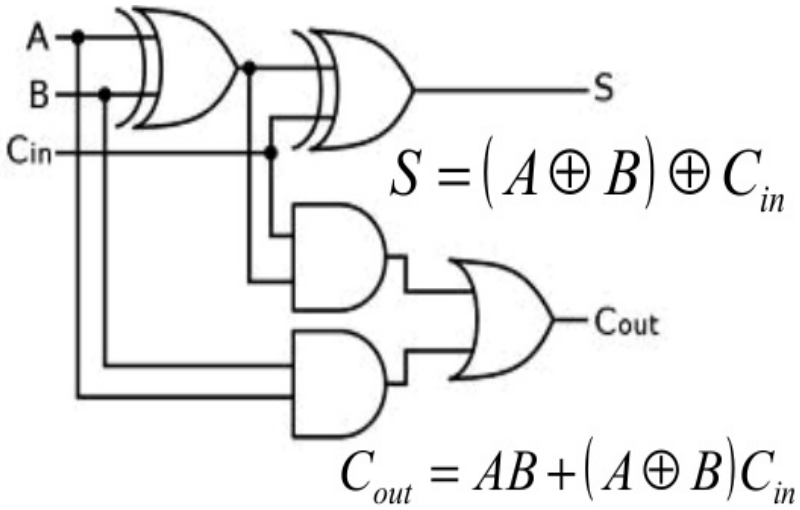
$S = A \oplus B$

$C = A \cdot B$



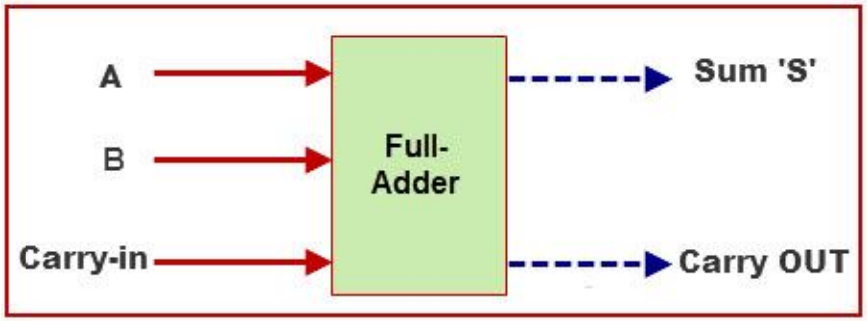
INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Logic specification for a stage of Binary Addition



$$S = (A \oplus B) \oplus C_{in}$$

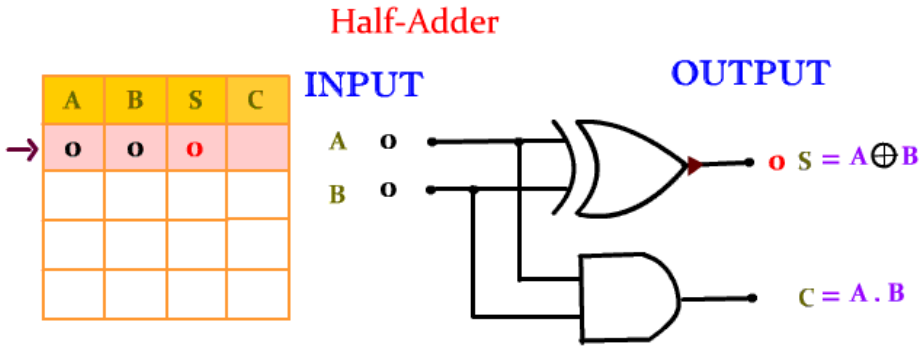
$$C_{out} = AB + (A \oplus B)C_{in}$$

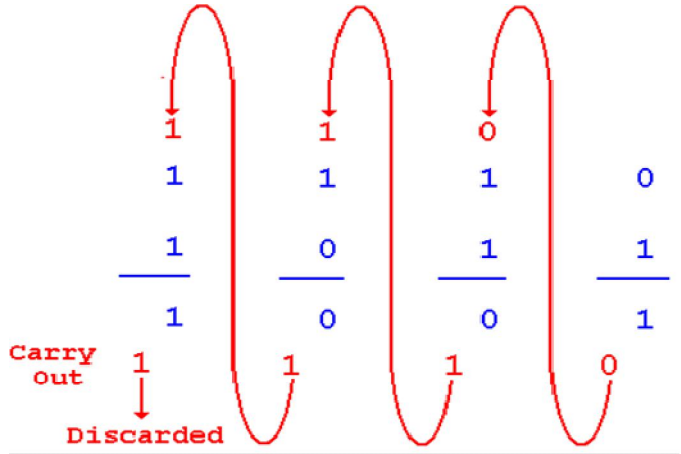


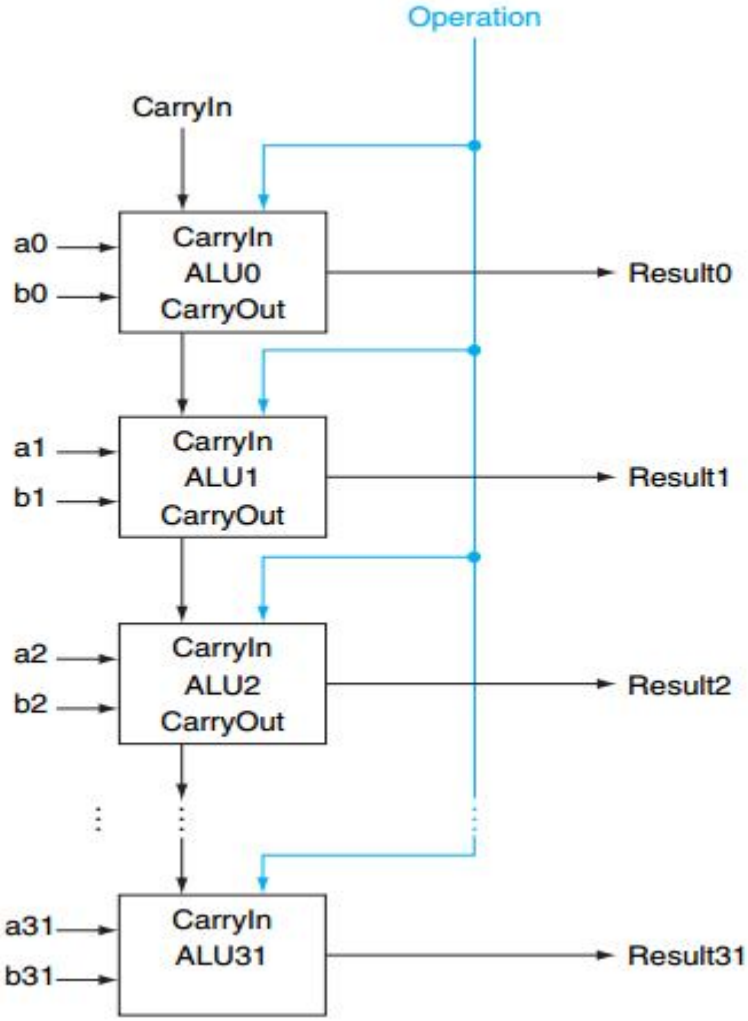
INPUTS			OUTPUT	
A	B	C-IN	C-OUT	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

EXECUTION & OPERATION INSIDE PROCESSOR

32 BIT ALU

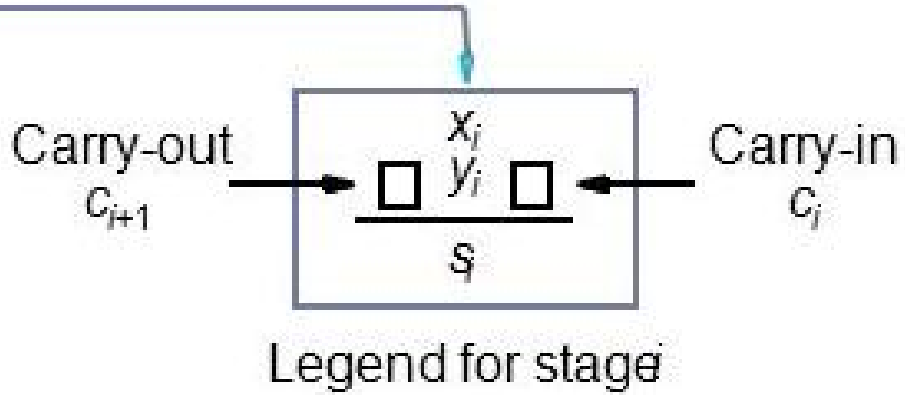






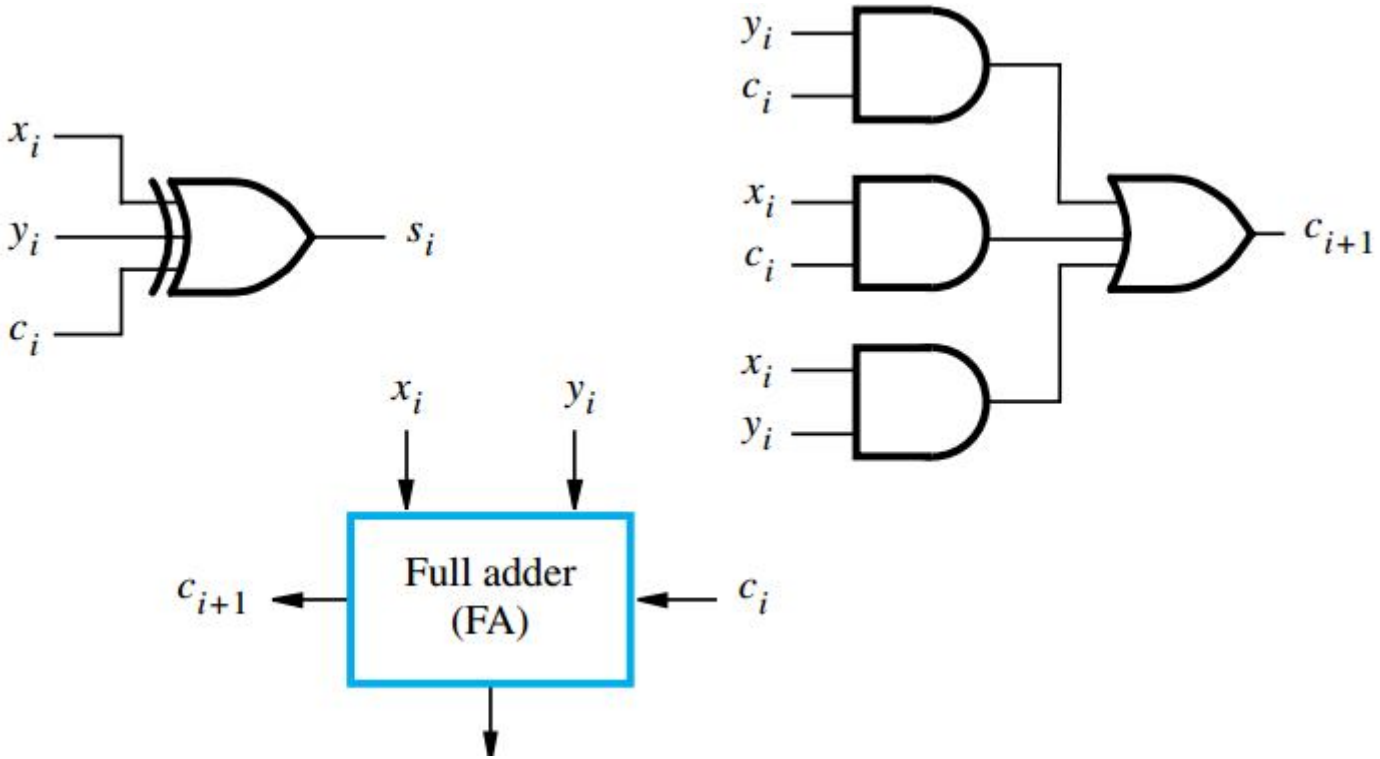
Example for Binary Addition

$$\begin{array}{rcrcr} X & & 7 & & 0\ 1\ 1\ 1 \\ +\ Y & = & +\ 6 & = & +\ 0\ 1\ 1\ 0\ 0 \\ \hline Z & & 13 & & 1\ 1\ 0\ 1 \end{array}$$

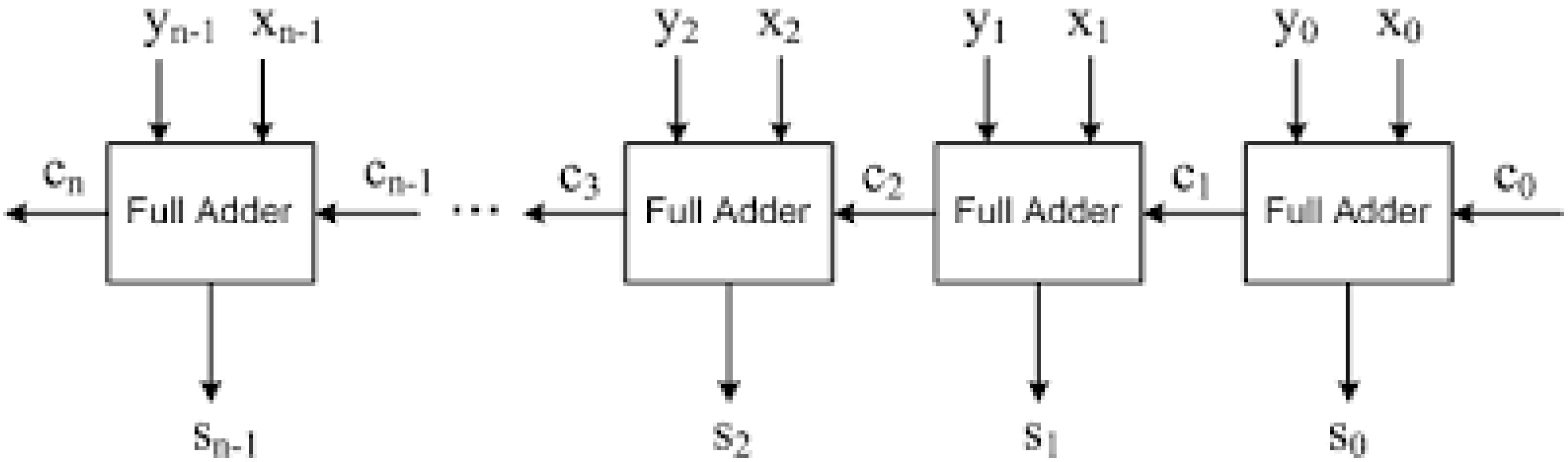


Addition and Subtraction Logic Unit

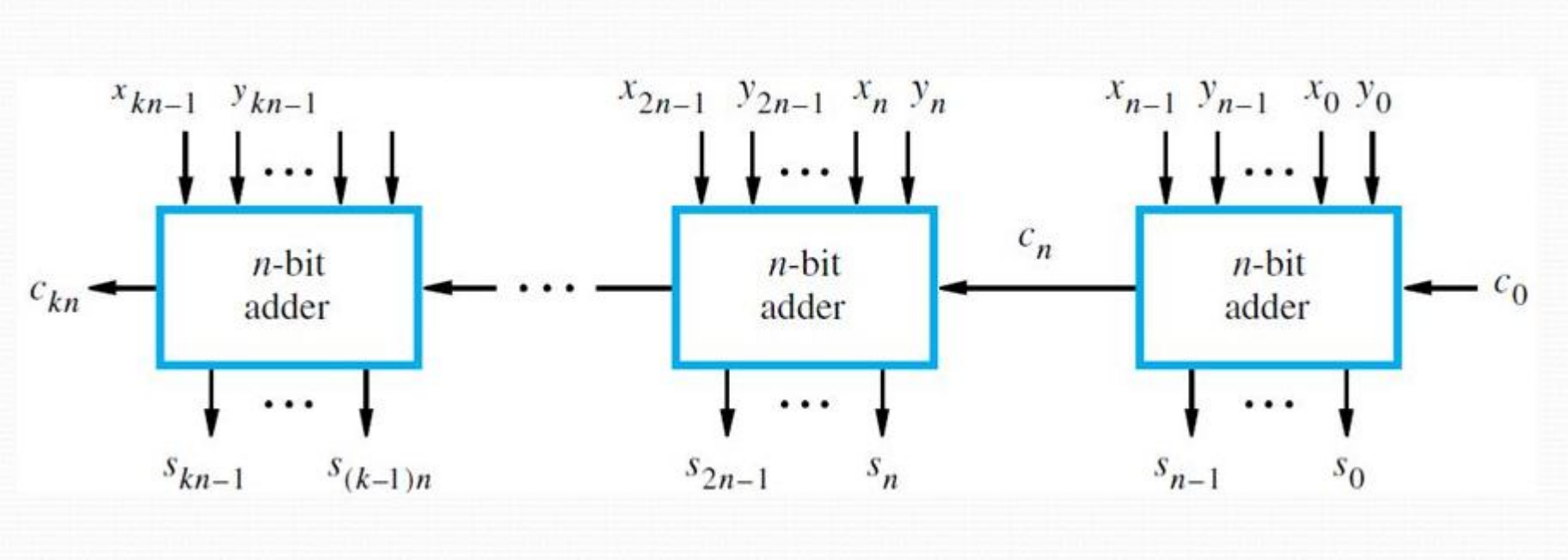
Logic for Single Stage



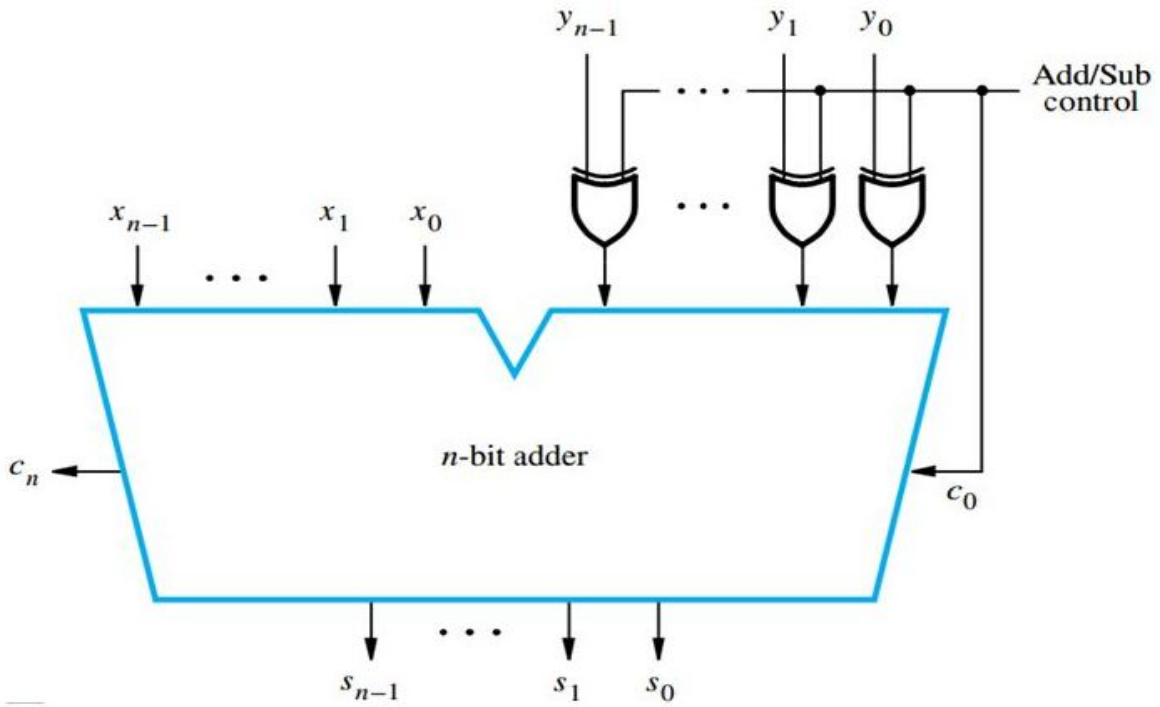
n bit ripple carry adder



cascaded k n bit adders



Binary addition and subtraction logic network



- Addition ➔ Add/sub control = 0.
- Subtraction ➔ Add/sub control = 1

Assessment

OR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

AND

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Neg-AND

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

NOR

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

NAND

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

Neg-OR

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

XOR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

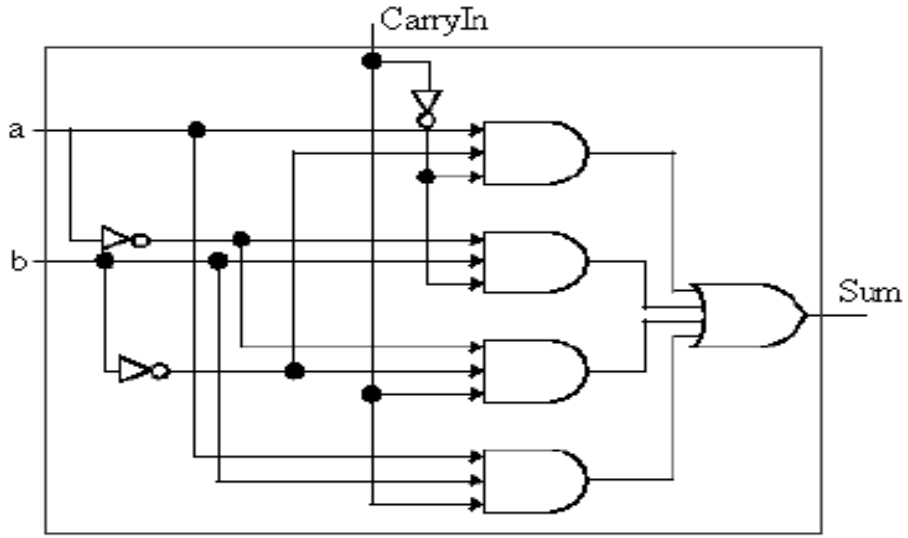
XNOR

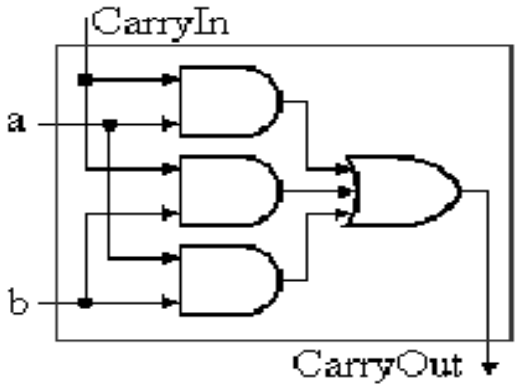
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

NOT

A	Output
0	1
1	0

Assessment





Carryout = (b.CarryIn)+(a.CarryIn) +(a.b)

Sum = (a.b'.CarryIn')+ (a'.b.CarryIn')+ (a'.b'.CarryIn)+ (a.b.CarryIn)

TEXT BOOK

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, McGraw-Hill, 6th Edition 2012.

REFERENCES

1. David A. Patterson and John L. Hennessey, “Computer organization and design”, MorganKauffman ,Elsevier, 5th edition, 2014.
2. William Stallings, “Computer Organization and Architecture designing for Performance”, Pearson Education 8th Edition, 2010
3. John P.Hayes, “Computer Architecture and Organization”, McGraw Hill, 3rd Edition, 2002
4. M. Morris R. Mano “Computer System Architecture” 3rd Edition 2007
5. David A. Patterson “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann; 5th edition 2011

THANK YOU