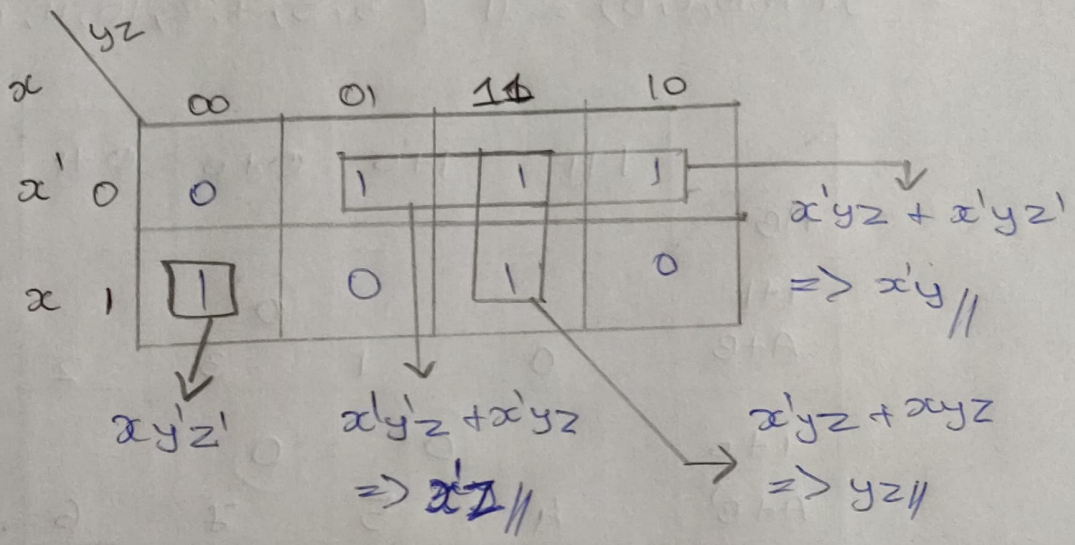
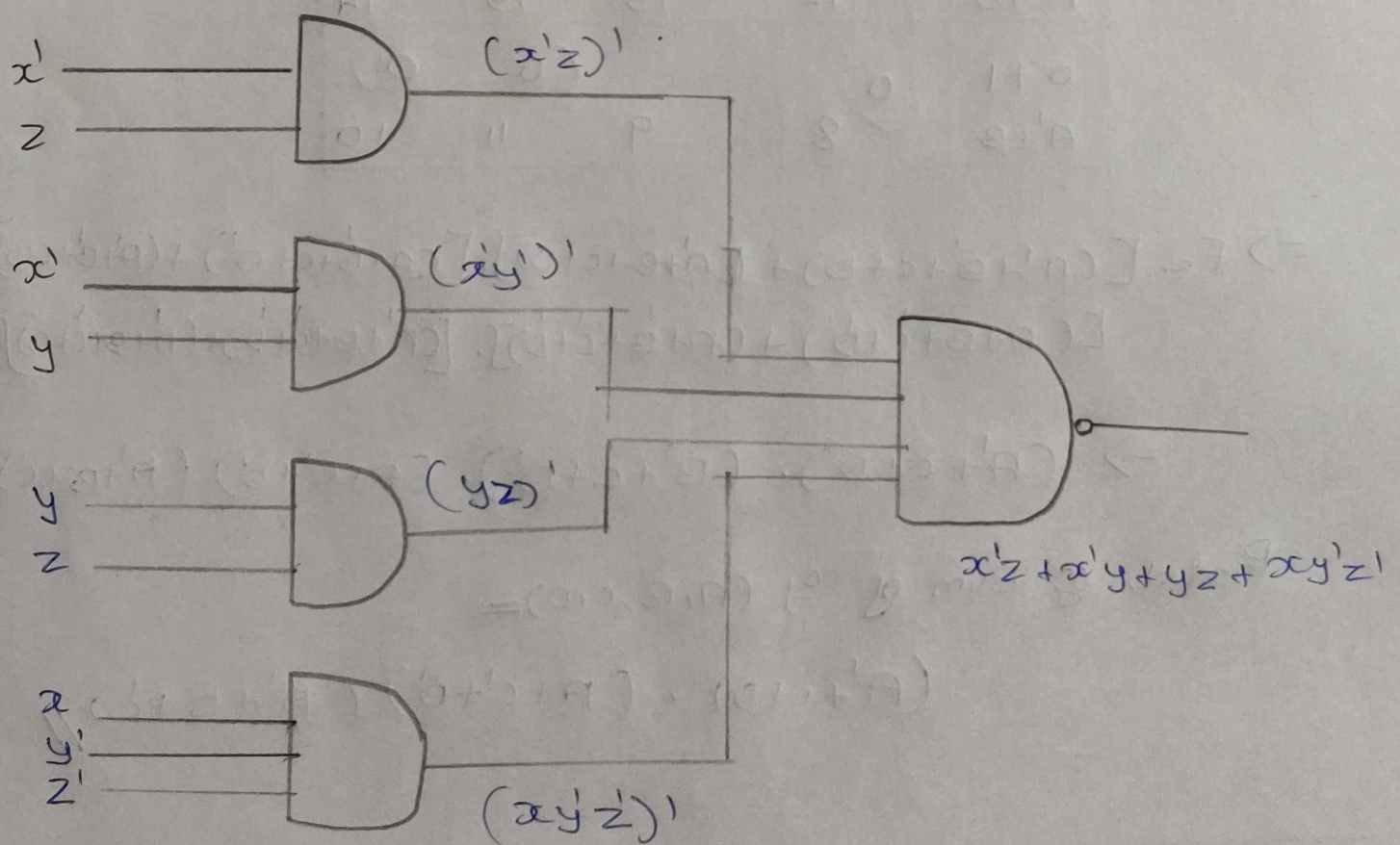
1. IMPLEMENT BOOLEAN FUNCTION:

$(x, y, z) = f\,[1, 2, 3, 4, 7]$ with NAND GATES.

$$y(x,y,z) = \Sigma(1,2,3,4,7)$$





2. Simplify THE FOLLOWING BOOLEAN FUNCTION USING K MAP method. $F(A,B,C,D) = \Pi(3,5,7,8,10,11,12,13)$

$\Rightarrow y(A,B,C,D) = \Pi(3,5,7,8,10,11,12,13)$

AB	1+1	1+0	0+0	0+1
1+1 A+B	0	1	0 3	2
1+0 A+B'	4	0 5	0 7	6
0+0 A'+B'	0 12	0 13	15	14
0+1 A'+B	0 8	9	0 11	0 10

$\Rightarrow F = [(A'+B'+C+D)+(A'+B+C+D)].[(A+B'+C+D')+(A'+B'+C+D)]$
$[(A+B+C'+D')+(A+B'+C'+D)].[(A'+B+C'+D')+(A'+B+C'+D)]$

$\Rightarrow (A'+C+D).(B'+C+D').(A+C'+D').(A'+B+C')$

Product of Sum of $y(A,B,C,D) =$

$(A'+C+D).(A+C'+D')(A'+B+C')$ //

3. A) Convert D-Flipflop into a Jk-Filp Flop :-

→ Avaliable FF = DFF

Required FF = Jk FF

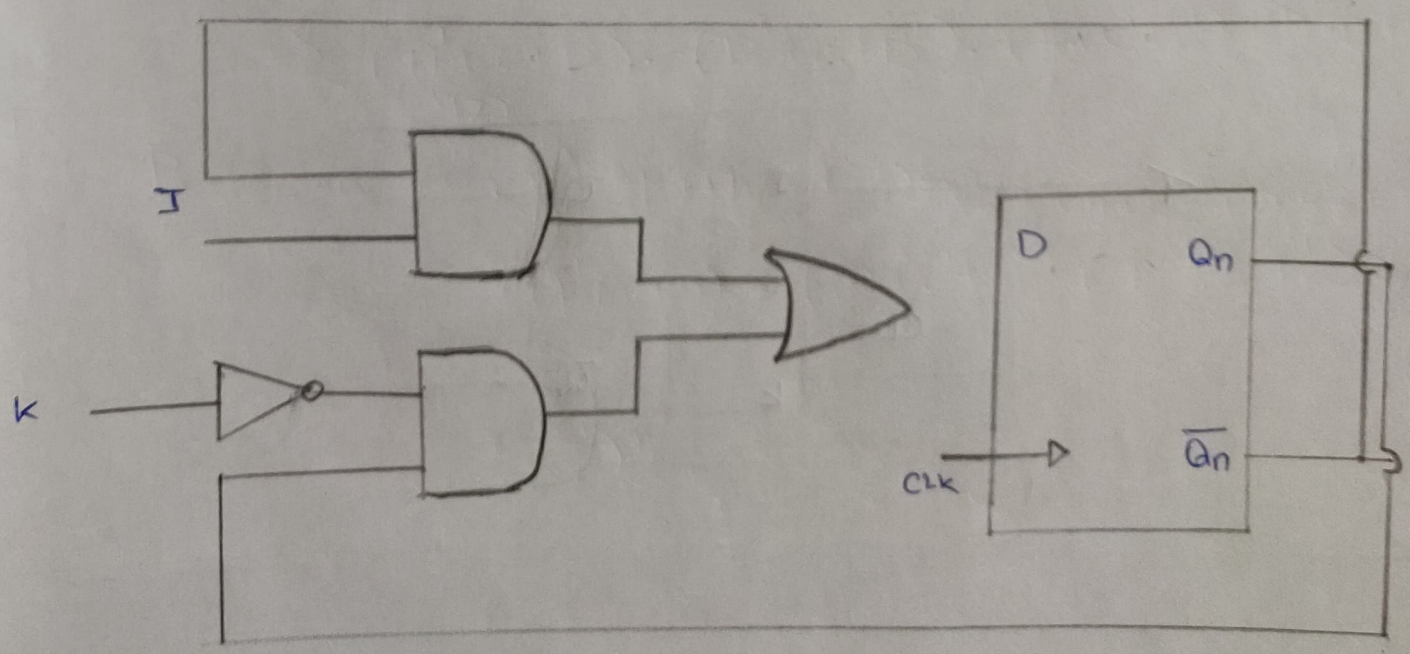
Qn	J	k	Qn+1	D
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Qn	Qn+1	D
0	0	0
0	1	1
1	0	0
0	1	1

D

Qn \ Jk	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$D = Qn\bar{K} + \bar{Q}nJ$



B) CONVERT Jk - Flip Flop into T-Flip Flop :-

- Available = Jk FlipFlop
Required = T FlipFlop

Qn	T	Qn+1	J	k
0	0	0	0	x
0	1	1	1	x
1	0	1	x	0
1	1	0	x	1

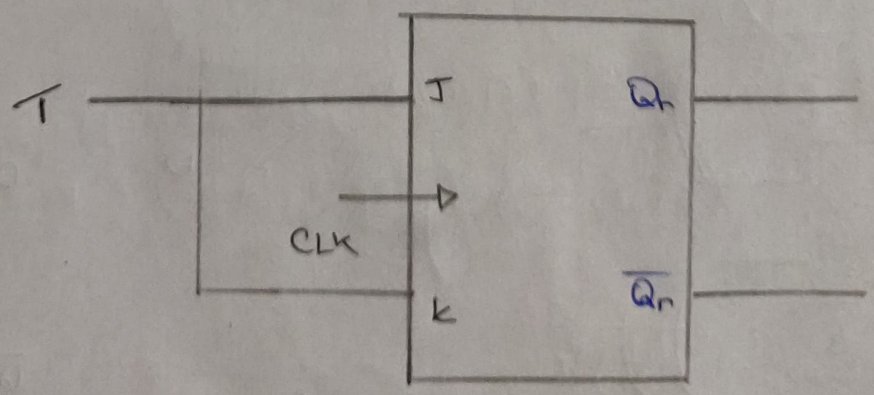
Qn	Qn+1	J	k
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

J	Qn \ T	0	1
	0	0	1
	1	x	x

J = T

K	Qn \ T	0	1
	0	x	x
	1	0	1

K = T



4. EXPLAIN THE VARIOUS INSTRUCTIONS FORMATS WITH ILLUSTRATIVE EXAMPLE.

A Computer performs a task based on the instruction provided. Instructions in computer comprises groups called fields. These fields contain different information as for computers every thing is in 0 and 1. So each field has different significance based on which a cpu decides what to perform.

The Most Common Fields are:-

* Operation field Specifies the operation to be Performed like addition.

* Address field which contains the location of the operand i.e Register or Memory location.

* Mode field which Specifies how operand is to be founded.

Instruction is variable length depending upon the number of address it contains, Generally, cpu organization is of three types Based on the number of address fields.

* Single Accumlator organization
* General register organization
* Stack organization.

* In the first organisation, the operation is done involving a special register called the accumlator. In Second or multiple registers are used for the Computation purpose.

* only a single organisation doesn't need to be applied. a blend of various organization is mostly what we see generally.

* Based on the number of address instruction are classified as:

* Note that we will use $X = (A+B) * (C+D)$ Exp to showcase the procedure.

1. Zero Address Instruction:

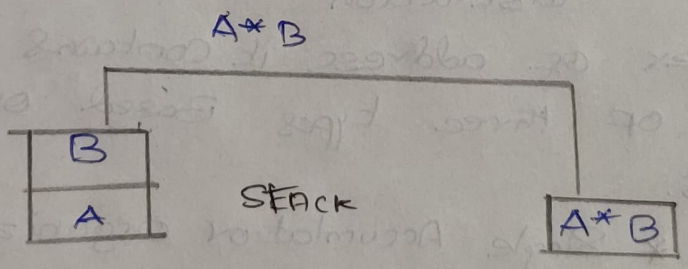
A Stack-Based Computer doesn't not use the instruction. To evaluate an expression first it is converted to reverse polish notation i.e: Postfix Notation.

Expression : $X = (A+B) * (C+D)$

Postfixed : $X = AB+CD+*$

Top means top of Stack

M[x] is any memory location.



PUSH A
PUSH B

PUSH	A	Top = A
PUSH	B	Top = B
ADD		Top = A+B
PUSH	C	Top = C
PUSH	D	Top = D
ADD		Top = C+D
MUL		Top = (C+D) * (A+B)
POP	X	M[X] = TOP

2. ONE ORDER INSTRUCTION:

This uses an Implied Accumulator register for data manipulation. One register for data Manipulation. one operand is the accumulator and the other is in the register or memory location. Implied means that the CPU already knows that one operand is in the accumulator so there is no need to specify it.

OPCODE	Operand address of operand	Mode.

Expression : $x = (A+B) * (C+D)$

AC is accumulator

M[ ] is any memory location.

M[T] is Temporary location.

Load	A	AC = M[A]
Add	B	AC = AC + M[B]
STORE	T	M[T] = AC
LOAD	C	AC = M[C]
Add	D	AC = AC + M[D]
Mul	T	AC = AC * M[T]
STORE	x	M[x] = AC.

3. Two Address Instructions:

This is common in commercial Computers. Here two address can be Specified in the instruction. Unlike earlier in one address instructions, the results was stored in the accumulator, here the result Can be Stored at different location rather than just accumulator, But Reauire more number of Bit to Represent address.

Opcode	Destination Address	Source Address	Mode.

Here, Destination address can be Containg operand also.

Expression: $X = (A+B) * (C+D)$

R_1, R_2 are registers

M[] is any memory location.

Mov	R_1, A	$R_1 = M[A]$
ADD	R_1, B	$R_1 = R_1 + M[B]$
Mov	R_2, C	$R_2 = C$
ADD	R_2, D	$R_2 = R_2 + D$
MUL	R_1, R_2	$R_1 = R_1 * R_2$
Mov	X, R_1	$M[X] = R_1$

4. THREE ADDRESS INSTRUCTIONS:

This has three address field to specify a register or a memory location. Program created are much short in size but number of bits per instructions increase. These instruction make creation of program much easier but it does not mean that program will run much faster because now instruction only contain more information but each micro operation [changing content of register, loading address in address bus etc]. will be performed in one cycle only.

Opcode	Destination address	Source Address	Source Address mode.

Expression : $x = (A+B) * (C+D)$

R_1, R_2 are registers

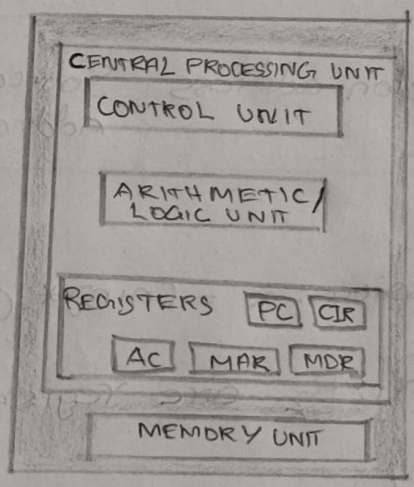
M[] is any memory location.

ADD	R_1, A, B	$R_1 = M[A] + M[B]$
ADD	R_2, C, D	$R_2 = M[C] + M[D]$
MUL	X, R_1, R_2	$M[X] = R_1 * R_2$

5. VON-NEUMANN ARCHITECTURE WITH A NEAT DIAGRAM:

VON-NEUMANN architecture was first Published By John VON NEUMANN in 1945. His computer Architecture Design consists of a Control unit, Arithmetic and Logic unit [ALU], Memory unit, Registers and INPUT/OUTPUTS.

* Von-Neumann Architecture is Based on the Stored program data are stored in the same memory. This design is still used in most Computer produced Today.



* CENTRAL PROCESSING UNIT:

* The Central processing unit is the electronic circuit response for executing the instruction of a computer program.

* The CPU contains the ALU, CU and a variety of registers.

* REGISTERS: Registers are high speed storage areas in the CPU. All data must be stored in a register before it can be processed.

MAR	MEMORY ADDRESS REGISTER	Holds the memory location of Data that needs to be accessed.
MDR	MEMORY DATA REGISTER	Holds data that is being Transfered to or from memory.
Ac	Accumulator	Where intermediate arithmetic and logic results are stored.
PC	PROGRAM COUNTER	Contains the address of the next instruction to be executed.
CIR	CURRENT INSTRUCTIONS REGISTER	Contains the current instruction during Processing.

* ARITHEMETIC AND LOGIC UNIT [ALU]

The ALU allows arithmetic [Add, Subtract, etc] and logic AND, OR, NOT, etc] operation carried out.

* CONTROL UNIT:

The control unit controls the operations of the computer ALU, memory and input/output devices, telling them how to respond to the program instruction it has read and interpretend from the memory unit.

* BUSES :

Buses are the mean by which data is transmitted from one part of a Computer to another connecting all major internal components to the CPU and memory.

Address Bus	Carriers the address of data between the processor and memory.
DATA Bus	Carriers data between the process the memory unit and the input / output Devices.
CONTROL Bus	Carriers control Signals / commands From the CPU in order to control and co-ordinate all the activities the computer.

* MEMORY UNIT:

The memory unit consists of RAM, Sometimes Referred to as Primary or main memory. Unlike a hard drive, This memory is fast and also directly accessible by the CPU.

RAM is split into Partitions, Each Partions consists of an address and its content [both in Binary Form].

The address will identify every location in the memory. Loading data from permanent memory into the faster and directly accesible temporary memory, allows the CPU to operate much quicker.