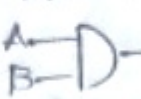
04-11-2024 Digital Electronics.

And Gate

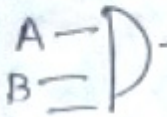
A	B	Output $y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1.

IC: 7408 will do the And operation.



If both voltage is high then result is also high value. If any value 0, Then result is 0.

~~OR function~~. And fn. has various types of ICs Inputs are more than 1.



A	B	C	Y = (A·B·C)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1 → last value is 1.

Any signal down. Then all the value in result = 0.

All pins = 1. then result = 1.

Timing Digram. of Input of And Gate.

Input signals. (Depend on waveform.

Input$_A$ — 0 1 0 0 1 0 1 1 0 0.

Input$_B$ — 1. 1 1 0 1 1 0 0 1 1 1.

0 1 0 0 1 0 0 1 0 0.

if 1 trigger state fire will go through 4 stages of internal t1 t2 t3 t4.

0 1 0 0 1 0 1

1 1 0 1 1 0 0

0 1 0 0 1 0 0

1, 16, 5

1 2 3 4 5

6th 7th 8th 9th 10, 11, 13

14, 16, 17, 19, 20, 21,

22, 23, 24.

18, ✓

OR-Gate

a	b	c	y = a+b+c
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1.

The IC: 7432 is OR gate.

$Y = A + B$. input.

If any signal is high, output signal is high.

If all signal 0 then output low

If any is high output is high

from 1 to 7. all the signals are high.

---

next class: Timing Dig.

Time Dig.

A - 0 1 0 0 1 0 1 1 0 0

B - 1 1 0 1 1 0 0 1 1 1

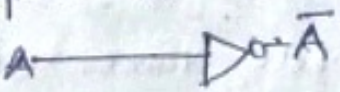
fundamental conv.

Memory.

Logic Gates (AND, OR, NOT) and Timing Digream.

NotGate: 7404.

NOTGATE: Input low then Output high & vic.

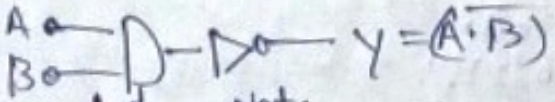


A	$\bar{A}$
0	1
1	0

1 Trigger will cross 2 stedes is Not Gate.

Nand Gate: is NOT and And Gate.

1st And Then NOT.



Ib IC: 7400 is Nand Gate.

Timing D'g

A 0 1 0 0 1 1 0 1

B 1 0 1 1 0 0 1 0

has only 2 stedes T_1 and T_2 stedes.

Input A	Input B	Output Y
0	0	1.
0	1	1.
1	0	1.
1	1	0

1, 2, 3, 4

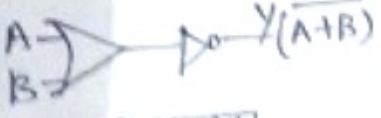
Computer Architecture:

06.11.2024.

NOR gate:

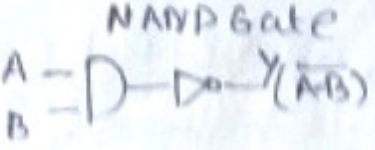
prev. N and Gate

design & not and or



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

NAND Gate



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

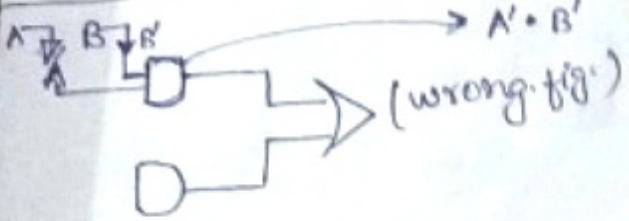
Executive Gate: X-OR.

X-or for 2 or more input

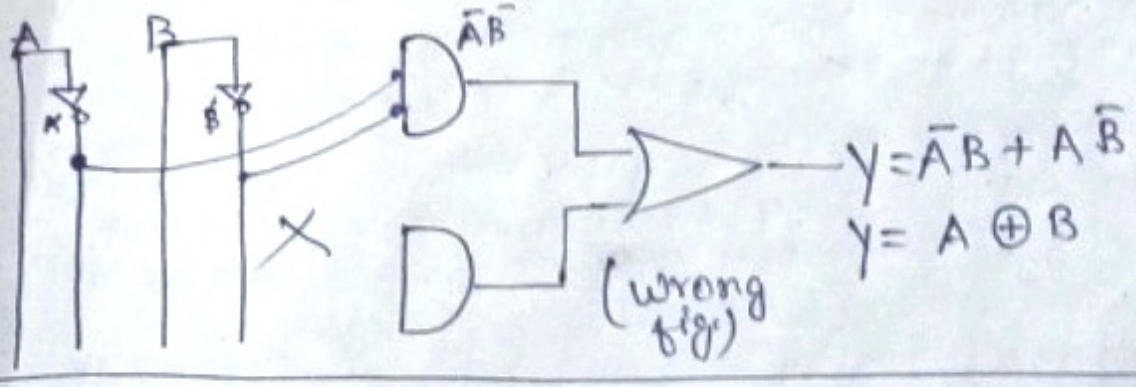
Output

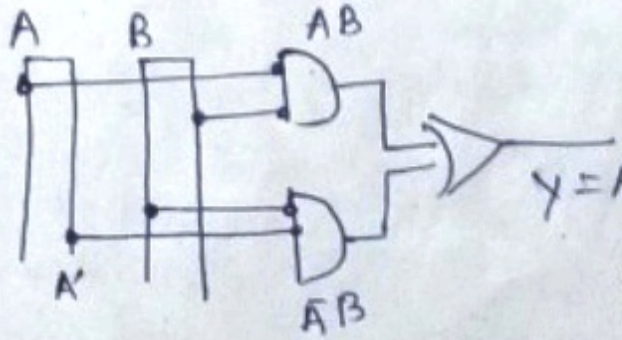
Both 0 then output is 1.

It is combination of 2 gates.



omit.





A+A' = 1
B+B' = 1
} property:

11.2024
Computer Arch. Unit 3.

Data Type: size, range.
General Terminology
Signed : -ve, +ve and 0 -2^{n-1} to $2^n - 1$.
Unsigned: 0 to +ve, 0 to $2^n - 1$.
Where n is no. of bits.
DevCpp Ide : 32bit.

Int (size of integer): 2byte
n = 16.
-2^{15} to $2^{15} - 1$
-32768 to 32767.
Error in Turbo Cpp. (exception).
in Devcpp : size of int : 4byte.

we reconsider the data type width.
double, float, long.
(i) check computer arch.
(ii) check variable size.

Unit - III
CPU Organization
stack org.
Accumulator org.
general purpose register org.

1. Fetch
2. Decode
3. Execute.

Accumulator register store previous value.
Drawback: If we switch/Expression change, we need to copy the value

else the value is ~~over~~ overwritten.

file is saved, close button trigger;
whether you want to save file?

So why we use General register processor used.

Search: (fetch). search. Primary unit
↳ many links (seo) : (decode). decison making.] control unit
↳ Open appropriate link : (execute) finilise output

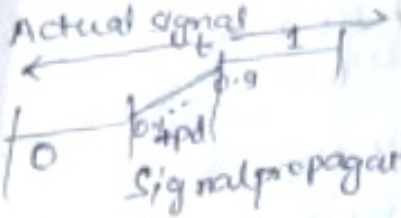
Task: analyse 5 problems resolved through fetch decode and execute operations.

- CPU take input from cache memory
hit - miss ratio : (OS)

various stages of CPU execution (Instruction-execution phases)

(i) fetch
(ii) Decoded
(iii) executed
(iv) storage; after closing file; data store to the same location.

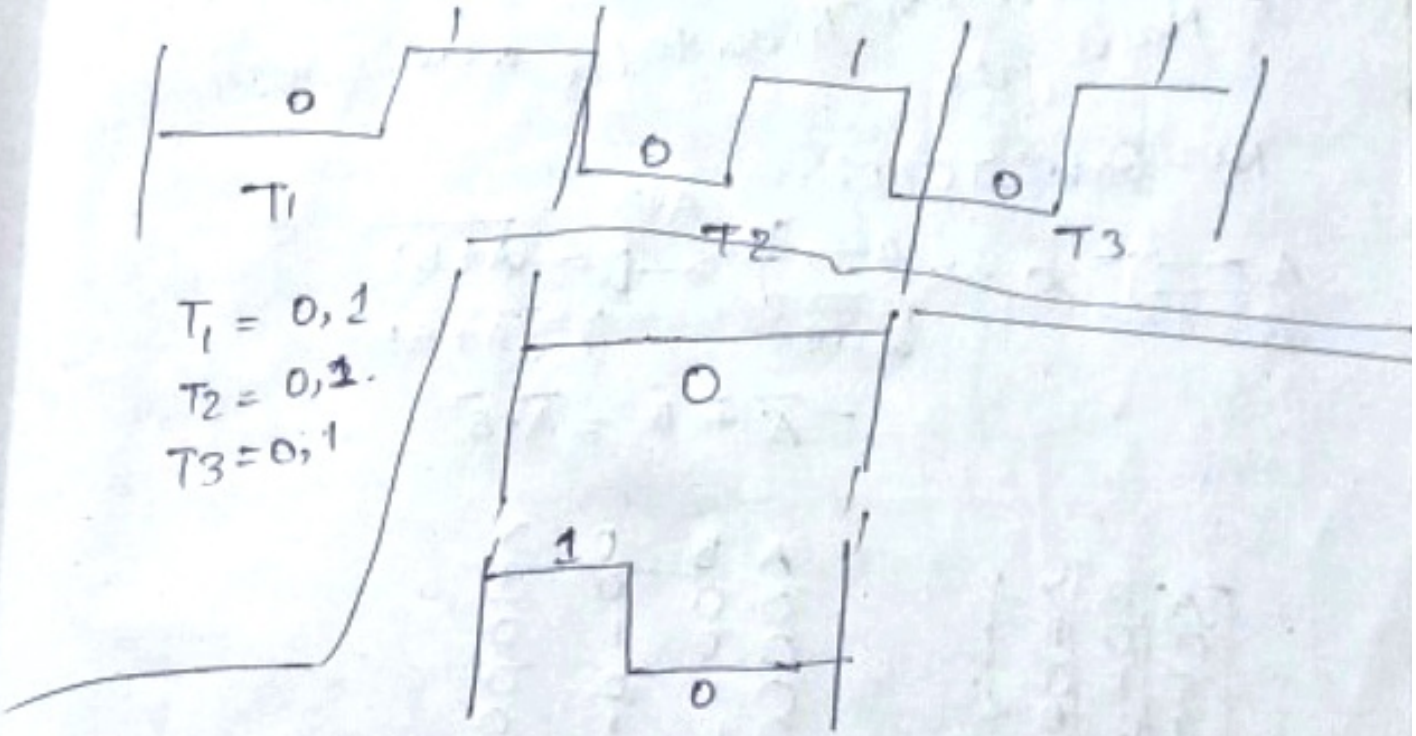
Digital Electronics.

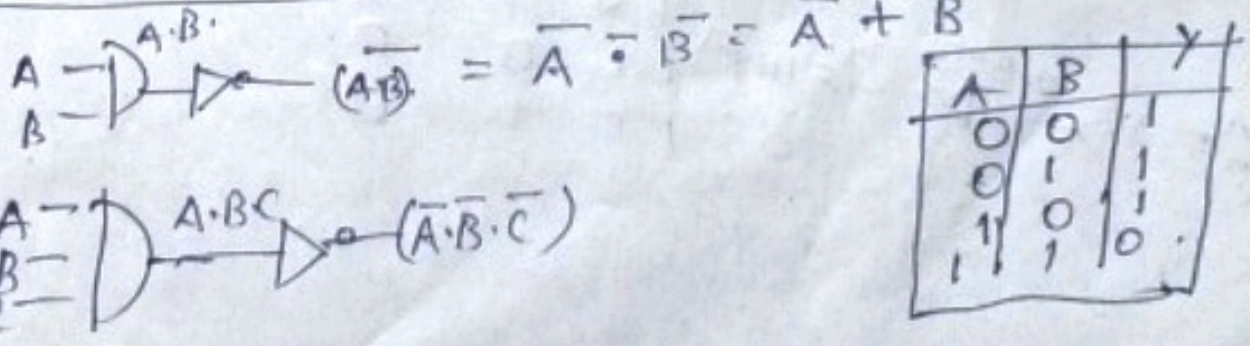


1. no system
2. Gate Theory
3. other NAND Gate.
4.

tpd = propagation delay Time.

16, 26 -- 56 will minimise Tpd.





A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

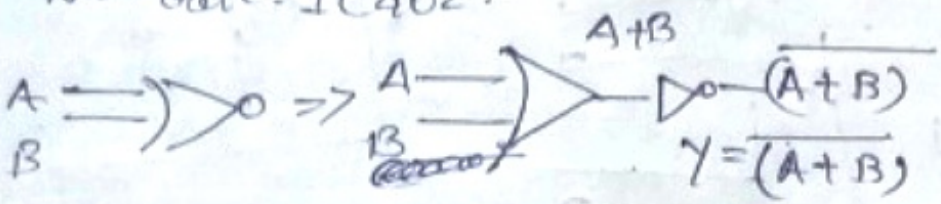
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Nand Gate IC 7400.

It is a combination of And / not gate. It has two or more than 2 inputs and only one of its input is (bar).

If A and B are two inputs then Y can be represented mathematically as $Y = A \cdot B$, here dot (·) denotes the AND operation and ( ) denotes inversion. Truth table and Symbol of Nand gate.

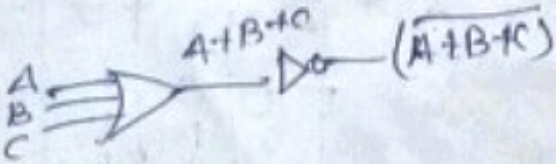
NOR Gate: IC402.



$= \bar{A} \cdot \bar{B} = \bar{A} \cdot \bar{B}$

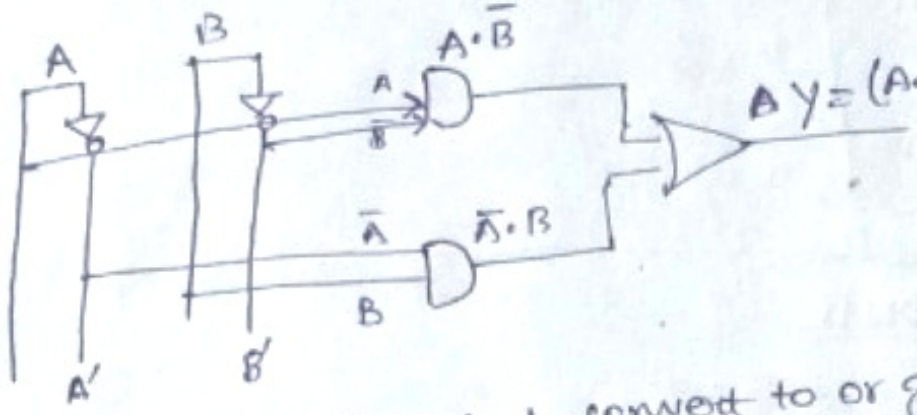
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



X-OR Gate IC 7486.

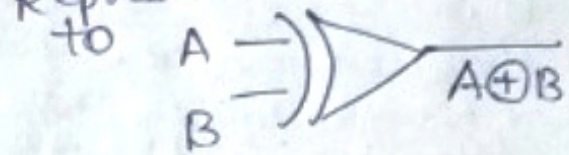
Executive gate

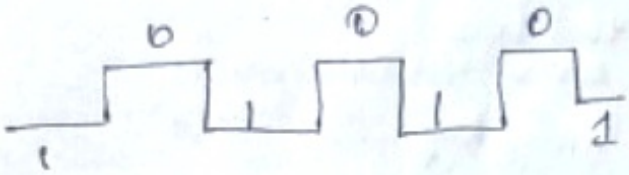


'Y' = is also X-OR.

Princ. of duality: And convert to or gate and or gate convert to And gate.

Represent equivalent to



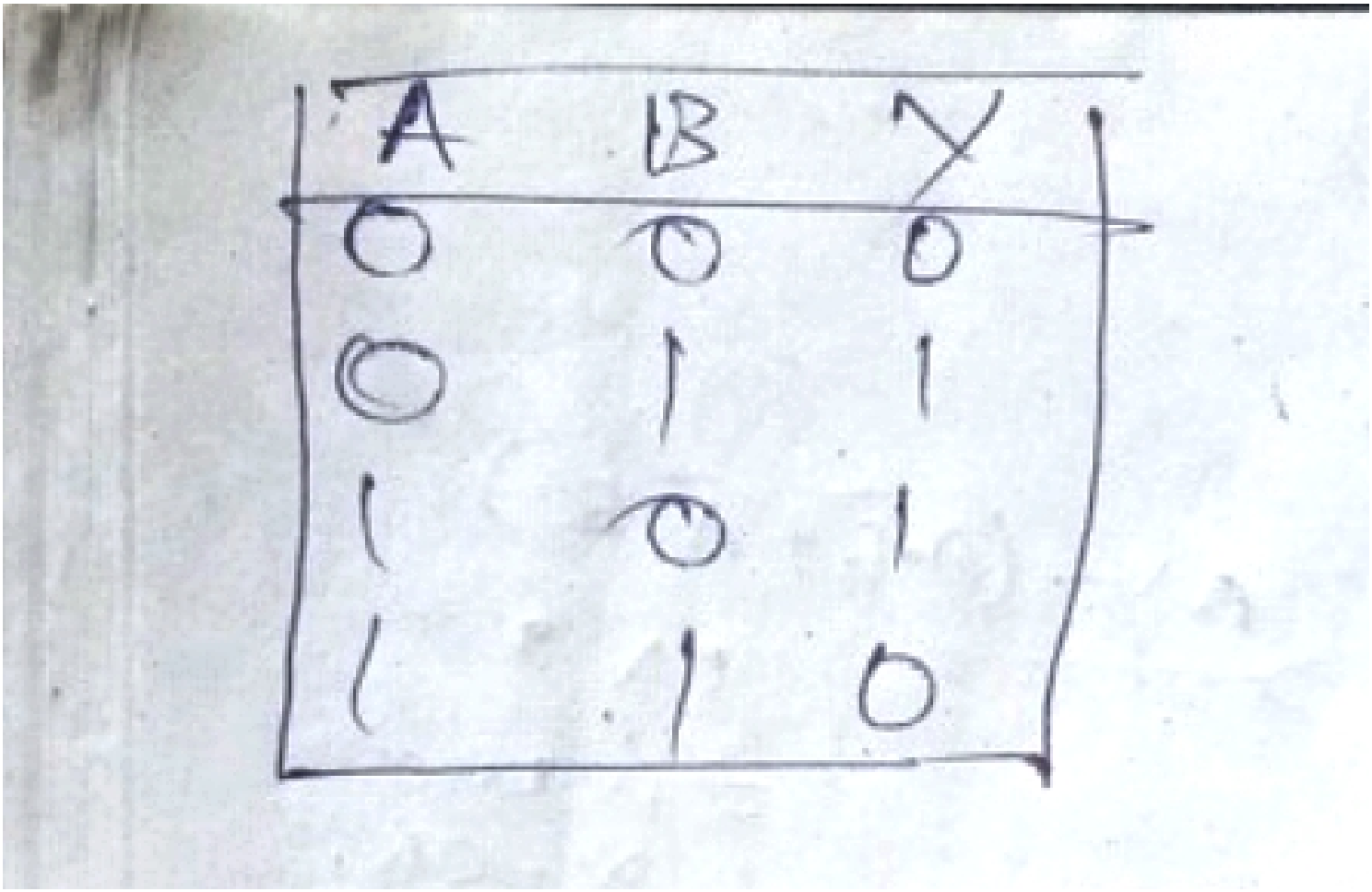


Theory: An X-OR gate is a gate with two or more than two inputs and one output.

The XOR gate gives High state. If one and only one input is 'High'. This is equivalent to saying that the output is High if either input A or input B is High exclusively and low when both are '1' or '0' at a time.

If 'A' and 'B' are two inputs. Then Output 'Y' can be given mathematically as $Y = A \oplus B$.

Here '$\oplus$' denotes the XOR operation and is equivalent to $A\bar{B} + \bar{A}B$. The Truth table and symbol of X-OR gate is shown.



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

13.11.2024. DMICRO CPU Org:

2nd memory → Primary memory → cache memory.

hit: CPU was not workless. Throughput high

miss: CPU did not go from primary to cache memory. Throughput low.

Skill - efficiency ratio.

fetch: Primary memory

Decode & Execute: CPU → CU, ALU, Registers

Components of CPU:

- CU
- ALU
- Registers.

Registers are 3 types:

(i) Accumulator.
(ii) General purpose register
(iii) Special purpose registers.

Accum. and GPR store intermediate result / inst. for tempory.

ALU has Accum. register. 1st operand in Accum, 2nd operand to ALU.

GPR is also same as Accum but not ALU specific.

ALU: Integer Math processing

GPA: Floating point processing.

Special purpose register:
① PCR - Program counter is register that hold add. of next instruction for execution.
PCR will circulate the jobs.
CPU will not go into ideal state.

Stack org. 1 add
Accumulator org. 2 add.
General Purpose Register org 2/3 add.

PCR send next add. to MAR

(ii) stack pointer register.
(iii) Index register.
(iv) Status flag register.
(v) Instruction Register
(vi) memory address register.
(vii) memory buffer register

microprocessor
↳ bit : 64 bit 128 bit specification
↳ application basis
micro controller:
TV, AC, bridge
has embedded system for specific purpose and no multipurpose.
chip is microcontroller
↳ Architecture / H/w
↳ CISC → small inst.
↳ RISC → mic. inst.
↳ VLIWP
↳ Both CISC, RISC for microinstruction programming:
EPIC:
Explicitly

Status in 0 and 1.
[0 → 1 (if ON).
↳ To store address of operator
It is container of 1 bit of info
0 is disabled mode
1 is enabled mode.
O flag / sign flag (for +ve / -ve) / Carry flag
ally. default = 0;

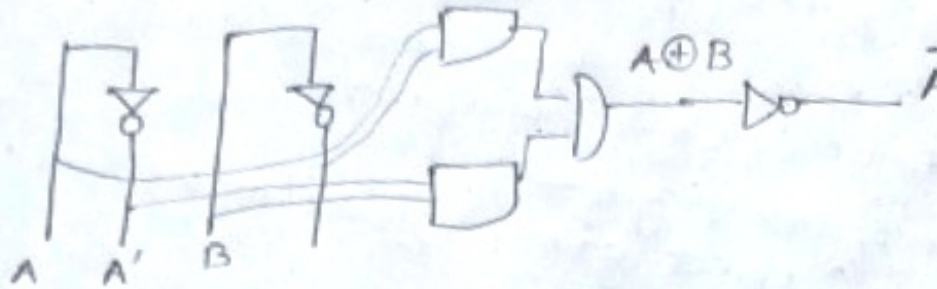
searching
flag = 0
[search ↻
[if (flag = 1)
item found.
else
item not found.

Task:
[CPU & its components]

We open file, it hold add. in mar.
The data is hold in mbr. data store in buffer.
pcr transfer info to mar. pcr does
Pipelining of job

* Obtain basic gate property of the universal gate.

XNOR:



$= (A\bar{B} + \bar{A}B)$

$= \overline{A\bar{B}} \cdot \overline{\bar{A}B}$

$= \cancel{\bar{A}B \cdot A\bar{B}}$

$\overline{A \oplus B} = A \odot B = (\bar{A}+B)\cdot(A+\bar{B})$

$= \bar{A}\cdot A + \bar{A}\bar{B} + AB + B\cdot\bar{B}$

$= AB + \bar{A}\bar{B} + A\cdot\bar{A} + B\cdot\bar{B}$

$= AB + \bar{A}\bar{B} + 1 + 1$

$= AB + \bar{A}\bar{B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

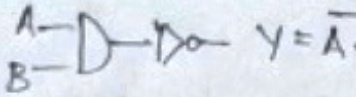
Universal Gates.

Universal gates are those gate which can be used for implementing any gate like AND OR, XOR, XNOR, NOT or any combination of basic gates.

NAND and NOR gates are universal gates.

But some rules that are need to be followed when implementing Nand or NOR based gates.

Nand Gate and NOR Gate:



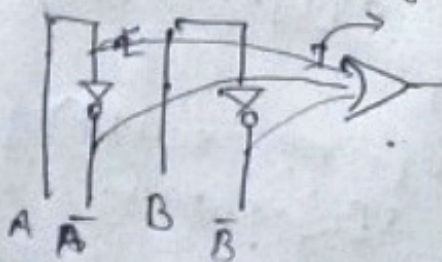
And

↓

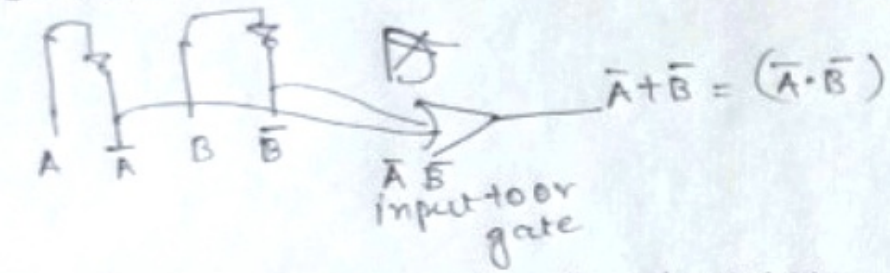
OR.

(equivalent).

we will design by



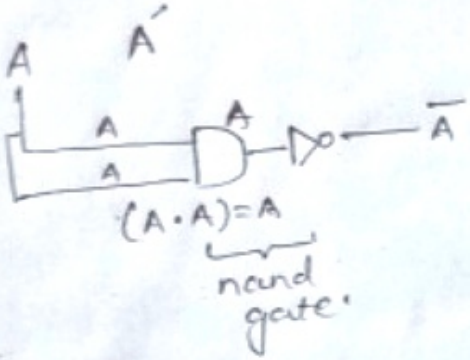
Corrected



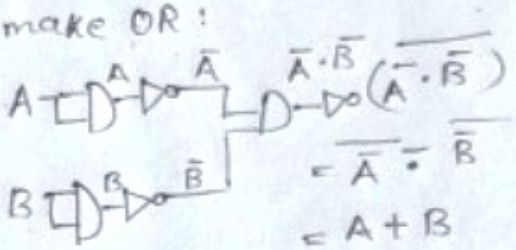
Universal Gate theory by NAND Gate;

Realize, NOT AND OR; NOR, XOR, XNOR.

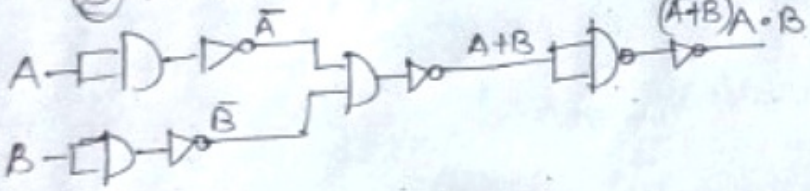
① make Not.



② make OR:



③ make and.



100

1 0 0.

Dec – 0-9, 10-19, 20-29 . . .

Bin – 0-1, 10-11, 100-101 . . .

Oct – 0-7, 10-17, 20-27 .

Hex – 0-F, 10-1F, 20-2F, 30-3F.

2^n

$2^0 = e^0 = x^0 = 1$.

1bit = 0 = 1

2bit = 1 = 2

3bit = 2 = 4.

4bit = 3 = $2^3 = 8$

G4	32	16	8	4	2	1
0 (1bit)						0
0 (4bit)			0	0	0	0
			0	0	0	1
1			0	0	1	0
2			0	0	1	1
3			0	1	0	0
4			:	:	:	:
:			1	1	1	1
15						