Digital logic

Gates:

1. $A \rightarrow \overline{A}$ — NOT Gate
2. $A, B \rightarrow AB$ — AND Gate
3. $A, B \rightarrow A+B$ — OR Gate
4. $A, B \rightarrow \overline{AB}$ — NAND Gate
5. $A, B \rightarrow \overline{A+B}$ — NOR Gate
6. $A, B \rightarrow A \oplus B$ — XOR Gate
7. $A, B \rightarrow A \odot B$ — X-NOR Gate

→ Even no. of not gates — Buffer / Bistable multivibrator

→ Odd no of NOT Gates — Inverter.

- Astable multivibrator
- Squarewave / clock generator

$T = 2n\,tpd$

$f = \frac{1}{2n\,tpd}$

(n = no of not gate in loop.

TTL — floating terminal = 1

ECL — floating terminal = 0

⇒ NAND, NOR are universal logic and follows Commutative law, but not associative.

⇒ Except NAND, NOR remaing all follows Commutative and Associative both

$\overline{A.B}$

$\overline{A+B}$

$\overline{A}+B$

$A+\overline{B}$

$\overline{AB}$, $A\overline{B}$

Multiplexer

Decoder + OR logic

All are Universal

No. of NAND & NOR for other gates

	NAND	NOR
NOT	1	1
AND	2	3
OR	3	2
X-OR	4	5
X-NOR	5	4
NAND	1	4
NOR	4	1.

No. of NAND Gates Required:

Case 1: $A.B.\overline{C}.D...$

$NAND = (2n-2)+k$, $NOR = (3n-3)-k$

(k = complement, n = variables)

Case 2: $A+B+\overline{C}+D...$

$NAND = (3n-3)-k$, $NOR = (2n-2)+k$

Case 3: $f = AB+CD$ ⇒ 3 gates

Case 4: $\overline{AB}+A\overline{B}$ ⇒ XOR = 4 gates

Case 5: $f = (A+B).(C+D)$ ⇒ 3 gates

$A \oplus A = 0$ (for even) → X-OR Gate output will be high when odd no of 1s are connected in the input so also called odd no. of 1s detector

$A \oplus A \oplus A = A$ for odd

$A \oplus 0 = A$

$A \oplus 1 = \bar{A}$ → X-NOR is also called even parity detector / equivalence logic

$A \oplus \bar{A} = 1$

Functions

SOP. POS

- Sum of product - product of Sum

$f(A,B) = AB + \bar{A}B$. $f(A,B) = (A+\bar{B})(B+\bar{A})$

⇒ Standard Canonical form: Each term should compulsorily contain all the variables.

Distributive Theorem.

$(A+B)(A+C)$

$= A.A + A.C + B.A + B.C$

$= A + AC + AB + BC$

$= A(1+C+B) + BC$

$= A + BC$

Consensus Theorem

$AB + \bar{A}C + BC$ ← Redundant term.

$= AB + \bar{A}C + (\bar{A}+A)BC$

$= AB + \bar{A}C + \bar{A}BC + ABC$

$= AB(1+C) + \bar{A}C(1+B)$

$= AB + \bar{A}C$

- 3 variable functions
- Each term consist of 2 variables
- Each variable repeated twice except one, in that one variable repeated in complement form.

Transpose Theorem

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

$= A\bar{A} + AC + \bar{A}B + BC$

$= AC + \bar{A}B + BC$

$= AC + \bar{A}B$

De morgans Law

$\overline{ABC} = \bar{A} + \bar{B} + \bar{C}$

$\overline{A+B+C} = \bar{A}.\bar{B}.\bar{C}$

- break the bar and change the sign.

→ With n variables: 2^{2^n} distinct Expressions can be formed

- 2^n minterms
- 2^n maxterms

k-map Rule: Make less no. of groups but bigger groups.

Implicants: Total no. of min terms (1s in k-map).

Prime implicants: Total no of possibilities of formation of groups.

Essential prime implicants: If there are 2 answers for a boolean expression then the terms common in both of them are called EPI

Reduced prime implicants: Terms that are not essential prime implicants.

Dual: Conversion from one logic to the logic and vice versa.

1	0
AND	OR
.	+
NAND	NOR
XOR	XNOR
Buffer	Buffer
Inverter	Inverter

Combinational Circuit design.

1. Find #inputs and #outputs
2. Write the truth table
3. Write logical expressions
4. Minimize logical expressions
5. Hardware implementations

N-bit Comparators

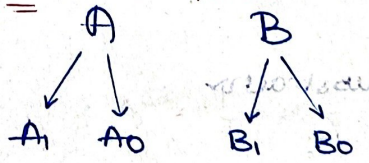
Total conditions $= 2^{2N}$

Equal conditions $= 2^N$

Unequal conditions $= 2^{2N} - 2^N$

Greater = less $= \frac{2^{2N} - 2^N}{2}$

2 Bit Comparator:



$X\ (A>B) = A_1\bar{B_1} + (A_1 \odot B_1)A_0\bar{B_0}$

$Y\ (A<B) = \bar{A_1}B_1 + (A_1 \odot B_1)\bar{A_0}B_0$

$Z\ (A=B) = (A_1 \odot B_1)(A_0 \odot B_0)$

3 Bit Comparator.

$X\ (A>B) = A_2\bar{B_2} + (A_2 \odot B_2)A_1\bar{B_1} + (A_2 \odot B_2)(A_1 \odot B_1)A_0\bar{B_0}$

$Y\ (A<B) = \bar{A_2}B_2 + (A_2 \odot B_2)\bar{A_1}B_1 + (A_2 \odot B_2)(A_1 \odot B_1)\bar{A_0}B_0$

$Z\ (A=B) = (A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$

↓ NOT, AND, OR → One 2×1 mux Required

↓ NAND, NOR, X-OR, X-NOR → Two 2×1 mux required.

Encoder: Any circuit used to convert any code to Binary.

Decoder: Circuit used to convert binary to any other code.

BCD: Binary Coded Decimal

- Weighted code
- Not a Self Complement code
- Decimal no represented by 4 bit

Excess-3 Code

- It is not a weighted code
- Self Complimented code

Half Substractor:

Diff = $A \oplus B$

Borrow = $\overline{A}B$

NAND/NOR = 5 required

Full Substractor

Difference = $A \oplus B \oplus C$

Borrow = $(\overline{A \oplus B})C + \overline{A}B$

$= \overline{A}B + \overline{A}C + BC$

NAND/NOR required = 9

1 Full Substractor = 2 Half Substractor + 1 OR Gate

Half Adder: 2 bit adder

Sum = $A \oplus B$

Carry = AB

NAND/NOR Gate Req = 5.

Full Adder: 3 bit adder

Sum = $A \oplus B \oplus C$

Carry = $(A \oplus B)C + AB$

NAND/NOR gate required = 9

1 Full Adder = 2 Half Adder + 1 OR Gate

Serial adder: Slowest adder.

$T = n \cdot T_{delay}$

(n → no of bit; T_{delay} → delay of adder)

Parallel Adder: Ripple carry adder.

$T = (n-1)T_{carry} + max\{T_{sum} + T_{carry}\}$

Look Ahead Carry Adder

$C_{i+1} = G_i + P_i C_i$

(G_i → carry generator term; P_i → carry propagating term)

	Store	Retrieve	Total
SISO	n	n-1	2n-1 → Slowest
SIPO	n	0	n
PISO	1	n-1	n
PIPO	1	0	1 → fastest.

Counters: to count no of clocks.

- used as frequency divider circuit
- used in analog to digital converter
- also known as pulse stretcher circuit

Asynchronous Counter	Synchronous Counter
(1.) One FF having external clock & o/p of that is clock for next.	All FFs are connected with the same clock.
(2.) Slower.	Faster.
(3.) Only ↑ and ↓ counting possible.	All types of counting possible.
(4.) Transition error.	No transition error.

Max no. of states = 2^n (n = no of FFs.)

Modulus of counter: Total no of states used by the counter

- up → up counter
- down → down counter

* Feedback reduces no of states.

N-bit asynchronous counter:

→ $T_{clk} \geq n \cdot t_{pdff}$.

→ $f_{clk} \leq \frac{1}{n \cdot t_{pdff}}$

Series Carry Synchronous Counter

→ $T_{clk} \geq T_{pdff} + (N-2) T_{pd\,AND}$

→ $f_{clk} \leq \frac{1}{T_{pdff} + (N-2) T_{pd\,AND}}$

* **N-bit Ring Counter**

MOD = N

N-bit Johnson Counter

MOD = 2N

Unused = $2^N - 2N$

Parallel Carry Synchronous Count.

→ $T_{clk} \geq T_{pdff} + T_{pd\,AND}$

→ $f_{clk} \leq \frac{1}{T_{pdff} + T_{pd\,AND}}$

'n' bit LACA:

for carry block → No. of AND Gate : $\frac{n(n+1)}{2}$

→ No. of OR Gate : n.

Latches: Basic memory element

- level triggered
- they have 2 output which are complement of each other

Flip flop: latches with control phenomena.

SR Flip flop (Set-Reset flip flop)

Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

$Q_{n+1} = S + \bar{R}Q_n$

1. —| — +ve level triggered
2. —o| — -ve level triggered
3. —> — +ve edge triggered
4. —o> — -ve edge triggered

JK Flip flop (Universal flip flop)

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$

* Level triggered JK flip flop suffers from "RACE AROUND" problem

* To avoid "Race Around Problem"
  1. $T_{pw} < T_{pdff} < T_{clk}$
  2. By master slave FF

Designing Flip Flop

1. Characteristic table of desired FF
2. Excitation table of avail. FF
3. Logical expn.
4. Minimization
5. Hardware implementation.

D flip flop

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

$\Rightarrow Q_{n+1} = D$

T- Flip flop

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

$\Rightarrow Q_{n+1} = T \oplus Q_n$

* To store n bit minimum 'n' FFs are required.
* Generally D-Flip Flops are used to design registers.

Base (Radix) : Total no of digit used in the system.

* Decimal Numbers — Base 10
* Binary Numbers — Base 2
* Octal Numbers — Base 8
* Hexa decimal numbers — Base — 16

Decimal to any base conversion

r	a_3 a_2 a_1 a_0
	b_0 ↑
	b_1 | Remainders
	b_2
	b_3

Any base to decimal :

$$\text{decimal} = \sum (\text{digit} \times \text{base}^{\text{dig.no.}})$$

→ 1's complement range : $(-2^{n-1}-1)$ to $(2^{n-1}-1)$

→ 2's complement range : (-2^{n-1}) to $(2^{n-1}-1)$