

Micro Computer Architecture and Logic Design.

Pdf ①

Vaccum tube - 1907

Crystal diode oscillator - 1909 (William Eccles)

Field effect transistor (FET) - 1925 (Julius Edgar)

Semiconductor - 1947 (John Bardeen, Walter Brattain, William Shockley)

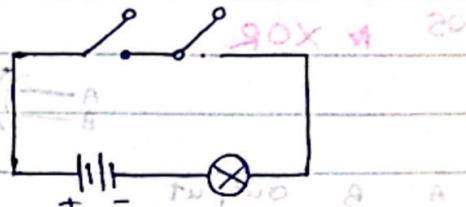
Basic logic gates.

01. AND Gate

A B A·B

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

A ————— AND ————— A·B

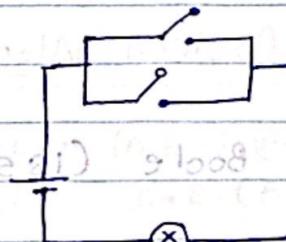


02. OR Gate.

A B A+B

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

A ————— OR ————— A+B



03. Not Gate.

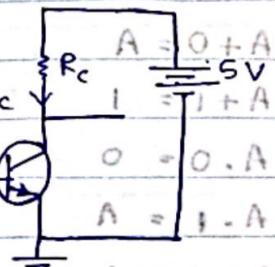
A \bar{A}

| | |
|---|---|
| 0 | 1 |
| 1 | 0 |

A ————— NOT ————— \bar{A}

$$\bar{A} = \overline{A \cdot A} = \overline{0 \cdot 0} = 1$$

$$\bar{A} = \overline{A + A} = \overline{1 + 1} = 0$$



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

When input = 0V ; $A = 0$ (0+0) current flowing in R_C is 0

Then $I_C R_C = 0V$. So $V_{CE} = V_{CC} - I_C R_C = 5 - 0 = 5V$. Output = 5V

When input = 5V ; $I_C R_C$ flowing through the transistor.

Then $V_{CE} = V_{CC} - I_C R_C = 5 - 5 = 0.7V \approx 0V$

Output = 0V $\bar{A} = \overline{A + A} = \overline{1 + 1} = 0$

04. NAND gate:

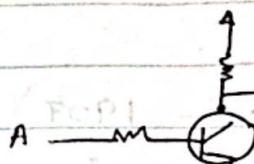
A B Output.

0 0

0 1

1 0

1 1



Output.

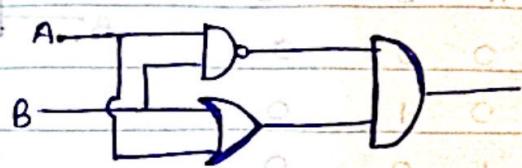
05 * XOR

A B Output



0 0 0
0 1 1
1 0 1
1 1 0

XOR gate using basic logic gates.



Boolean Algebra.

George Boole (1854).

Basic Rules of Boolean Algebra.

$$A + 0 = A$$

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A + A = A$$

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

$$A \cdot A = A$$

$$A \cdot \bar{A} = 0$$

$$\bar{A} \cdot A = 0$$

$$A + AB = A$$

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

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

De Morgan's Theorem.

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

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

Law of Identity

$$A = A \quad | \quad \bar{A} = \bar{\bar{A}}$$

Commutative Law

$$A \cdot B = B \cdot A$$

Associative Law of Multiplication

$$A \cdot (B \cdot C) = A \cdot B \cdot C$$

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

Idempotent Law

$$A \cdot A = A$$

$$A + A = A$$

Double negative law

$$\bar{\bar{A}} = A$$

Complementary Law

$$A \cdot \bar{A} = 0$$

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

Law of Intersection

$$A \cdot 1 = A$$

$$A \cdot 0 = \bar{A} + \bar{B} \cdot \bar{A} = 0$$

Law of Union

$$A + 1 = 1$$

$$A + 0 = A$$

DeMorgan's Theorem

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

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

Distributive Law

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$

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

Law of Absorption

$$A \cdot (A + B) = A$$

$$A + (AB) = A$$

Law of Common Factor

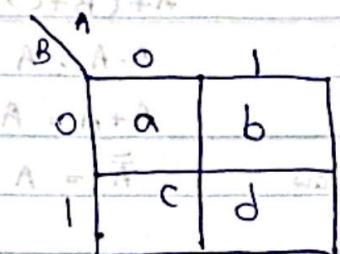
$$A \cdot (\bar{A} + B) = AB$$

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

Karnaugh maps

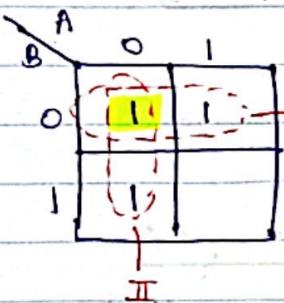
The Karnaugh map provides a simple and straightforward method of minimising Boolean expressions.

| A | B | F |
|---|---|---|
| 0 | 0 | a |
| 0 | 1 | b |
| 1 | 0 | c |
| 1 | 1 | d |



Ex:- ①

$$Z = \overline{A} \cdot \overline{B} + A\overline{B} + \overline{A}B$$



II

1. $\overline{A}\overline{B}$ දුර්වලයි, $A\overline{B}$ දැක්වා යොමු වේ. නොමැති අංකය ඇති A හි.
එහි මෙහෙයුම් ප්‍රතිඵලියක් නොමැති. $\overline{A}B$ දැක්වා යොමු වේ. $(\overline{A} \text{ මෝ})$.
 $\therefore \overline{A}$ වේ.

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

$$\therefore \text{Answer: } (\overline{A}) = \overline{A} \overline{A} + \overline{B} //$$

එහිදී රෙඛු තොරතුව ඇත් ලෙසේ දැක්වා ඇති Input සිංහල නොමැති අංකය ඇත් ඇති නොමැති අංකය ඇත් නොමැති අංකය ඇත්.

Ex:

- A, B, C භාවශක ඇති 1. Input හෝදී එන් තොරතුව C ඇති අංකය 0 භාවශක ඇති A, B තොරතුව ඇති නොමැති අංකය.

* Group නැතු නැතු කිරීමේදී ඇති අංකය ඇති අංකය ඇති නොමැති අංකය ඇති නොමැති අංකය.

Ex: ②

| | AB | C | 00 | 01 | 11 | 10 |
|--|----|----|----|----|----|----|
| | 00 | 00 | 1 | 1 | 1 | 1 |
| | 01 | 01 | 1 | 1 | 1 | 1 |
| | 11 | 11 | 1 | 1 | 1 | 1 |
| | 10 | 10 | 1 | 1 | 1 | 1 |

- ඔහි $A = 0$ ලෙස ජෑති. $C = 0$ ලෙස ජෑති. B

වොර්ට් ප්‍රතිඵලීය ප්‍රකාශනය සඳහා ප්‍රතිඵලීය ප්‍රකාශනය සඳහා ප්‍රතිඵලීය ප්‍රකාශනය සඳහා ප්‍රතිඵලීය ප්‍රකාශනය

වොර්ට් ප්‍රතිඵලීය ප්‍රකාශනය $\bar{A} \bar{C}$ නේ මෙයින් ප්‍රතිඵලීය ප්‍රකාශනය

වොර්ට් ප්‍රතිඵලීය ප්‍රකාශනය \bar{B} නේ

වොර්ට් ප්‍රතිඵලීය ප්‍රකාශනය $\bar{A} = 1$ ලෙස ජෑති. $C = 1$ ලෙස ජෑති. B වැනි

වොර්ට් ප්‍රතිඵලීය ප්‍රකාශනය $\bar{A} \bar{C}$ නේ.

Answer: $\bar{A} \bar{C} + B + AC$

③

| | AB | CD | 00 | 01 | 11 | 10 |
|--|----|----|----|----|----|----|
| | 00 | 00 | 1 | 0 | 0 | 1 |
| | 01 | 01 | 0 | 0 | 0 | 0 |
| | 10 | 10 | 1 | 0 | 0 | 1 |

$$\textcircled{1} \quad Z = \bar{A} \bar{B} \bar{C} \bar{D}$$

$$\textcircled{2} \quad Z = \bar{A} \bar{B} C \bar{D}$$

$$\textcircled{3} \quad Z = A \bar{B} \bar{C} \bar{D}$$

$$\textcircled{4} \quad Z = A \bar{B} C \bar{D}$$

By combining the first two terms above

$$\bar{A} \bar{B} \bar{C} \bar{D}$$

$$\bar{A} \bar{B} C \bar{D}$$

Brings two
outcomes

$$\bar{A} \bar{B} \bar{D}$$

$$\bar{C} \bar{B} \bar{D}$$

comes
straight

By combining the last two terms above.

$$\bar{A} \bar{B} \bar{C} \bar{D}$$

$$A \bar{B} C \bar{D}$$

horizontal
reduces

$$A \bar{B} \bar{D}$$

$$C \bar{B} \bar{D}$$

horizontal
reduces

Combine these two \rightarrow

$$\bar{A} \bar{B} \bar{D}$$

$$B \bar{D}$$

| | AB | CD | 00 | 01 | 11 | 10 |
|--|----|----|----|----|----|----|
| | 00 | 00 | 1 | 0 | 0 | 1 |
| | 01 | 01 | 0 | 0 | 0 | 0 |
| | 11 | 11 | 1 | 0 | 0 | 1 |
| | 10 | 10 | 1 | 0 | 0 | 1 |

Combinational logic; output is a function of the present value of the inputs only. not depend on past inputs.

- It has no memory
- Soon as inputs are changed, the output changes.
- The information about the previous inputs is lost.
Ex:- Adder, Encoder, Decoder, Multiplexers, Demultiplexers

Sequential logic; outputs are also dependent upon past inputs.

- It has some form of memory.
Ex:- Flip Flops, Registers, counters.

Sequential Circuits.

Sequential circuits fall into two classes:

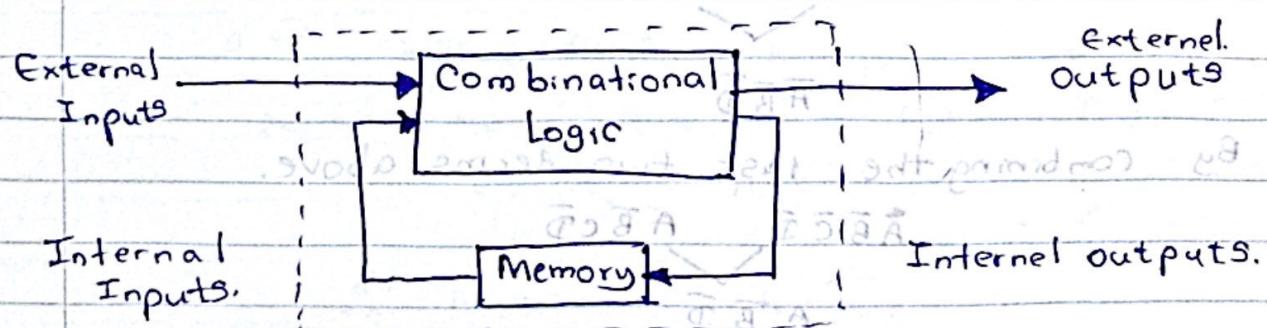
01. **Synchronous.** (மொத்த மெம்புகள் ஒன்றாக).

02. **Asynchronous.** (memory அல்லது கால நிலை விடை அல்லது flip flop அதிகாரிகள் இல்லை).

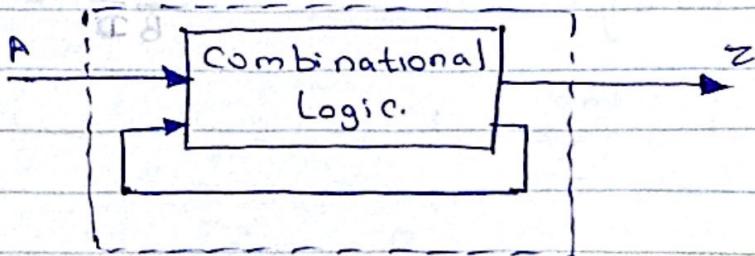
Sequential circuits are combination of combinational circuit and memory.

It depends on the present input as well as past output

Form of a Sequential Circuit

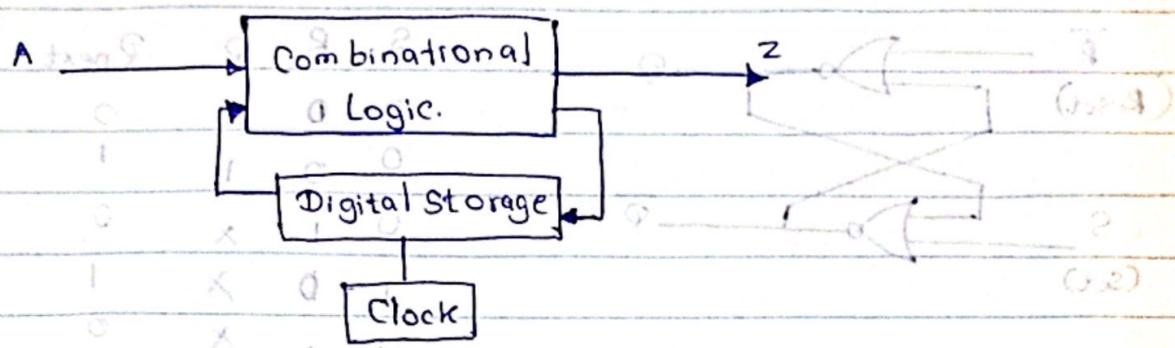


01. **Asynchronous Circuits.** (clock என்ற தீவிரமாக).



- The circuit is considered to be asynchronous if it does not employ a periodic clock signal to synchronize its internal changes of state. එම සඳහා පෙන්වනු ලබයි

02 Synchronous circuits පෙනු ඇත්තා 92



The inputs are pulses with certain restrictions on pulse width and circuit propagation delay. Use same clock pulse. එමගින් මෙහෙයුමක් අනුමත වෙන්කර තිබුණුයි.

Two types :-

01. Clocked Sequential.

02. Unclocked/pulsed Sequential

In a clocked sequential circuit which has flip flops or some instances, gated latches for its memory elements.

Latch.

A latch is a circuit that has two stable states and can be used to store state information.

අස්‍යෝජ්‍ය පෙනීම: (clock නොමැතිවා)

- Building block of sequential circuits
- without clock.
- Built from logic gates.

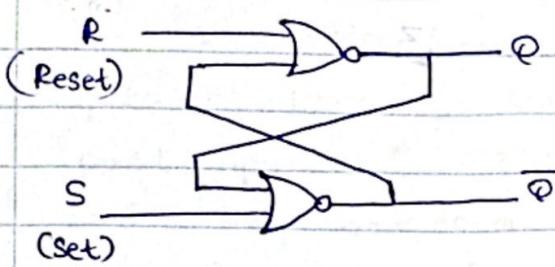
01 SR Latch

7: மூன்றாணி சுதா விகாரை அல்லது சுதா.

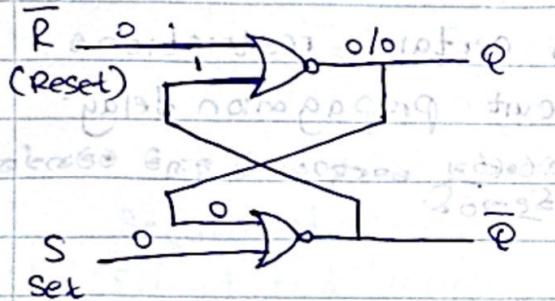
அதே பொருள் என்றும் அழைப்பது விரைவாக இருக்கிறது.

One bit of information to store.

SR Latch Using NOR gates.



| | | * தெரியாத சமயம் | | NOR Gate நிலை | | NOR Gate நிலை | |
|--|--|-----------------|---|---------------|-------|---------------|------------|
| | | S | R | Q | Q-bar | Q next | Q-bar next |
| | | 0 | 0 | 0 | 1 | 0 | 1 |
| | | 0 | 1 | 1 | 0 | 1 | 0 |
| | | 1 | 0 | x | x | 1 | 0 |
| | | 1 | 1 | x | 0 | 0 | 1 |



| S | R | Q | Q-bar |
|---|---|---|-------|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | x | x |
| 1 | 1 | x | 0 |

NOR ஒரு

மூன்றாணி சுதா விகாரை அல்லது சுதா.

(Input S கோஷி தடை, Output அதை எங்களும்)

| | | | |
|-------|-------|-------|------------------------------------|
| 0 0 1 | 1 0 0 | 0 0 1 | Invalid. |
| 0 1 0 | 1 1 0 | 0 1 0 | $Q = 0 \vee Q = 1 \text{ என்று}$ |
| 1 0 0 | 1 1 1 | 1 0 0 | ஏதோ ஒன்று என்று. |
| 1 1 0 | 1 1 1 | 1 1 0 | $\therefore \text{Invalid என்று.}$ |

* NOR என்றால் சுதா விகாரை அல்லது சுதா.

ஏனென்றால் சுதா விகாரை அல்லது சுதா.

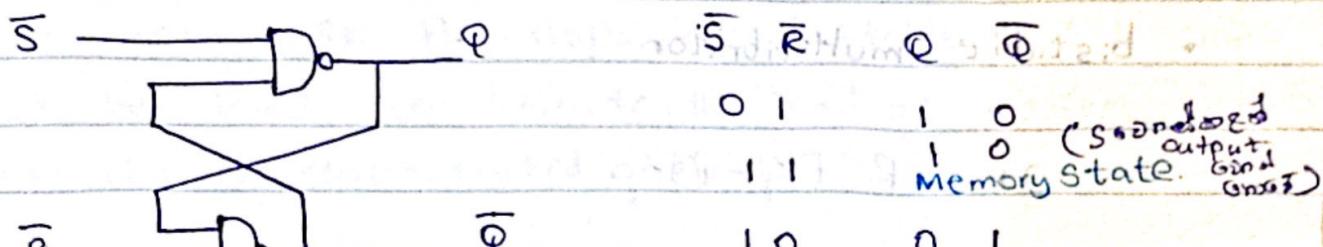
- S R Active low

0 0 என்று சுதா விகாரை அல்லது சுதா.

S, R ↑ என்று சுதா விகாரை அல்லது சுதா.

Allas

SR Latch using NAND gates



NAND ලේ

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

* මෙහි දැක්වා ඇතුළුව

output නෑගෙනු

ඉගිරි රික්

සිංහල්.

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 |

memory state.

Invalid.

(Q, Q-bar පෙනෙමි විය ගොනාකේ.)

* NOR - SR Latch වල memory state තුළුවාට යෝම්බාවේ යෝම්බාවේ

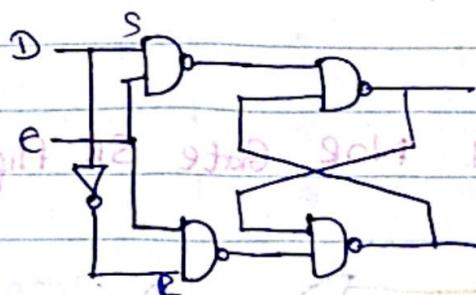
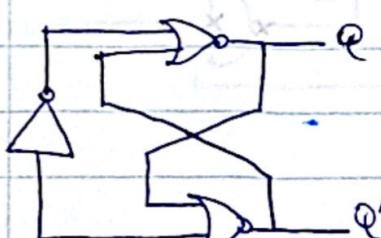
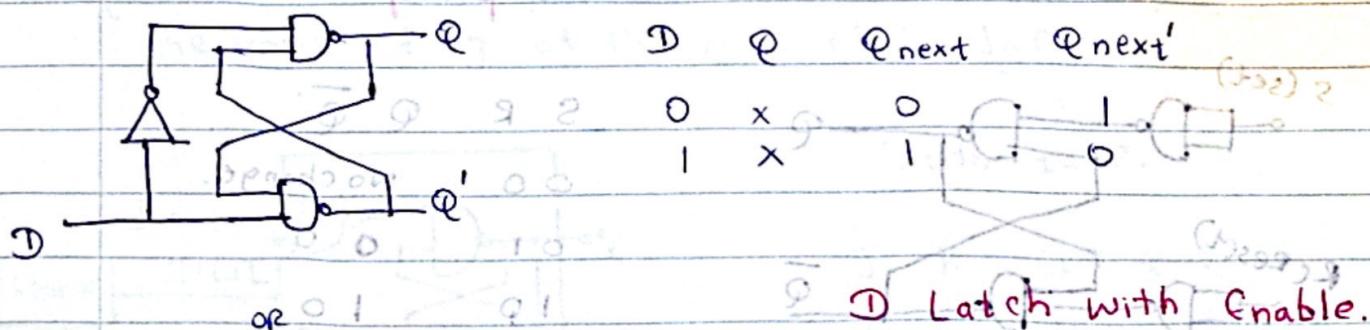
Input යුතු ඕ විවෘතු.

* NAND - SR Latch වල memory state තුළුවාට යෝම්බාවේ

Input යුතු ඕ විවෘතු.

Disadvantage:- need to ensure that two inputs S and R are never de-asserted at the same time!

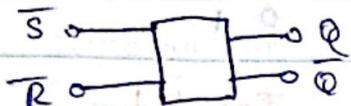
02. ① Latch.



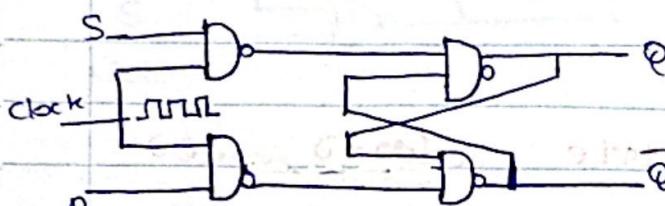
Flip Flop

- bistable multivibrator.

Q1. SR Flip-Flop



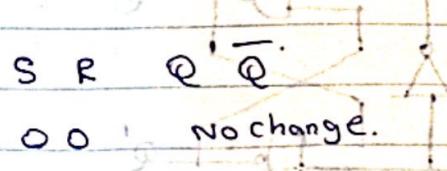
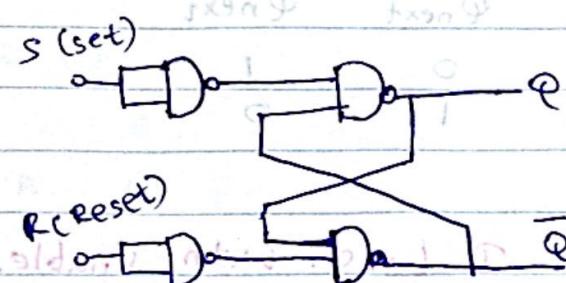
I. Basic SR flip flop



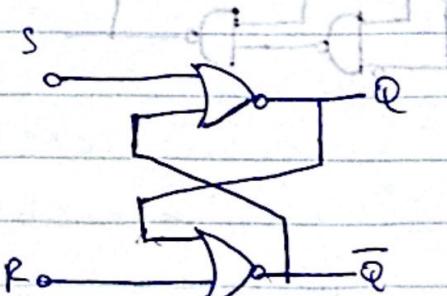
- କେତ୍ରିତ NAND ଦେ କାହାରେ ପାଇଲା
- $x, y = 1$ ହାତ ମେମୋରୀ ସ୍ଟେଟ୍
- କେବଳ ଅନୁଷ୍ଠାନିକ କାର୍ଯ୍ୟ

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |

II. Positive NAND Gate SR Flip Flop



III. NOR Gate SR, flip flop.

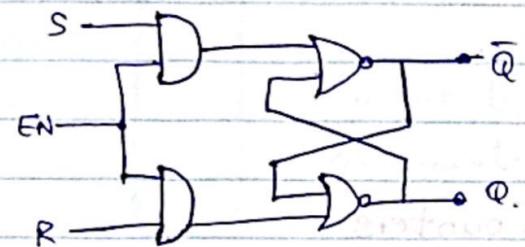
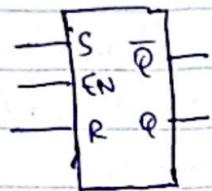


| S | R | Q | \bar{Q} | |
|---|---|---|-----------|------------|
| 0 | 0 | 0 | 1 | No change. |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | X | X | Invalid |

Switch Debounce

Set-reset SR Flip-flops or Bistable latch circuits can be used to eliminate this kind of problem, and this is demonstrated below.

Clocked SR Flip-Flop.



EN - Extra conditional input.

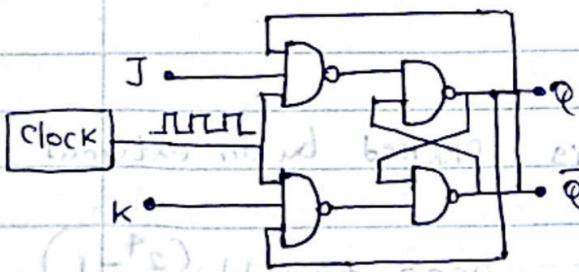
JK Flip-Flop



Most versatile of the basic flip-flops. A : output out

J and K are different then the output Q takes the value of J at the next clock edge.

Truth table.



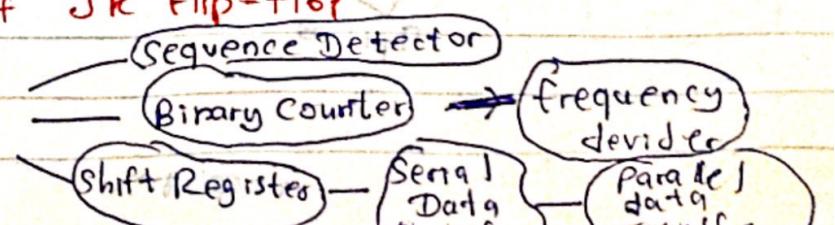
| J | K | CLK | Q |
|---|---|-----|---------------|
| 0 | 0 | 0 | Q (no change) |
| 1 | 0 | 1 | High |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | Q (toggles) |

J, K both Low \rightarrow no changes.

J, K both high \rightarrow toggle from one state to other.

Applications of JK flip-flop

JK flip flop



* No Latch. - A Latch is a circuit that has two stable states and can be used to store state information.

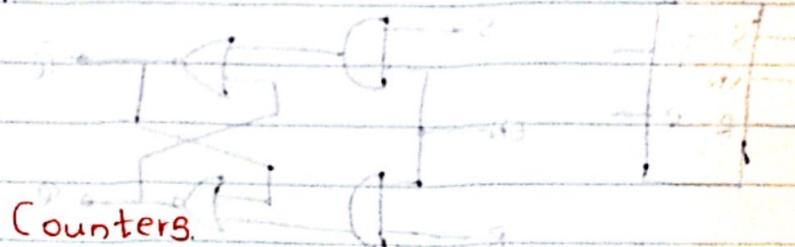
Latch

Flip-Flop.

- has a feedback path.
- memory devices.
- asynchronous
- Output can change as soon as the inputs do.

- edge-triggered and only changes state when a control signal goes from high to low or low to high.

PDF (4)



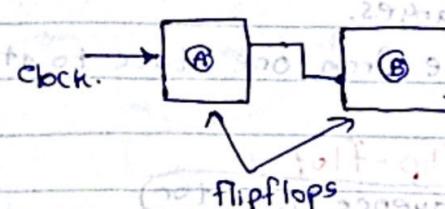
A counter is a digital sequential logic device that will go through a certain predefined states based on the application of the input pulses.

Two types: Asynchronous counters (Ripple counters.)

Ripple counter.

- An asynchronous counter.
- Only the first flip-flop is clocked by an external clock.
- Range of a 4-bit counter is 0000 to 1111 ($2^4 - 1$)

Asynchronous counters;



Output of a flip flop is a clock for B flip flop.

Asynchronous CounterSynchronous Counter

- 1. Flip flops are connected in such a way that the output of the first flip flop drives the clock of next flip flop.
- 2. Flip flops are not clocked simultaneously (sequentially).
- 3. Circuit is simple for more number of states.
- 4. Speed is slow as clock is propagated through numbers of stages.
- 5. There is no connection between output of first flip flop and clock of next flip flop.
- 6. Flip flops are clocked simultaneously.
- 7. Circuit becomes complicated as number of states increases.
- 8. Speed is high as clock is given at same time.

Decade Counter (BCD)

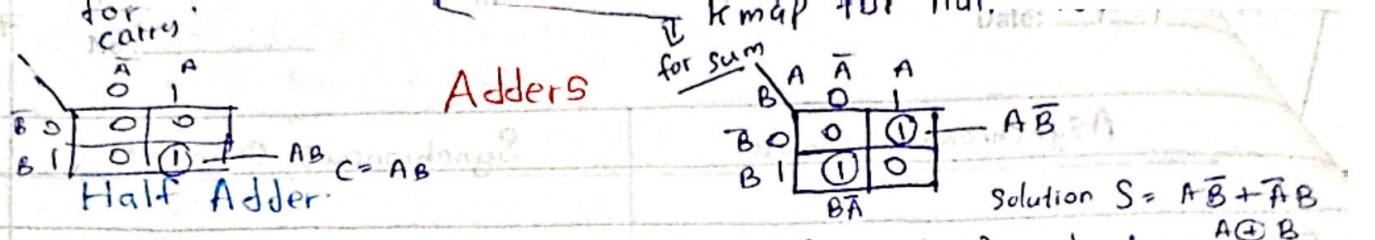
A decade counter is a binary counter that is designed to count to 10_{10} or 1010_2 .

Necessity of Synchronous Counters

- Asynchronous counters are slower than synchronous counters. Reason \rightarrow delay in the transmission of the pulses from flip-flop to flip-flop.

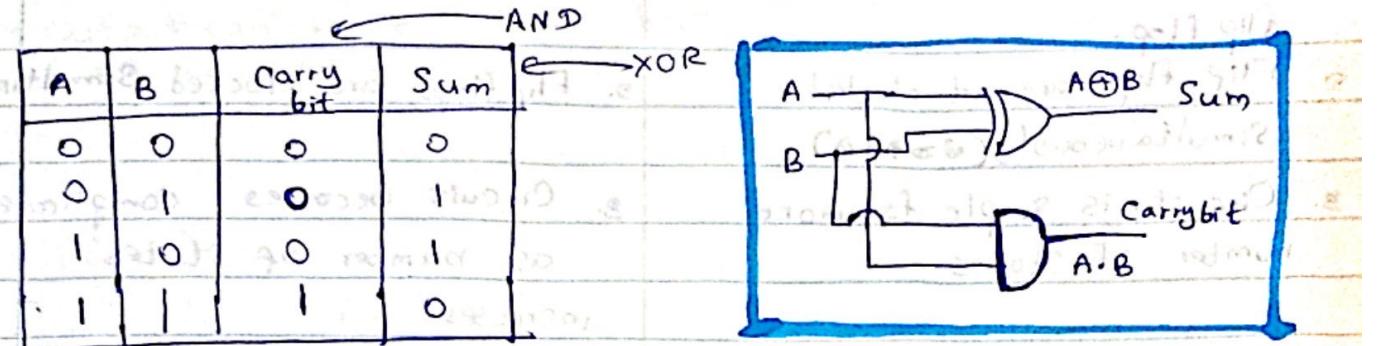
Counter and Decoder

14 pin 7490 counter chip and 16-pin decoder chip are often used together to drive 7-segment displays.



So we will need two inputs (a and b) and two outputs.

Bits 2 ഫീ ഉള്ള കീറ്റോ കാലിക്ക കരുവ്.



Half Adder at a Glance.

Carry bit $\rightarrow \underline{\underline{1}} 0$

$\left. \begin{array}{l} \text{Sum} = A \text{ XOR } B = A \oplus B \\ \text{Carry} = A \text{ AND } B = A \cdot B \end{array} \right\}$

Full Adder.

for Sum

Kmap \rightarrow

| | | | | |
|---|-----|-----|----|----|
| | 00 | 01 | 10 | 11 |
| 0 | 0 | (1) | 0 | 1 |
| 0 | (1) | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |

$\text{Sum} = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC$

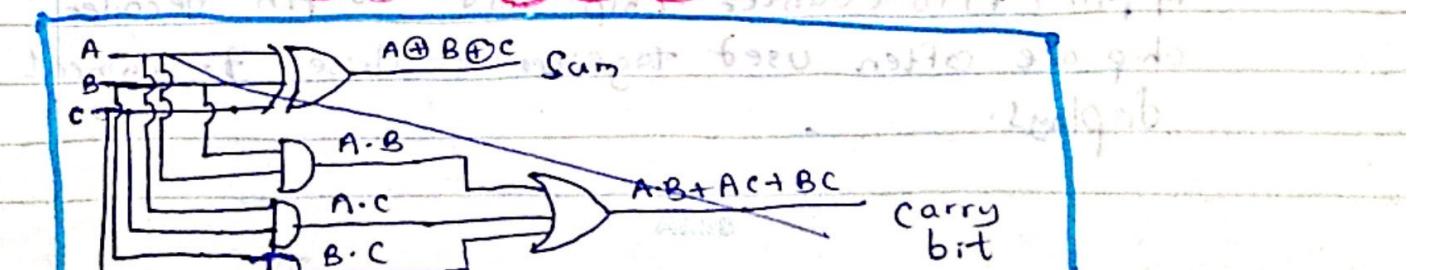
$= \bar{A}(B\bar{C}) + A(B\bar{C})$

$S = A \oplus B \oplus C$

| A | B | C_{in} | C_{out} | Sum |
|---|---|----------|-----------|-----|
| 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 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |

for C_{out} .

$\left. \begin{array}{l} \text{Carry Out} = A \cdot B + A \cdot C + B \cdot C \\ \text{Sum} = A \oplus B \oplus C \end{array} \right\}$



Why we need Full adder

There is no provide Carry-in from the previous circuit when adding together multiple data bits in Half adder.

How can a full-adder be converted to Subtractor?

$$A - B = A + (2^3 \text{ complement of } B)$$

Ex: $A = 12, B = 7$

$1^3 \text{ complement of } B = 1000$

$2^3 \text{ complement of } B = 1001$

So $A + (2^3 \text{ complement of } B) = 1100$

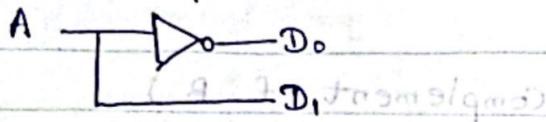
$$\begin{array}{r} 0 0 1 0 \\ + 1001 \\ \hline 10101 \end{array}$$

Inverting the Carry bit the answer will $00101 = 5$

Decoder.

A decoder is a circuit that changes a code into a set of signals.

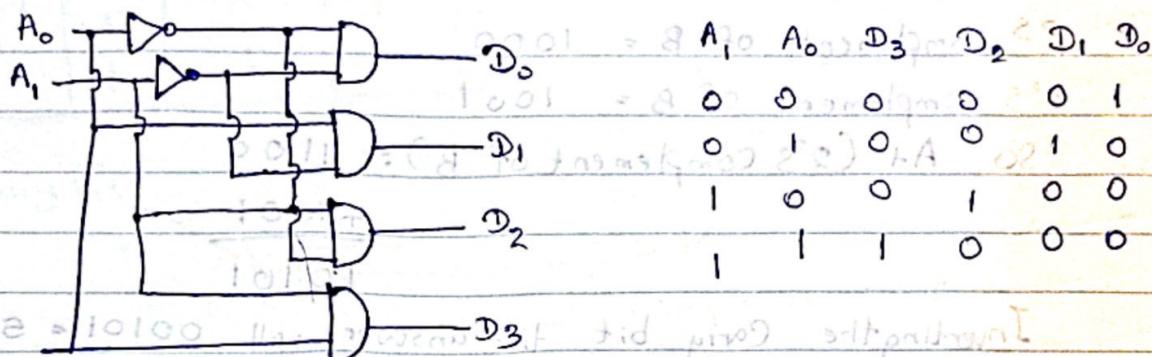
Simple 1 to 2 line decoder. Not a very useful



| A | D ₀ | D ₁ |
|---|----------------|----------------|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

1 to 4 line decoder.

$$F = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$



- Seven Segment decoder is one of the most useful decoder.

Memory registers.

Memory address register (MAR)

- CPU register, store the memory address from which data will be fetched to the CPU or the address to which data will be sent and stored.
- Holds the memory location of data that needs to be accessed
- When reading from memory, data addressed by MAR is fed into the MDR and then used by the CPU.
- When writing to memory, CPU writes data from MDR to the memory location whose address is stored in MAR.

Encoder

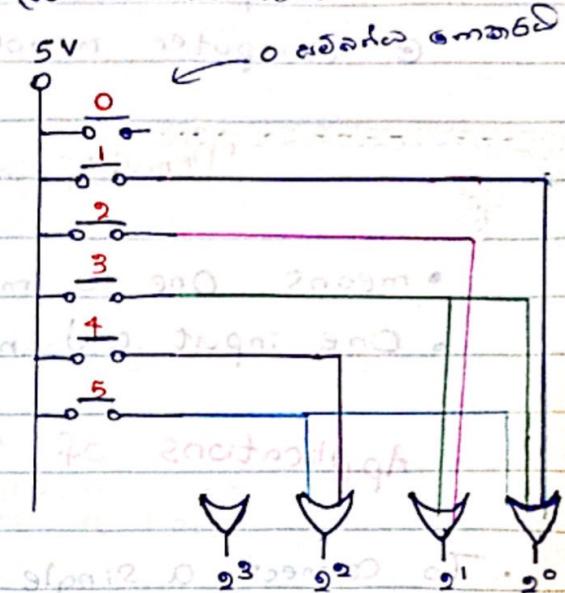
කේතී BCD නොවුම් දූෂ්චරණය සඳහා 4 digits ලබාදිය යුතු හැකි නි.

එසේම තුළ නොවුම් OR දීමාරු මෙහෙයුම් කරනු ලැබේ.

Convert Decimal to binary.

| Decimal Digit | BCD |
|---------------|---------|
| 0 | 0 0 0 0 |
| 1 | 0 0 0 1 |
| 2 | 0 0 1 0 |
| 3 | 0 0 1 1 |
| 4 | 0 1 0 0 |
| 5 | 0 1 0 1 |

$$2^3 \quad 2^2 \quad 2^1 \quad 2^0$$



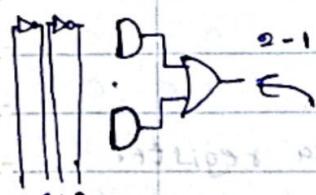
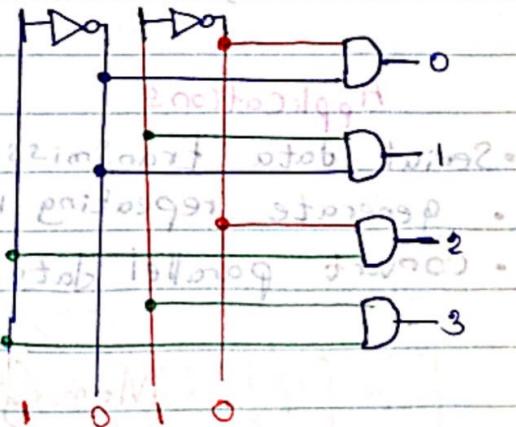
මෙහෙයුම් සඳහා 0 සිංහල පිටපත් නොවුම් විය ඇති නි.

Decoder.

Convert binary to Decimal

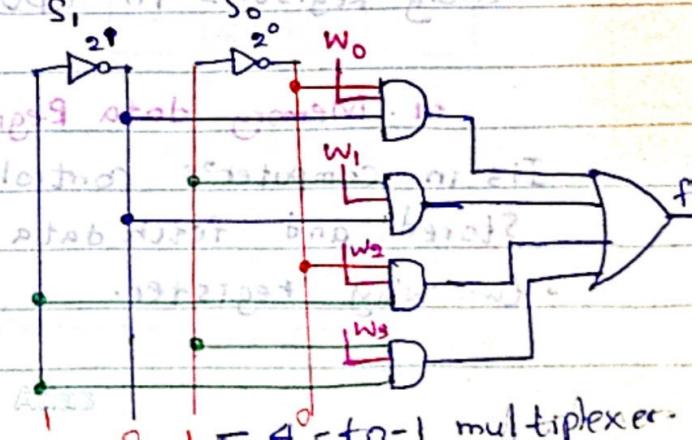
2³ 2² 2¹ 2⁰ සඳහා 0 1 2 3 4 5 6 7 8 9 නි.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |



Multiplexer

| S ₁ | S ₀ | f | w ₀ |
|----------------|----------------|---|-----------------------------------|
| 0 | 0 | 0 | W ₀ (GND) |
| 0 | 1 | 1 | W ₁ (V _{DD}) |
| 1 | 0 | 1 | W ₂ (V _{DD}) |
| 1 | 1 | 0 | W ₃ (GND) |



Many into one
many to one
(many to one)

Applications of Multiplexers.

- Many Sensors connected to the microcontroller
- Communication System
- Telephone network
- Computer memory

Demultiplexer.

- means One to many.
- One input and many output.

Applications of Demultiplexer.



- To connect a single source to multiple destinations
- Communication System
- ALU
- Serial to parallel converter.

Shift Registers

like counters, are a form of sequential logic.

Applications

- Serial data transmission
- generate repeating waveforms.
- Convert parallel data to serial form.

Memory Registers

Memory Registers in CPU \rightarrow Memory data register

Memory address register

Memory data Register (MDR)

- Its in computer's control unit. Contains data to be stored and fetch data from the computer storage.
- two way register.

(In start of pdf's note
in this chart note)

$$C + \overline{BC}$$

$$C + \overline{B} + \overline{C} \quad \begin{array}{l} (\text{De Morgan's}) \\ (\text{Commutative, Associative Laws}) \\ (\text{Complementary Law}) \end{array}$$

1 . (Euclid) Identity Law.

2) $\overline{AB} (\overline{A} + B) (\overline{B} + B)$.

$$\overline{AB} (\overline{A} + B) \quad | \quad (\text{complementary Law})$$

$$(\overline{A} + \overline{B})(\overline{A} + B) \quad (\text{Demorgan's Law}).$$

$$\overline{A} + \overline{B}B \quad (\text{Distributive Law})$$

$$\overline{A} \quad (\text{complement, Identity})$$

3) $(A+C)(AD + A\overline{D}) + AC + C$ (Distributive, Associative Law)

$$(A+C) A(D + \overline{D}) + AC + C \quad (\text{Associative Law})$$
$$(A+C) A \times 1 + AC + C \quad (\text{complementary Law}), \text{Identity.}$$
$$A((A+C) + C) + C \quad (\text{commutative, Distributive})$$
$$A(A+C) + C \quad (\text{associative, Idempotent.})$$
$$AA + AC + C \quad (\text{Distributive})$$
$$A + C \quad (\text{Identity})$$

4) $\overline{A}(A+B) + (B+AA)(A+B)$

$$\overline{A}A + \overline{A}B + B + AA + B + A$$

5) Kmap

Kmap

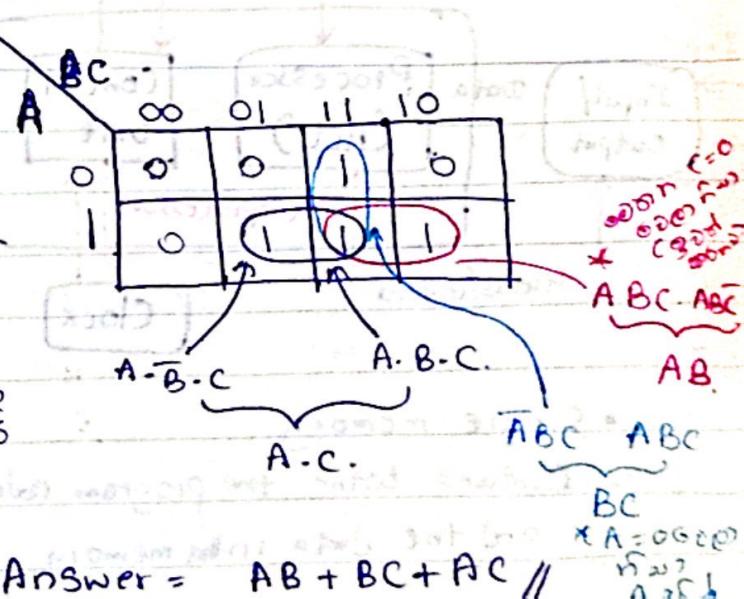
Date.....

Note.....

c)

| A | B | C | Z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

* ലൈംഗിക ഫലം
Input കുറഞ്ഞെങ്കിൽ
പാരമായ രീതി പുറത്ത്
Input കുറഞ്ഞെങ്കിൽ



$$2). Z = \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot C$$

| | AB | 00 | 01 | 11 | 10 |
|---|----|-----|-----|-----|-----|
| C | 0 | (1) | (1) | (1) | |
| | 1 | | (1) | (1) | (1) |

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

$$\bar{A}\bar{C} + B + AC$$

* මෙහි Inputs සඳහා ප්‍රකාශනය කළ නිසුම (AC | AB)
දැන ඇත

Microcomputer.

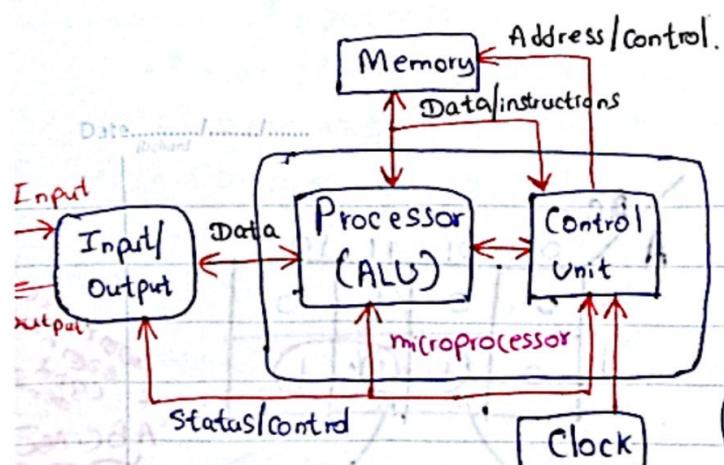
A single printed circuit board usually connects the ICs, making a computer called a microcomputer.

Microcomputer - Relatively small and inexpensive computer that is contained on one or few chips.

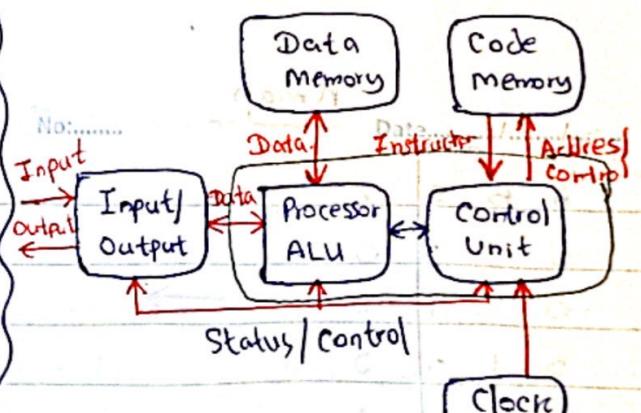
Microcontroller - A single-chip microcomputer.

Microprocessor - The processor and control unit part of the single-chip computer.

Princeton Computer Architecture (Von-Neuman)



Harvard Architecture.



- Single memory.
- Contains both the program code and the data in the memory

- Two separate memories:
- One memory contains only data and other one contains program code.

Major Components

1. Processor

- Contains ALU (Arithmetic Logic Unit)
- do operations → addition, subtraction, bit-wise AND and OR, shift operators.
- has registers (groups of D-flipflops used to store binary values)
- also generates signals that indicates when values are negative, zero or when arithmetic overflow occurs.

2. Control Unit

- A synchronous sequential machine that coordinates the flow of data between the other units and operations of other blocks.

3. Memory

Stored program code and data.

A sequence of directly addressable locations.

4. Clock

A periodic signal for the sequential machine in the control unit.

5. Inputs / Outputs

Interface between the internals of the microcomputer and the outside world.

Instruction codes.

It consists Operation Code and Operand.

Operation Code - Microcomputer what action to perform and how to interpret the operand.

Operand.

Contains the data that microcontroller will perform the action onto → ~~instruction~~ object.

~~byte~~ Instruction length. To equip → register read.

- Fixed length → Each instruction is the same size and has the same number of bits as all others.
- Variable length → The length of each instruction may be different.

~~address~~ ROM (Read Only Memory)

ROM ← EPROM (Erasable PROM)

EEPROM (Electrically Erasable PROM)

RAM ← Static RAM (SRAM) FPM (Fast Page Mode)

Dynamic RAM (DRAM)

 ECC (Error Correcting Code)

 EDO (Extended Data Output)

 SDRAM (Synchronous Dynamic RAM)

Universal gates සාර්ව්‍ය කාරුණික පෙන්‍ය සහ පෙන්‍ය ස්ථූල් (Basic)

සෑම්බුල්.

Universal gates විෂය යුතු කිරීමේදී. එහාට NAND නීතු NOR gates සඳහා.

NAND

01) NAND \rightarrow NOT

| NOT | |
|-----|-----------|
| A | \bar{A} |
| 0 | 1 |
| 1 | 0 |

| NAND | | |
|------|---|-------------------------|
| A | B | $\bar{A} \cdot \bar{B}$ |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

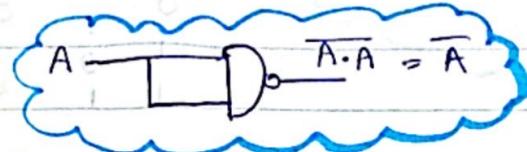
NAND gate එකේ Input දැක්වා

එකතු ඇත්තේ තුළු මුදල Output එක සහ NOT මුදල නෑත්තේ.

$$\begin{matrix} A & B \\ 0 & 0 \\ 1 & 1 \end{matrix} \rightarrow \begin{matrix} \text{Out} \\ 1 \\ 0 \end{matrix}$$

\therefore NAND gate එකතු Input දැක්වා

එකතු ඇත්තේ දුන්වේ Output එක NOT value එකශෝර්.

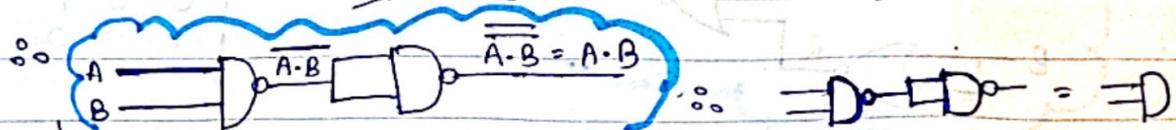


$$\therefore \neg \neg D = D$$

02) NAND \rightarrow AND

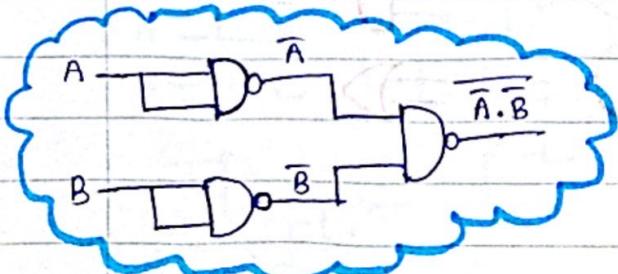
NAND එකේ Inverse (NOT) එක AND නීතු NAND

සෑම්බුල් NOT මුදු ඇතුළු තුළු ආකෘති තැපැලු.



| A | B | $\bar{A} \cdot \bar{B}$ | $\bar{\bar{A}} \cdot \bar{\bar{B}}$ |
|---|---|-------------------------|-------------------------------------|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |

03) NAND \rightarrow OR

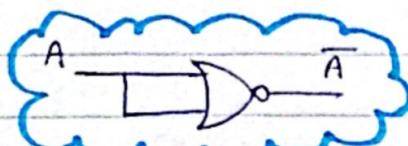


| A | B | \bar{A} | \bar{B} | $\bar{A} \cdot \bar{B}$ |
|---|---|-----------|-----------|-------------------------|
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

OR.

NOR

01) NOR \rightarrow NOT

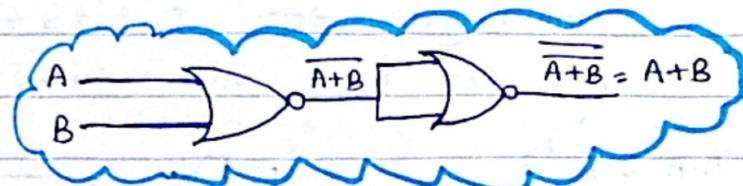


Input නුවම එකතු කළ හා පෙන්වන
Input විසින් NOT තොකැලුවේ

A B $\overline{A+B}$

| | | |
|---|---|-----------------|
| 0 | 0 | $\rightarrow 1$ |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | $\rightarrow 0$ |

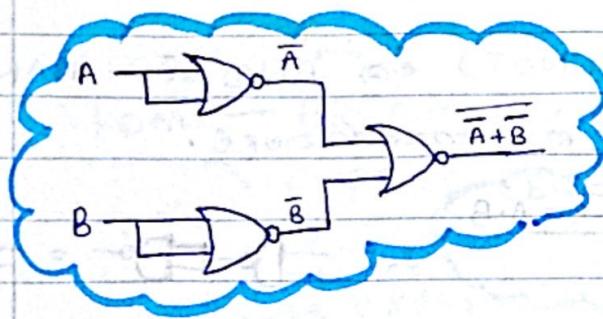
02) NOR \rightarrow OR



| A | B | $\overline{A+B}$ | $\overline{\overline{A+B}}$ |
|---|---|------------------|-----------------------------|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |

OR

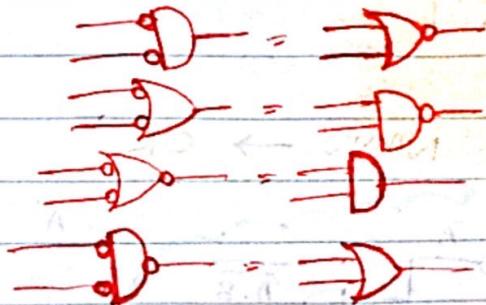
03) NOR \rightarrow AND

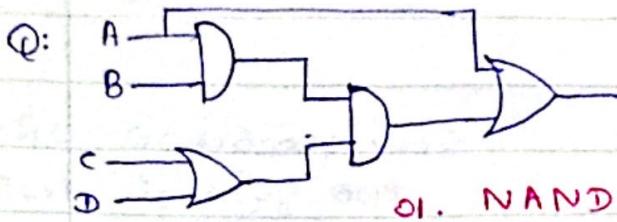


| A | B | $\overline{A}\overline{B}$ | $\overline{\overline{A}\overline{B}}$ |
|---|---|----------------------------|---------------------------------------|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 |

AND

- Bubbled AND = NOR
- Bubbled OR = NAND
- Bubbled NOR = AND
- Bubbled NAND = OR

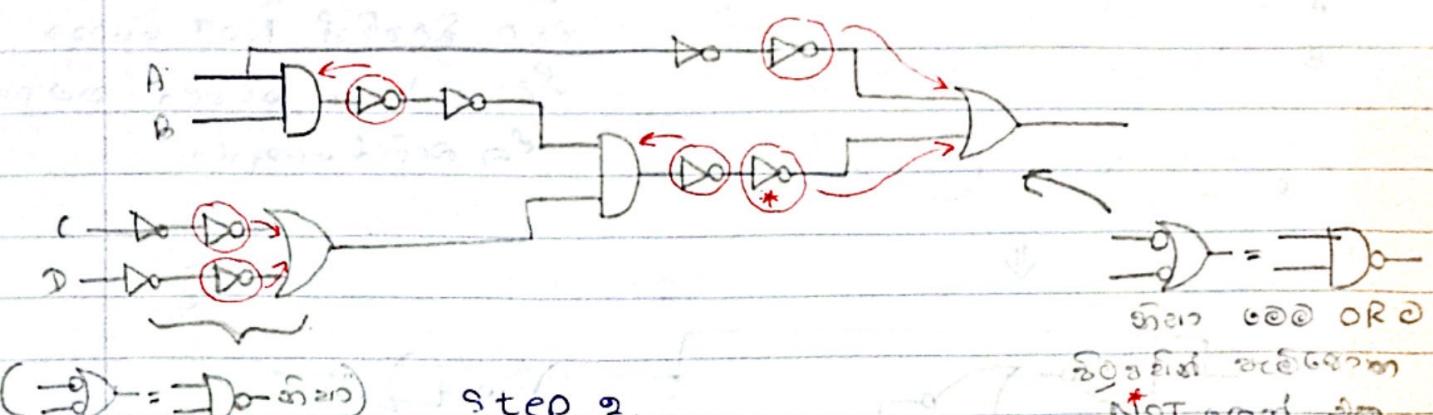




I Convert this -NAND gates
and NOR gates.
Q1. NAND gates only.

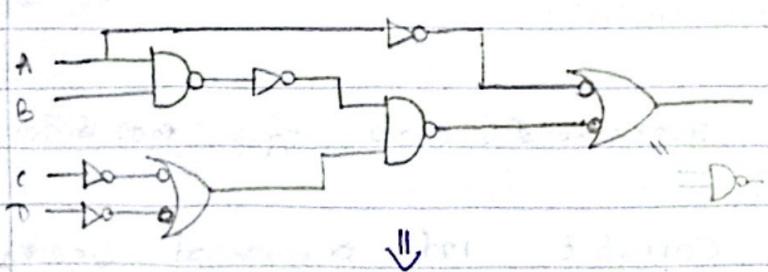
Step 1.

මෙහිතිනේ AND අනු OR gate
ක්‍රම තුළ ඇති අවසාන NOT gates ඉදිකිරීමේ පෙනුමෙන්.
උග් balance කිරීමෙන් සහ NOT gate එකට ඉදිකිරීමේ තැන් NOT
gate එකක් යොමු කළා.

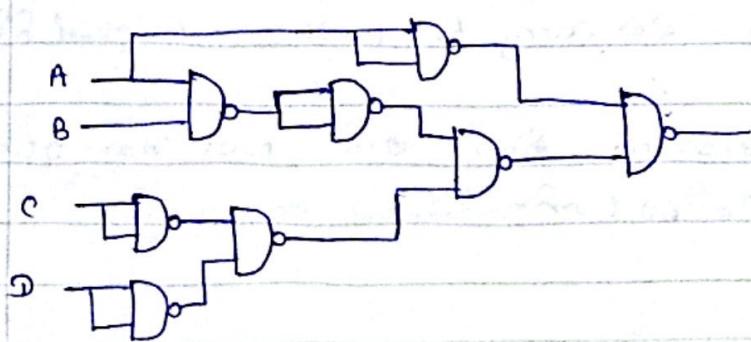


Step 2.

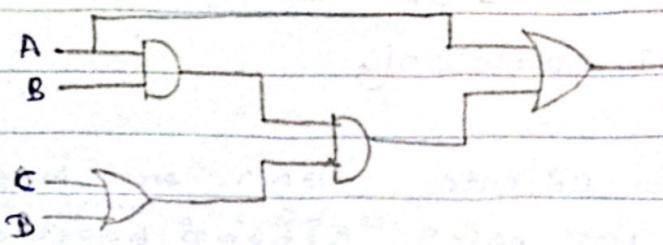
මෙයේ ලෙස නැත්තු නොවන NOT gates
OR, AND මඟ ඉදිකිරීම සහ තුළ ඇති නිෂ්පාදන
(Bubble ලෙසෙන) NAND gate නිශ්චාරු කළයායා. එහි ප්‍රතිඵලි NOT
ඉහිරි NOT gate $\overline{\overline{D}0}$ එකිනෙකුත් ඇති නිෂ්පාදන.



(Bubble පෙනීමේ ප්‍රතිඵලි)



Q9. NOR Gates only.



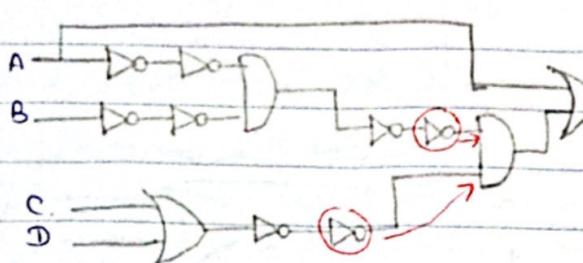
ඉහත ආකාරයට OR
විවර ලුද්ධීයෝ NOT

පෙනුව NOR වාච්‍යමය

එම්. NOT Balance සිංහ

නම NOT එකක් ලුද්ධීයන්

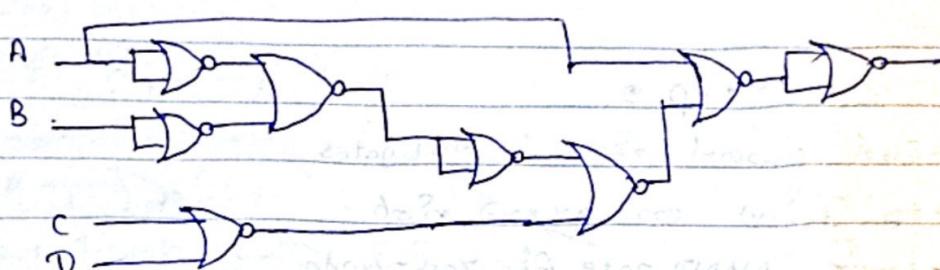
ගෙවුණා.



$\neg \neg = \neg$ AND

විවර තුළ හිරියක් NOT පෙනුව

සිංහ balance පිළිස්ස තානා
එක එකිනෙයුතු.



One's Complement.

+ අංකය \rightarrow දේශීල ආකාරයට එය අඟ සූල කිරීම
ක්‍රියාත්මක.

එහිදී carry bit ව්‍යුහයේ මෙහෙයුම් පෙනෙනු

ක්‍රියාව එම carry bit නැවත තුළ දෙන උග්‍රීත්වයි

- අංකය \rightarrow දේශීල ආකාරයට එය එහි NOT නැත ඇඟෙනය
එය සූල කිරීම (ගැනීමක්ද යොමු කිරීමෙන්).

Ex: $115 - 27$.

$$115 \rightarrow 0111\ 0011$$

$$27 \rightarrow 11011 = 00011011 \quad (115 හි 6-bit තාක්කය පෙනෙනුදී)$$

$$\begin{array}{r} 0111\ 0011 \\ + 1100\ 100 \\ \hline 1110\ 1011 \end{array} \quad \downarrow \quad \begin{array}{r} 1110\ 100 \\ + 0101\ 011 \\ \hline 1110\ 1011 \end{array} \quad \text{එම්බු තැබුණු තැබුණු නිස්සු නිස්සු. } 115 - 27 = 88$$

Second complement.

+ අංකය → එම අංකයක් ජ්‍යෙෂ්ඨ සංඛ්‍යාවල තිරිතුරුනේ.
 - අංකය → දේශීය අංකයක් හෝ එහි NOT නිස්සා.

එම NOT එකට 1 ජ්‍යෙෂ්ඨ පැහැදිලි.

Second complement මේ Carry bit උග්‍රාණය ඇත
 ගොනුදියා හේ.

Ex: 5 - 3

$$5 \rightarrow 00000101$$

$$-3 \rightarrow 00000011$$

$$\begin{array}{r}
 & & & & & \downarrow \text{Not} \\
 0 & 0 & 0 & 0 & 1 & \\
 + & 1 & 0 & 0 & 1 & \\
 \hline
 0 & 1 & 0 & 0 & 1 &
 \end{array}$$

$$1111101$$

$$+ \underline{00000101} (5)$$

$$\begin{array}{r}
 \text{00000010} \\
 \text{X} \quad \text{X} \\
 \hline
 00000010
 \end{array}$$

$$\text{Answer: } \underline{\underline{00000010}}$$

තිරිතුරු හිටුවේ දැන් නැලීම.

2^7 's complement මැලිදී තිරිතුරු හිටුවේ
 දැන් නැලීම මූල්‍ය.

දැන් එහි 0 ප්‍රතිනිශ්චිත මැලිදී තිරිතුරු + පාත.

මෙහිදී ලැබෙන, තිරිතුරු දැන් එහි පාත්‍ර තිරිතුරු නැවෙ.

අවශ්‍ය උඩු යුතු තිරිතුරු → නම් ඔවුන් නැවෙ
 ඔවෙන තිරිතුරු NOT කිරී නිස්සා ඇතුළත
 ඇත්‍ය නැවෙ.

Gray Code.

1. Convert decimal number to binary.

$$5 \rightarrow 101$$

2. එහි පරිභේද දැක්වනු කළ ගස්සා. $\rightarrow 1$

3. ඉහේ මෙත් හෝ තියුල්ලින් යෝජිත කළ ඇත්තේ

XOR අනුවත් ගස්සා.

$$1 \text{ XOR } 0 \rightarrow 1$$

$$0 \text{ XOR } 1 \rightarrow 1$$

\therefore Gray Code of 5 = 111 \leftarrow උතු කිරීම නැංවය

තෙවෙන නිපුණ්.

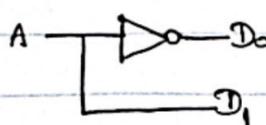
නැතුම පෙනා ජෙතු යොමු ඇති.

Decoders.

• Draw 2-to-4 line decoder using 1-to-2

2-to-4 line decoder.

1-to-2 binary decoder.



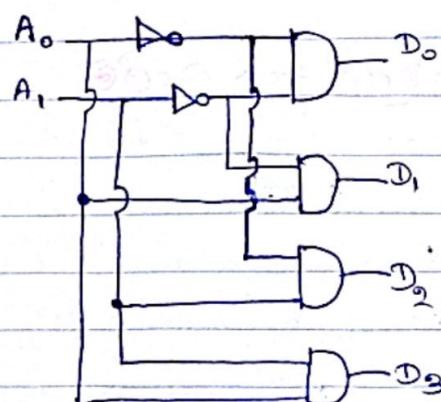
| A | D ₀ | D ₁ |
|---|----------------|----------------|
| 0 | 0 | 0 |
| 1 | 1 | 0 |
| | 1 | 1 |

① 2-to-4 තැබූ ඇතුළු පෙනු ඇත්තා එහි එහි අනුමත නියම NOT, AND සංඛ්‍යා නූත්‍ය කළ ඇති.

② ගැටුම් නියම නිවාර්තනය.

2-to-4 line decoder.

③ ප්‍රධාන.



$$D_0 = \bar{A}_0 \cdot \bar{A}_1$$

$$D_1 = \bar{A}_1 \cdot A_0$$

$$D_2 = A_0 \cdot \bar{A}_1$$

$$D_3 = A_0 \cdot A_1$$

• Draw 3-to-8 decoder Using 2-to-4.

