

Operations of Electronics: Truth Tables and State Transition Diagrams

This document explains the operations of electronics using **truth tables** for combinational logic and **state transition diagrams** for sequential logic.

1. Combinational Logic (Truth Tables)

NOT Gate

This inverts the input, for example if the input is 0 output will be 1 and vice versa

A	NOT A
0	1
1	0

AND Gate

In this, output is 1 when all inputs are 1

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

In this, output is 1, if any input is 1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

NAND Gate

This is the opposite of AND; output is 0 when all inputs are 1

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate

Opposite of OR; output is 1, only when all input are 0

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

XOR Gate

Output is, if the inputs are different

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

XNOR Gate

All outputs are 1 if the inputs are all the same

A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

2. Small Combinational Building Blocks

Half-Adder

This adds 2 bits, producing a sum and carry. It is very useful building larger adders

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full-Adder

This adds 3 bit(sum, carry-in,carry-out). It is used in cascaded addition

A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2-to-1 Multiplexer

Routes one of the several inputs to a single output, controlled by a select signal

S	A	B	Out
0	A	-	A
1	-	B	B

Decoder (2-to-4)

Converts binary input into a one-hot input

Input	Y0	Y1	Y2	Y3
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

3. Sequential Logic (Memory + Clocks)

SR Latch

Stores 1 bit; Set forces output high, Resets forces it low, Hold keeps the previous state

S	R	Q(next)	Notes
0	0	Q(prev)	Hold
0	1	0	Reset
1	0	1	Set
1	1	?	Invalid state

D Flip-Flop (on rising edge)

Transfers input (D) to output (Q) at a clock edge.

D	Q(next)
0	0
1	1

JK Flip-Flop

It is a versatile flip-flop. It can set, hold, reset or toggle based on inputs provided

J	K	Q(next)
0	0	Q(prev)
0	1	0
1	0	1
1	1	\neg Q(prev)

T Flip-Flop

Toggles the output whenever T = 1

T	Q(next)
0	Q(prev)
1	\neg Q(prev)

4. State Machine Example 1 – 2-bit Synchronous Counter (mod-4)

State Table

This 2-bit synchronous counter cycles through 4 states (00 -> 01 -> 10 -> 11 -> 00)

Each clock pulse increments the state by one in binary order

Current (Q1 Q0)	Next (Q1' Q0')	Decimal
00	01	1
01	10	2
10	11	3
11	00	0

State Transition Diagram

[00] → [01] → [10] → [11] → [00]

5. State Machine Example 2 – Sequence Detector (101) (Mealy)

State Table

This detects the bit pattern; each state represents how much of the pattern has been matched so far. When the full pattern is seen, the machine outputs 1 otherwise it continues checking.

Current	x=0 → Next,z	x=1 → Next,z
S0	S0,0	S1,0
S1	S2,0	S1,0
S2	S0,1	S1,0

State Transition Diagram

[S0] --1/0--> [S1] --0/0--> [S2] --1/1--> [S1]
 [S1] --1/0--> [S1]
 [S2] --0/0--> [S0]

(Note: a/b means input=a, output=b)