Dated:

THEORITICAL PERSPECTIVE OF COMPUTING

Theory of computation (TOC) is a branch of computer science that is concerned with how problems can be solved using algorithms and how efficiently they can be solved.

The essensce of the theory of computation is to help develop mathematical and logical models that run efficiently.

Importance:

* Writting efficient algorithms that run in computing devices.
* Programming language research and their development.
* Efficient compiler design and construction.

Types:

Autometa Theory	Computability Theory	Complexity Theory.
Is the study of abstract computational devices. It forms a formal framework for designing and analyzing computing devices as quantum/biocomputers. Basic features: * set of input symbols * configuration states * Output. Types: * Finite Autometa * Context Free Grammers (CFGs) * Turing Machines.	defines whether a problem is solvable by any abstract machine. Some problems are computable while others are not. Computation is done by various computation models depending on the nature of the problem at hand.	This theoretical branch is all about studying the cost of solving problems while focusing on resources (time and space) needed as the metric. The running time of an algorithm varies with the inputs and usually grows with size of input. Vastly used complexity theory is time complexity noted as, $T(n)$.

Dated:

Here,
we are using our Adder Circuit as an example and differiantiate it into different computational models:-

* MANUAL OPERATION:

Adding $101011101_{(2)}$ with $001110110_{(2)}$ manually;

$$
\begin{array}{r}
101011101_{(2)} \\
001110110_{(2)} \\
\hline
1110100011_{(2)}
\end{array}
$$

* MACHINE OPERATION:

Input =	10	01	11	10	11	01	11	00	10
Output =	1	1	0	0	1	0	1	1	1
state =	S_0 (starting state)	S_0	S_1	S_1	S_1	S_1	S_1	S_0	S_0

* FINITE STATE TABLE:

State \ Input	00	01	10	11
$S0$	$0/S0$	$1/S0$	$1/S0$	$0/S_1$
S_1	$1/S0$	$0/S01$	$0/S_1$	$1/S_1$

Scanned with CamScanner

Dated:

* MATHEMATICAL MODEL

Basic Terms needed for Mathematical model:

$$\{S, I, O, f(x), g(x), S_{start}\}$$

where;

S :- States

I :- Inputs

O :- Outputs

f(x) :- state transition function

g(x) :- Output function

S_{start} :- starting state.

Hence;

$S = \{S_0, S_1\}$, $I = \{00, 01, 10, 11\}$, $O = \{1, 0\}$, $S_{start}\ \{S_0\}$

for f(x) :- $\therefore$ next state = f(Input, current state)

$f(00, S_0) = S_0$
$f(01, S_0) = S_0$
$f(10, S_0) = S_0$
$f(11, S_0) = S_1$

$f(00, S_1) = S_0$
$f(01, S_1) = S_1$
$f(10, S_1) = S_1$
$f(11, S_1) = S_1$

for g(x) :- $\therefore$ output = g(constant state / input)

$g(00, S_0) = 0$
$g(01, S_0) = 1$
$g(10, S_0) = 1$
$g(11, S_0) = 0$

$g(00, S_1) = 1$
$g(01, S_1) = 0$
$g(10, S_1) = 0$
$g(11, S_1) = 1$

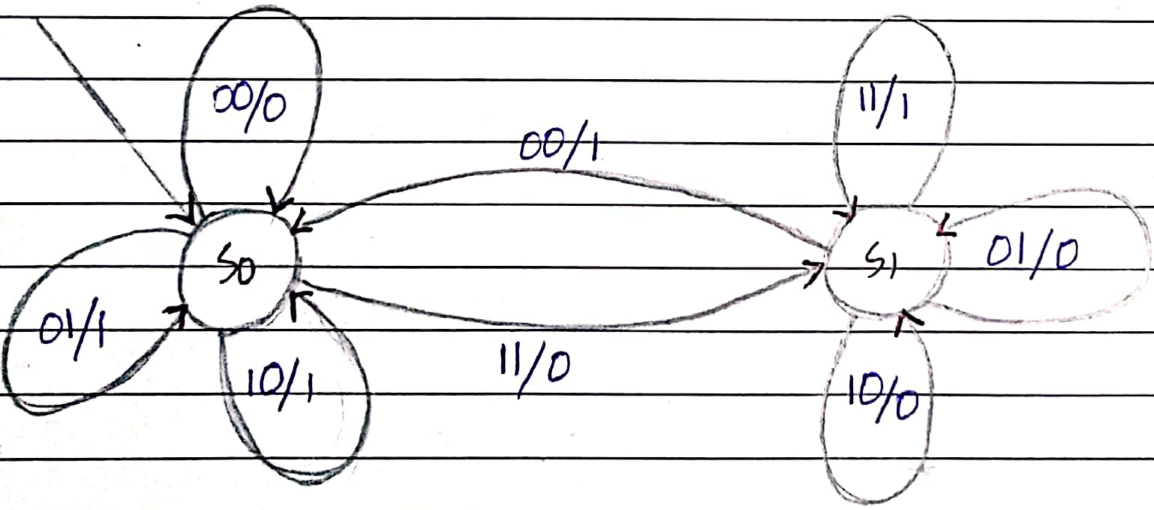
Dated:

*FINITE STATE DIAGRAM:

firstly there is two state in which carry is 0 and in other 1. So and S_1 respectively. Then we draw all possible outcomes and inputs (Truth table).

So carry=0	S1 carry = 1
00 / 0	00 / 1
01 / 1	01 / 0
10 / 1	10 / 0
11 / 0	11 / 1

Hence;



* Node: one state of machine

* Arc is defining the state changing of machine.

* If machine changes state we direct it with arrow Line.

* writting technique: input/output

* Starting state always get an arrow attached to nothing.

Ps: Mujhe pata hai bht buri bni hai :)

Dated:

We can also give solution with these models following:-

logic circuit

* Schematic
* Block

Programming Language code.

Algorithm.

Pseudo Code

Natural link description

flowchart and many other ways.