Algorithm State Machine (ASM)

→ The binary information stored in a digital system can be classified as either data or control information.

→ Data are discrete elements of information that are manipulated to perform arithmetic, logic, shift and other similar data processing tasks. These operations are implemented with digital components such as adders, multiplexer, counters and shift registers.

→ Control information provides command signals that supervises the various operations in the data section in order to accomplish the desired data processing tasks.

→ Fig 1. shows the relationship between the control logic & data processing in a digital system.

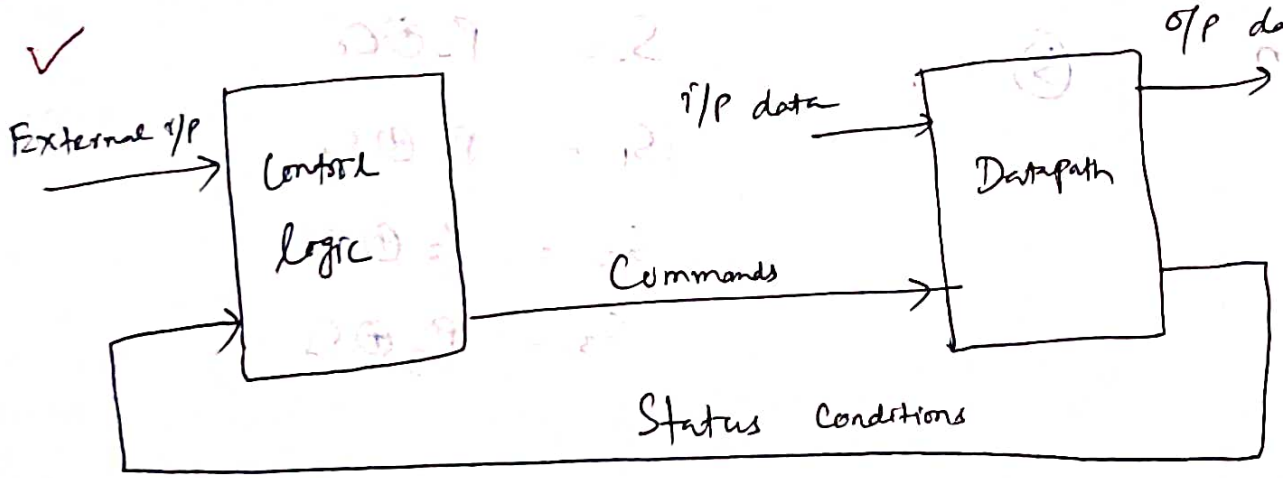


Fig 1:- Interaction between control logic & datapath

The data processing path, commonly referred to as the datapath, manipulates data in registers, according to the system's requirements.

The control logic initiates a sequence of commands to the datapath. The control logic uses status conditions from the datapath to serve as decision variables for determining the sequence of control signals.

To design a digital system, we have to design 2 subsystem - datapath subsystem and control subsystem. A datapath subsystem consists of digital circuits which are required to perform specified operations on the data information.

A control subsystem is a sequential circuit which issues commands for the subsystem. The control considers the status condition & other external i/p's to determine the next state to initiate other operations.

A state machine is another term for a sequential circuit which is the basic structure of a digital system.

Just as a flowchart serves as a useful aid in writing software programs, algorithm state machine (ASM) charts help in the hardware design of digital system.

A Conventional flow chart describes the sequence of procedural steps and decision paths for an algorithm without any concern for the time relationship.

The ASM chart on the other hand describes the sequence of events as well as the timing relationship between the states of a sequence controller and the events that occure while going from one state to next state after each clock edge

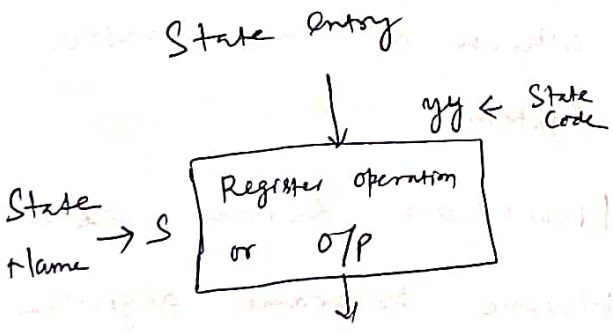
Components of ASM Chart:-

There are 3 Components of ASM Chart -

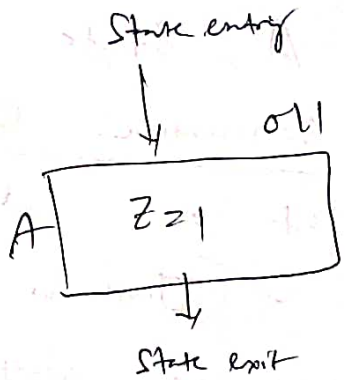
1) State Box 2) Decision Box 3) Conditional o/p Box

1) State Box

A state of a clocked sequential circuit is represented by a rectangle called State Box. It is equivalent to a node in the state diagram or a row in a state table



(a) General description

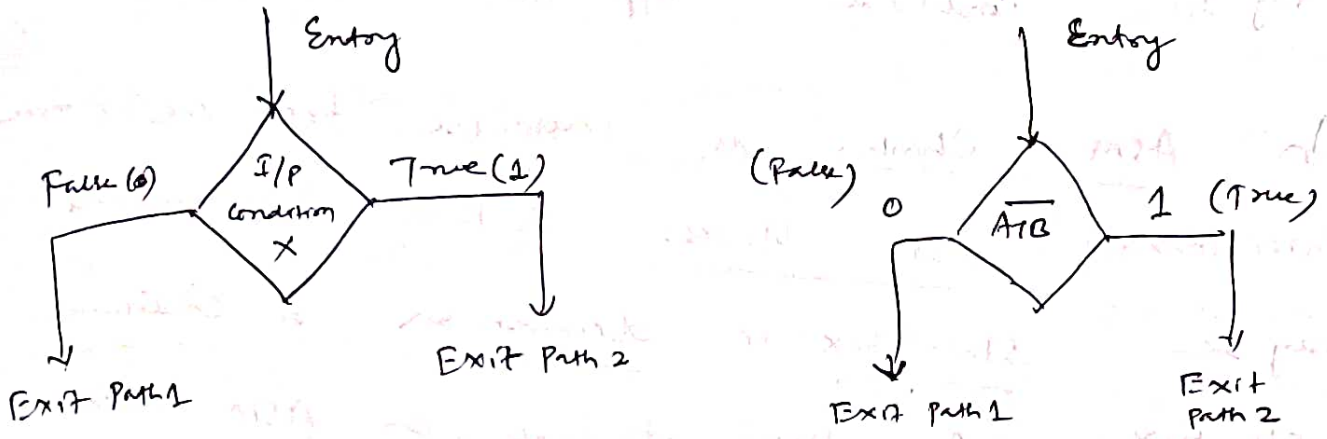


(b) Specified Example

Fig 2: State Box

2) Decision Box

The decision box or conditional box is represented by a diamond-shaped symbol with one i/p and two or more o/p paths.



(a) General Description (b) Specific example

Fig 3:- Decision Box

For ex:- 'X' is written inside the box indicates that the decision is based on the values of X, where as $\overline{A+B}$ written inside the box indicates that the true path is chosen if $\overline{A+B}=1$ and the false path is chosen otherwise

3) Conditional o/p Box

The Conditional o/p box is represented by a rectangle with rounded corners or by an oval with one i/p line and one o/p line. The o/ps that depend on both the state of the system & i/ps are indicated inside the box. These are Mealy type o/ps.

Fig 4:- Conditional O/P Box.

An ASM Chart is constructed from one or more interconnected ASM blocks. So these ASM blocks may be state box or decision box or Conditional o/p box. A path through an ASM block from entry to exit is referred to as link path.

Salient Features of ASM Chart :-

1) An ASM Chart describes the sequence of events as well as the timing relationship between the states of a sequential controller and the events that occur while going from one state to the next.

2) An ASM Chart contains one or more interconnected ASM blocks.

3) Each ASM block contains exactly one state box together with the decision boxes and Conditional o/p boxes associated with that state.

4) Every block in an ASM Chart specifies the operations that are to be performed during one common clock pulse.

5) An ASM block has exactly one entrance path and one or more exit paths represented by the structure of the decision boxes.

6) A path through an ASM block from entrance to exit is referred to as a link path.

7) The operations specified within the state and conditional o/p boxes in the block are performed on the datapath subsystem.

8) Internal feedback within an ASM block is not permitted. Even so, following a decision box or conditional o/p boxes, the machine may reenter the same state.

9) Each block in the ASM Chart describes the state of the system during one clock pulse interval.

When a digital system enters the state associated with a given ASM block, the o/p's indicated within the state box become true. The conditions associated with the decision boxes are evaluated to determine which path or paths to be followed to enter the next ASM block.

Some examples of ASM Charts:-

1) Mod-6 Counter

The state diagram of a Mod-6 counter is shown in Fig. 5.(a). ASM Chart equivalent of Figs.(a) is shown

In Fig 5.(b). The states are now represented by state boxes instead of nodes.

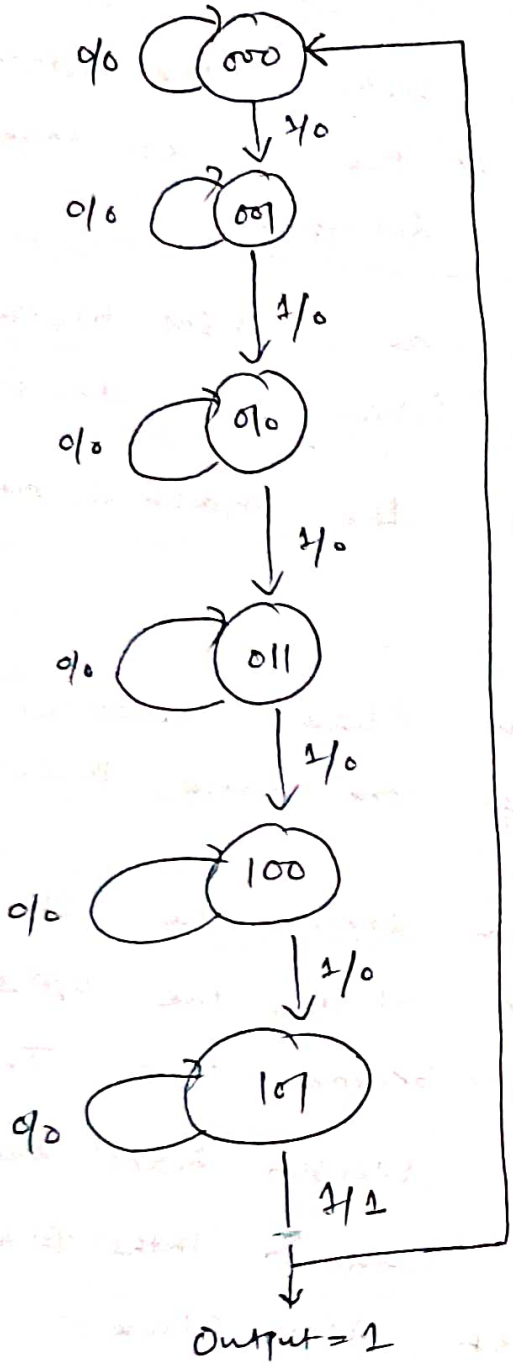
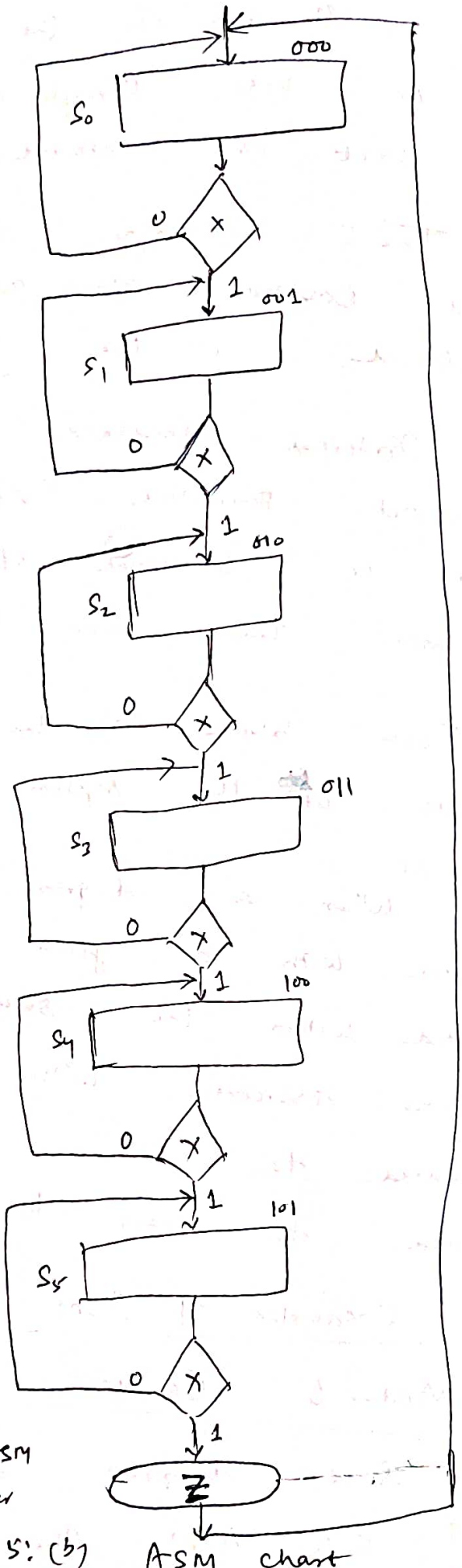


Fig 5 (a):- State diagram



5: (b) ASM chart

Fig 5:- State diagram & ASM chart for Mod-6 Counter

The transitions are still represented by arrows but the I/Ps indicated adjacent to arrows in the state diagram are replaced by decision boxes with true and false branches.

In state diagram the O/Ps are indicated along with I/Ps using a separator slash or comma. In ASM chart are now indicated in conditional O/P box.

The ASM chart has 6 state boxes named from S_0 to S_5. For every clock pulse, X is sensed. If X=1, the machine goes to next state; if X=0, then the machine remains in the same state as indicated by a return branch. All these happens in the same clock cycle.

When the machine is in state S_5, on the occurance of the clock pulse, if X=1, the machine produces an O/P pulse indicated as 'Z' and goes to the initial state S_0. The states are assigned binary numbers (ex: 100) using three F/Fs.

Let's design it using D F/Fs. With D F/Fs, the excitation D_i has to be same as next state variable Y_i. [Note:- In 'D' F/F O/P is same as I/P to D]

Observing only state assignment indicated 248 in ASM chart, we notice that the next value Y_0 has to become 1 for the present states Y_2, Y_1, Y_0 = 000, 010, 100 only.

Note - For present state → Next state

Present State	Next State	
000 →	001	Here last bit is changing from 0 to 1. So Y_0 becomes = 1.
010 →	011	
100 →	101	

Hence using decimal codes & remembering that states 6 (110) & 7 (111) never occur, we get

$$D_0 = Y_0 = X \cdot \left[\sum m(0, 2, 4) + d(6, 7) \right]$$

→ multiplied.

Note: D → FIC → Y_0 & X is multiplied or ANDed

$X = 0$, D_0 & $Y_0 = 0$

$X = 1$, then $Y_0 = \left[\sum m(0, 2, 4) + d(6, 7) \right]$

Similarly

$$D_1 = Y_1 = X \cdot \left[\sum m(1, 2) + d(6, 7) \right]$$

Note

Present State	Next State	
001 →	010	
010 →	011	(Y_0 has to 1 & Y_1 should be remained 1)

Similarly

$$D_2 = Y_2 = X\left[\Sigma m(3,4) + d(6,7)\right]$$

because

P.S	N.S
011	100
100	101

Similarly

$$Z = X.\left[\Sigma m(5) + d(6,7)\right].$$

because the state is 101 or 5 and

$X = 1$ the $O/P = 1$.

(Z)

Notice that the I/P X is ANDed with each of expressions for the excitations. If the I/P X is exclusively provided, then the circuit counts the number of clock pulses in the duration when X is at level 1. In other words 'X' enables the Counter.

Ex:-2 Sequence Detector

Consider the example we have done earlier :- 1010 sequence detector with overlapping permitted.

The state diagram of the sequence detector is shown in Fig 6.(a). The ASM Chart of the sequence detector is shown in Fig. 6 (b).

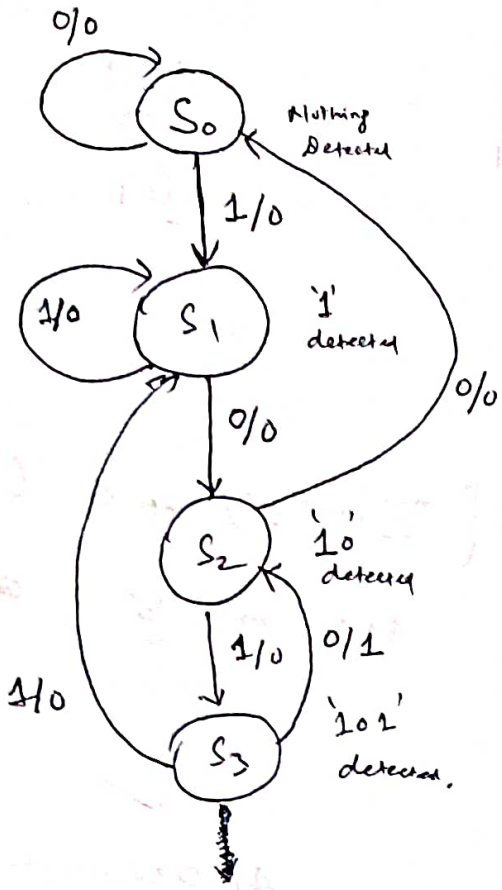
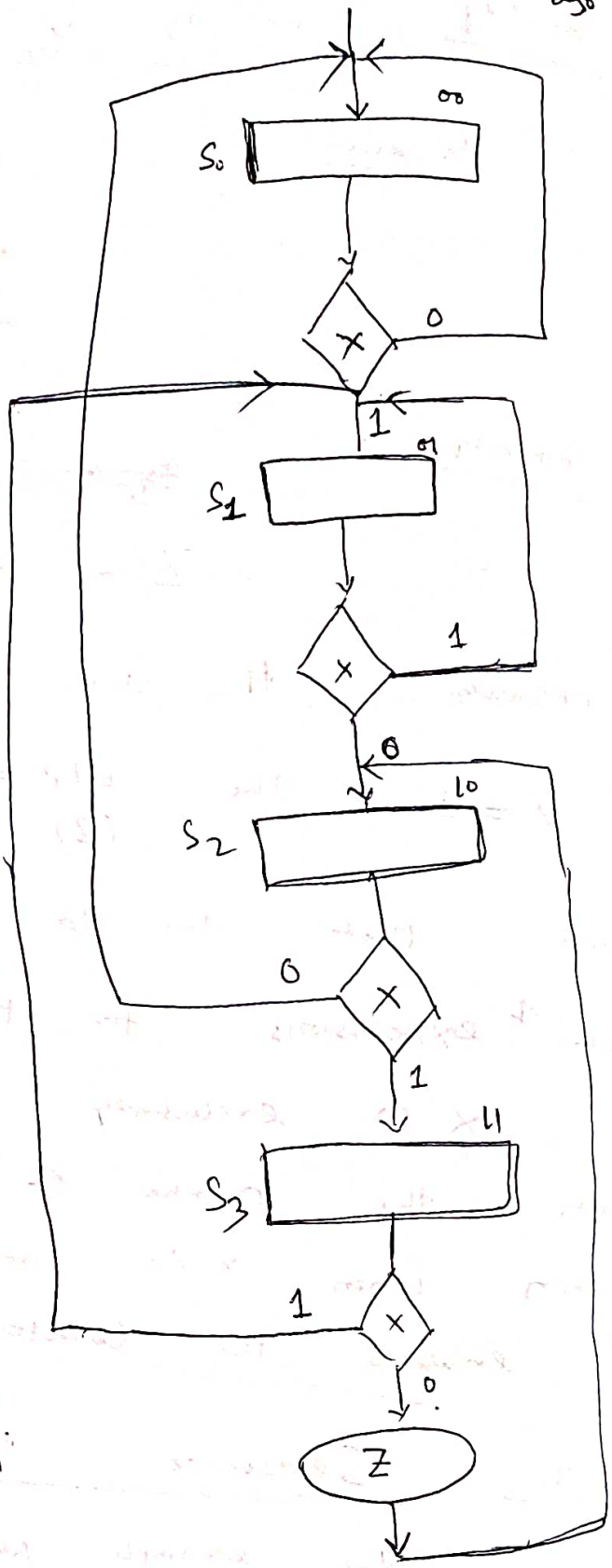


Fig 6(a) State diagram



6(b) ASM Chart

The machine has 4 states. So, 4 state boxes are required. While at each state, the machine has to decide to which next state it has to go when the i/p 0 or 1 is given. Four decision boxes with two branches each, are also required. It has to O/P a '1' once the sequence 1010 is detected. So it requires a conditional O/P box. The state assignment is arbitrary i.e. Let it be

$S_0 = 00$, $S_1 = 01$, $S_2 = 10$, $S_3 = 11$.

There are 4 states and $y_1 y_0$ are the binary values assigned to the states. If we decide to use two D F/Fs, the excitation table is identical to the transition table and $D_i = Y_i$.

In this ASM Chart, note that the next state variable Y_0 becomes 1 in states S_1 (01) & S_3 (11). The corresponding present states are S_0 ($y_1 y_0 = 00$) & S_2 ($y_1 y_0 = 10$) and transition occurs if $X = 1$.

Note :- i.e

P. State			N. State	
(S_0)	00	$\xrightarrow{1}$	01	(S_1)
S_2	10	$\xrightarrow{1}$	11	(S_3)

Thus, we may write the expression for excitations by inspection as follows.

$$D_0 = Y_0 = X \bar{y}_1 \bar{y}_0 + X y_1 \bar{y}_0 = X \bar{y}_0 (\bar{y}_1 + y_1)$$

$$\therefore D_0 = X \bar{y}_0 \quad \text{---(1)}$$

From ASM Chart we observe that Y_1 becomes 1 for S_2 on $X = 0$ from S_1, on $X = 1$ from S_2.

Similarly $Y_1 = 1$, for S_3 i.e

(S_1) 01 $\xrightarrow{X=0}$ 10 (S_2)

10 (S_2) $\xrightarrow{X=1}$ 11 (S_3)

$\therefore D_1 = Y_1 = \bar{x}\, \bar{y}_1 y_0 + x y_1 \bar{y}_0$

The o/p $z = 1$ when i/p $x = 0$ & Present state $y_1 y_0 = 11$. Therefore,

$$z = \bar{x}\, y_1 y_0 .$$

A logic circuit can be drawn based on the above expressions.

ASM for Binary Multiplier :-

The binary multiplier is designed using the add shift algorithm. The manual working is given in Fig 7.

Fig 7: Manual multiplication algorithm

```
                1 1 0 1   ← multiplicand (13)10
                1 0 1 0   ← multiplier (10)10
              -----------
              0 0 0 0     ← partial product 1
            1 1 0 1       ←  "      "    2
          0 0 0 0         ←  "      "    3
        1 1 0 1           ←  "      "    4
      -----------------
      1 0 0 0 0 0 1 0     ← product (130)10
```

→ The product of two n digit numbers can be a number of '2n' digits atmost.

Digital implementation requires the following changes.

1) In manual working, we perform left shift on the subsequent partial product which is yet

to be formed, but this kind of anticipatory job is not done by a physically realizable machine. A real machine can operate only on the existing operands but not on future results. Hence, we shift the partial product already formed to the right by one bit and add the next partial product in its normal position. This would produce the correct result as the reality positions of the operands for additions are as they should be.

2) Instead of forming all the partial products and then adding, which would require a large number of registers to store them, each partial product is added to a register A (Accumulator) and shifted right. This job is repeated n times where n is the number of bits on the multiplier.

Data Subsystem for Binary Multiplier:-

Fig. '8' shows the datapath subsystem for the binary multiplier. It comprises the following

1. Register B to hold the multiplicand
2. Register Q to hold the multiplier.
3. Register A, called Accumulator, to hold the cummulative sum of the products.

4. A parallel adder circuit

5. One F/F 'C' to hold the Carry, if any produced in addition

6. A Counter 'p' which is initialized to word length 'n' (i.e no. of digits in the multiplier)

7. Provision to decrement p to check for zero.

8. Provision to concatenate the registers C, A, Q to form the Combined register of length $1+n+n = 2n+1$ bits with facility to shift right.

→ Fig 9. illustrates how a digital computer performs multiplication using add-shift algorithm. Consider the example (1101×1010).

Initially the multiplicand is ~~loaded~~ loaded into register B and multiplier is loaded into register Q. C & A are cleared i.e 0 & 0. The word length 'n' is loaded into register P which acts as a Counter.

The least significant bit Q_0 of the multiplier Q is sensed by the machine.

Cont-I If $Q_0 = 1$, then form the partial product

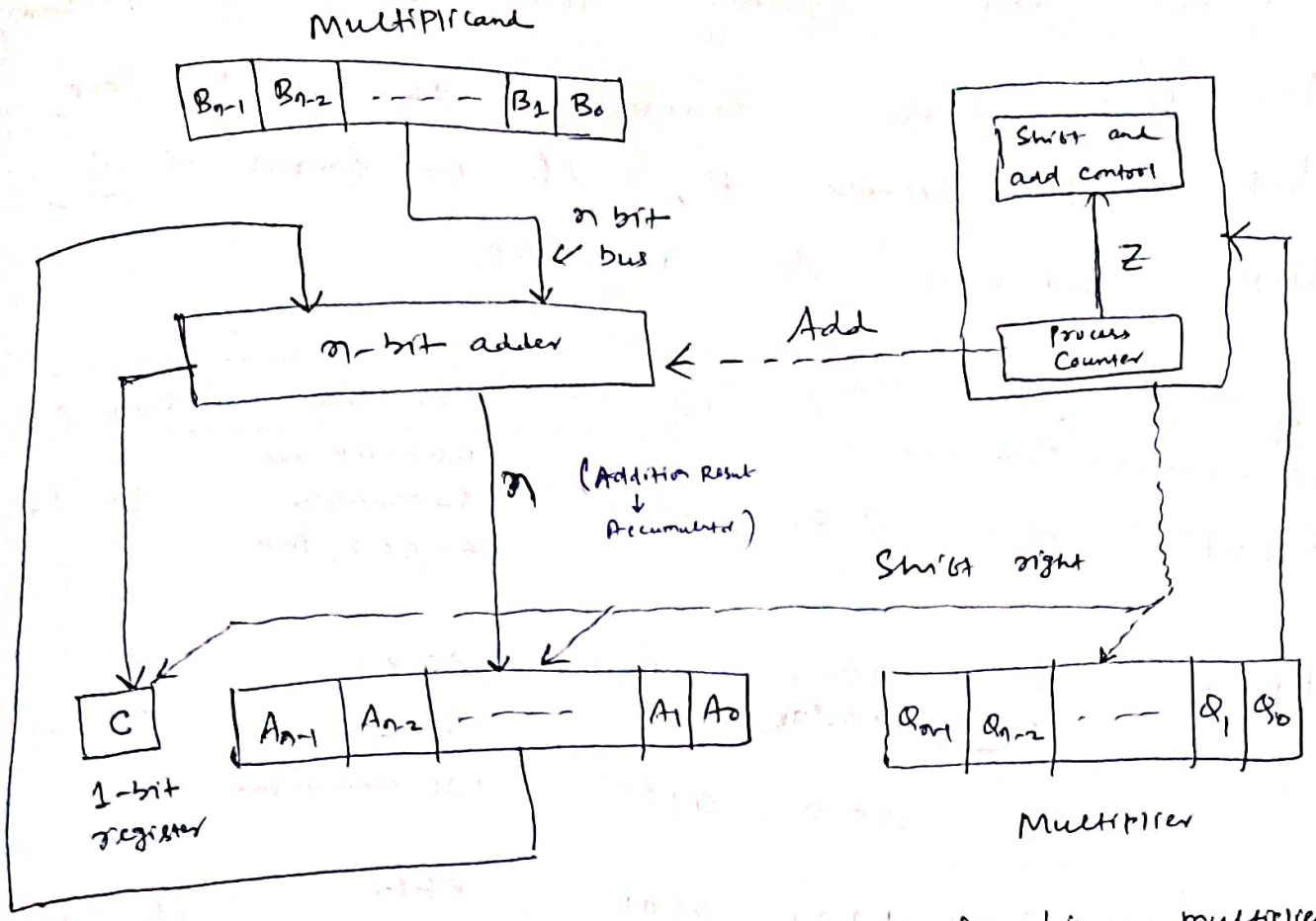


Fig 8:- Datapath subsystem for binary multiplier

{by adding B to A}, that is A ← A+B.
then store it in A

and then shift the combined register CAQ to the right by 1 bit position so that 'C' goes to 'A_{n-1}' & 'A_0' goes to 'Q_{n-1}' & Q_0 goes out. In place of Q_0, previous Q_1 becomes Q_0 for the next iteration.

Case-II

If Q_0 i.e LSB of multiplier Q is '0', then the partial product is 0 and hence there is no need to add, so only shift CAQ to the right. Repeat the process 'n' times

in a loop starting from (n-1) and proceed to '0'. After traversing through the loop, test the counter P, if it shows as 0 stop and exit from the loop.

B	C	A	Q	Comments	Count P
1101	0	0000	1010	B ← Multiplicand Q → Multiplier A ← 0, C ← 0, P ← n	100 (4)
1101	0	0000	1010	P ← P-1 $Q_0 = 0$	011 (3)
	0	0000	0101	CAQ shift right	
1101	0	1101	0101	P ← P-1 $Q_0 = 1$ A ← A+B CAQ shift right	010 (2)
	0	0110	1010		
1101	0	0110	1010	P ← P-1 $Q_0 = 0$ CAQ shift right	001 (1)
	0	0011	0101		
1101	1	0000	0101	+ 1101 P ← P-1 $Q_0 = 1$, A ← A+B CAQ shift right	000 (0)
	0	1000	0010		

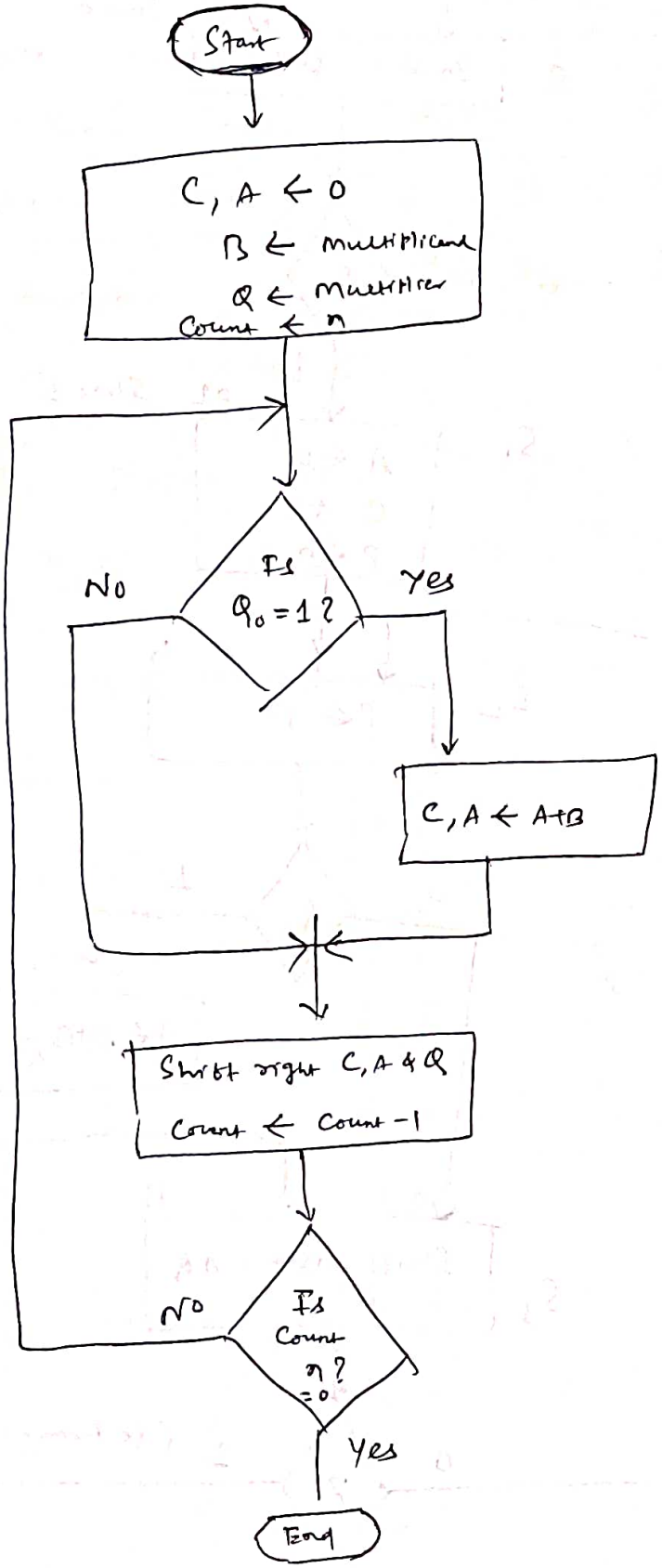
Ans ✓

Fig 9:- Flow chart for multiplication of 1101 & 1010.

∴ Product is $0\ 1\ 000\ 0010 = (130)_{10}$

Flow Chart for Binary Multiplier

Prg 10. Show the ASM Chart for binary multiplier. The explanation is same as discussed just before.



During
A+B, addition
Sum → A+B → A
Carry → C

Fig 10:- Flow Chart for multiplication operation

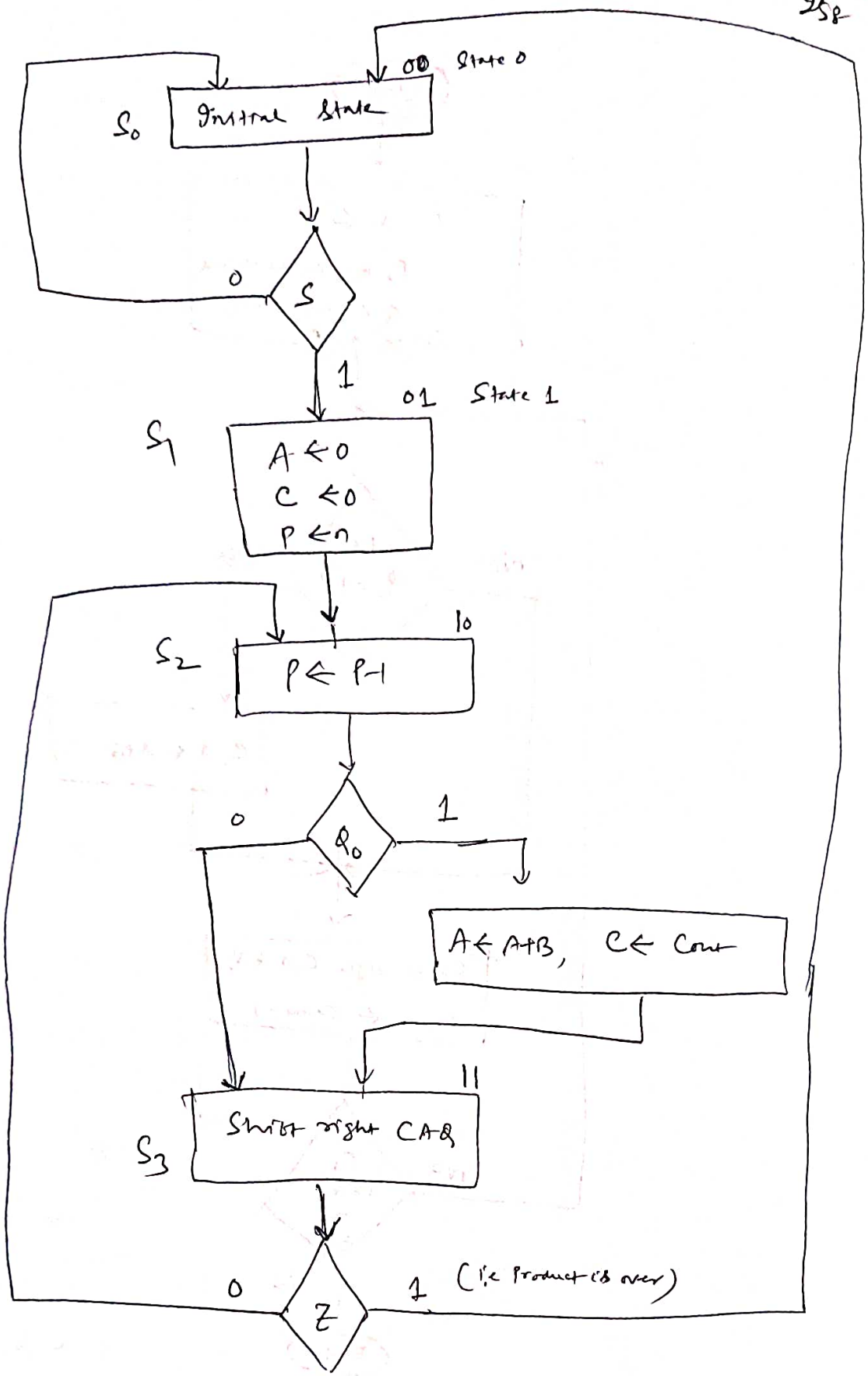


fig 11:- ASM Chart for binary multiplier

Control Subsystem Using logic gates :-

From the ASM Chart, shown in Figure 11, we notice that the binary multiplier has 4 states and three I/Ps. The state table for the Control Subsystem of a binary multiplier is shown in Table 1.

Look at, the present state PS ($y_1 y_0$) & next state NS ($Y_1 Y_0$) Columns first. [Refer the ASM chart & design the table]

Table 1:- State Table for the Control Subsystem

PS	I/Ps	NS	O/Ps (Timing Signals)				
y_1 y_0	S Q_0 Z	Y_1 Y_0	T_0	T_1	T_2	T_3	Q_0T_2
00	0 — —	0 0 — —	1				
00	1 — —	0 1 — —		1			
01	— — —	1 0 — —			1		
10	— 0 —	1 1 — —				1	
10	— 1 0	1 1 — —				1	1
11	— — 0	1 0 — —					1
11	— — 1	0 0 — —					1

Notice in the ASM Chart that there are two transitions from state 00. They are 00 → 00 & 00 → 01, when S=0 & S=1 respectively.

Other I/Ps for these transitions become don't care entries, shown by dashes.

For P.S '01', there is only one transition to state '10' in the ASM chart. Hence, only one row is provided. This transition has no I/P conditions and hence the columns of S, Q_0, Z become don't cares indicated by dashes.

For P.S '10', the transition to state '11' occurs in 2 paths depending on Q_0. If $Q_0 = 0$, then the machine goes straight to state '11'. If $Q_0 = 1$, then the machine has to add B to A and go to state 11.

For P.S of '11', there are two possible transitions. If Z = 0 (i.e multiplication is not complete) goes to 10 state & decrement counter P. If Z = 1, (i.e multiplication is over), it goes to initial state '00'.

Now, for the O/P columns, T_0 is equal to 1 for P.S '00', Similarly T_1 is 1 for PS '01', T_2 is 1 for PS '10', T_3 is '1' for PS 11.

Finally, the O/P $Q_0 \cdot T_2$ is for the purpose of enabling addition in state 10.

All these o/ps eventually provide the control timing signals.

The control unit needs two F/Fs. If we choose D F/Fs, it is easy to get the following expressions for the excitations as functions of state variables $\underline{y_1, y_0}$ and the i/ps $\underline{S, Q_0, Z}$.

[Check the expression in table where $Y_1 = 1$]

$$D_1 = Y_1 = \bar{y}_1 y_0 + y_1 \bar{y}_0 \bar{Q}_0 + y_1 \bar{y}_0 Q_0 + y_1 y_0 \bar{Z}$$

$$\boxed{D_1 = \bar{y}_1 y_0 + y_1 \bar{y}_0 + y_1 y_0 \bar{Z}}$$

Taking common

$$\left[\because y_1 \bar{y}_0 (\bar{Q}_0 + Q_0) = y_1 \bar{y}_0\right]$$

$$D_0 = Y_0 = \bar{y}_1 \bar{y}_0 S + y_1 \bar{y}_0 \bar{Q}_0 + y_1 \bar{y}_0 Q_0$$

$$\boxed{D_0 = y_1 \bar{y}_0 S + y_1 \bar{y}_0}$$

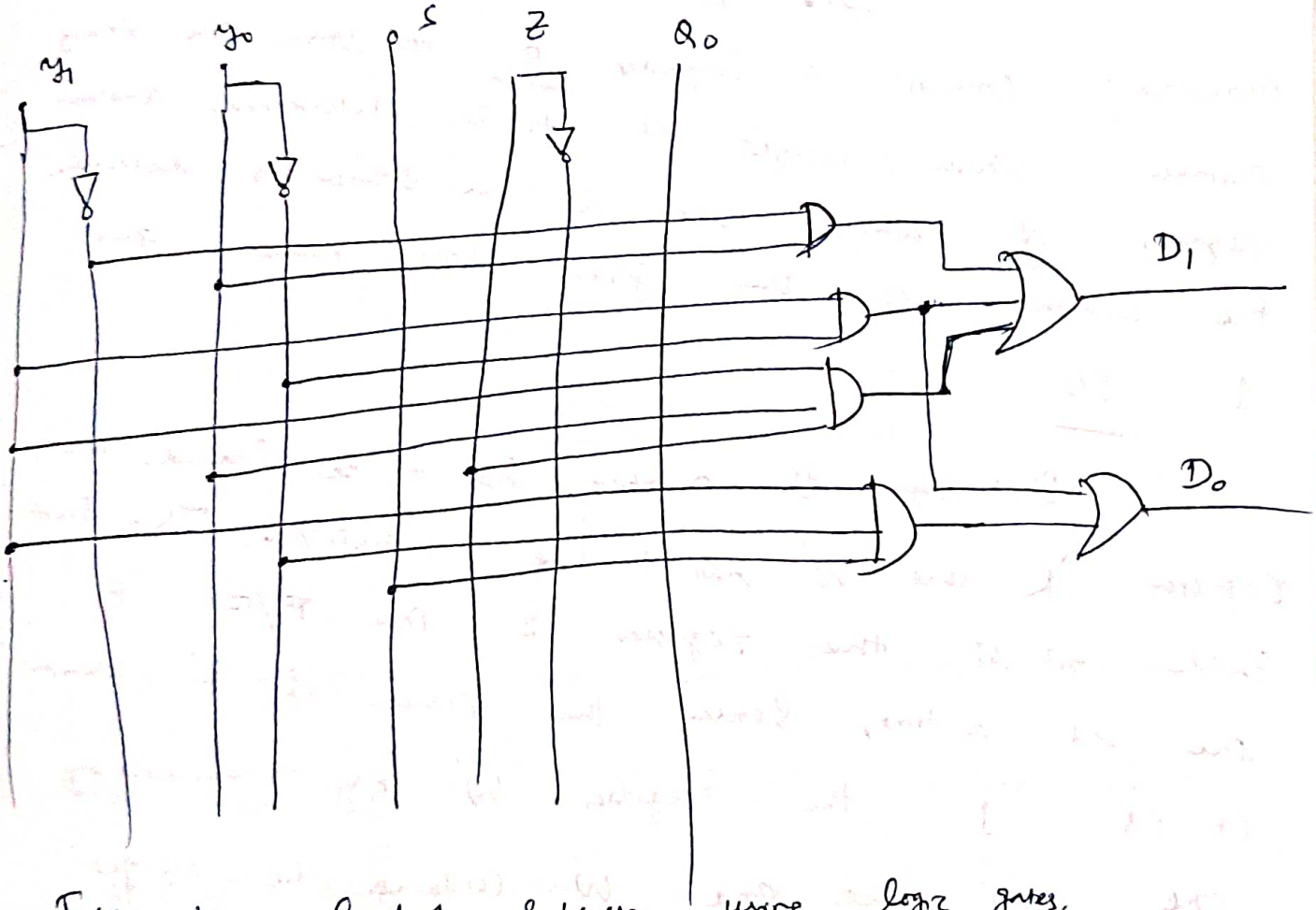


Fig:- 12 Control subsystem using logic gates.

Note:- The IP Q0 nearly enables 2 bit adder in State 10 & hence it doesn't participate in excitation functions. It has to produce the control signal $Q_0 T_2$ to enable the adder.

ASM for Weighing Machine

The weight of a binary number is defined as the number of '1's present in its binary representation.

Here we want to design a sequential machine which will calculate the weight of a given binary number or in other words compute the number of 1s in a given binary number

Let the digital system (weighing Machine) contain a register R to store the binary number whose weight is to be determined, another register 'W' which acts as a counter to determine the weight of the given binary number and 1 F/F.

Initially the number has to be loaded into register 'R' and 'W' has to be initialized. Then shift each bit of the register R into F/F 'F', one at a time, Sense the value of 'F'; whenever it is '1', the register 'W' is incremented by one. At the end 'W' contains the weight of the word.

The ASM Chart for the weighing machine is shown in Figure 13.

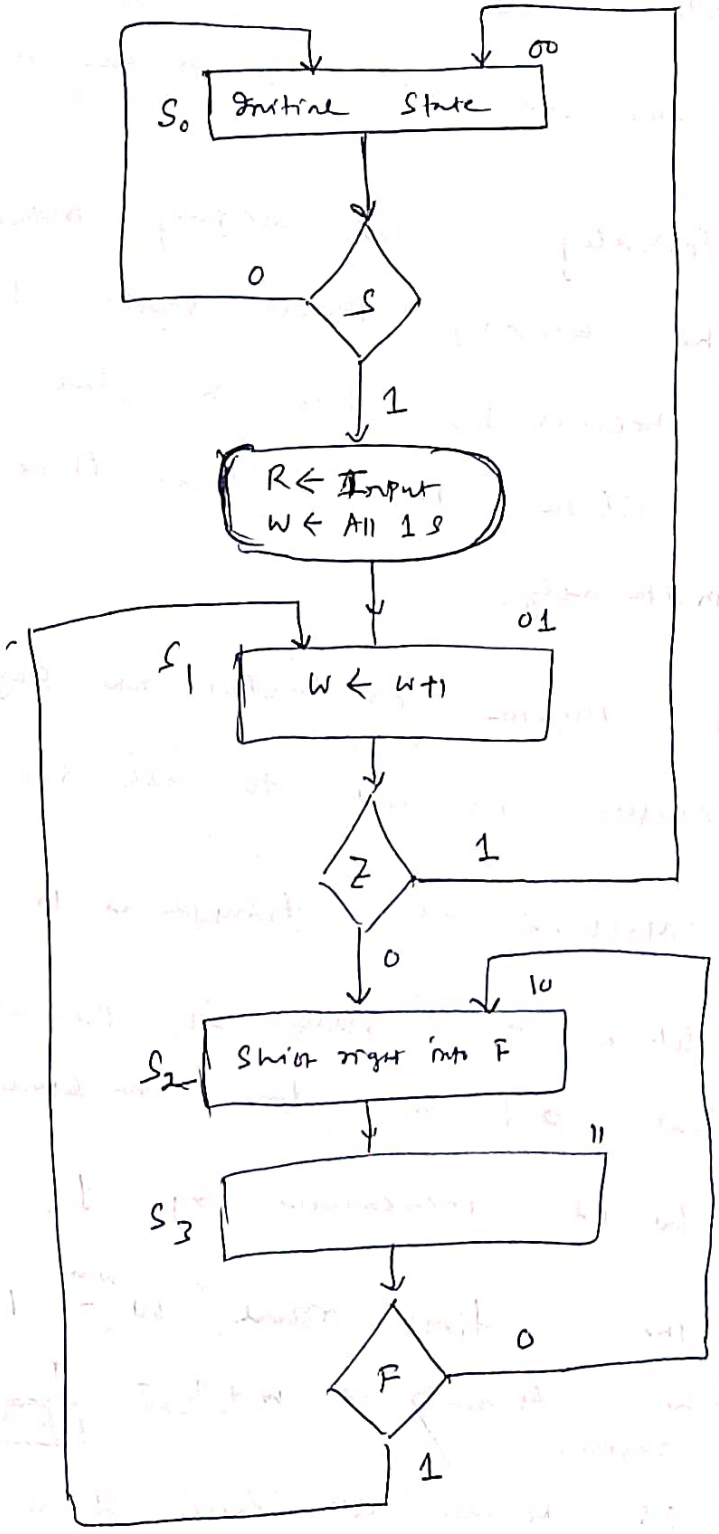


Fig 13:- ASM Chart for weighing machine

→ It has 3 i/ps S (Start), Z (Zero) and F (Flip Flop) and 4 states S_0, S_1, S_2 and S_3. S is the start i/p (S=1), starts the weight computation process), Z

is for sensing all zeros in R (Z=1 indicates all zero in R,) and the value of F decides whether W is to be incremented or not (F = 1 indicates that W has to be incremented)

State S_0:- Initially the weighing machine is on State S_0. The weighing process starts when Start (S) signal becomes 1, while in State S_0, if S = 1, the clock pulse causes three jobs to be done simultaneously.

1. Binary number is loaded into Register R
2. W register is set to all 1s.
3. The Machine is transferred to State S_1.

State S1:- While in State S1, the clock pulse causes two jobs to be done simultaneously.

1. Counter W is incremented by 1.

Note:- For the first round, $W^{Was} = 1111$ (Let's say) (Let's say W is 4 bit register.) and $W + 1 =$ $\begin{array}{r} 1111 \\ +\quad\ 1 \\ \hline 1\,0000 \end{array}$

So all 1s becomes all Zeros. W is resetted to 0000.

2. If Z = 0, the machine go to the State S2;
   if Z = 1, the machine goes to State S_0.
   (i.e all shifting in R is completed & R becomes 0; Weight of word indicated by W, and machine go back to initial state)

Z=0 ; means 'R' is not zero ; right 265
if not Complete a still some weights are there.
So it goes to next state S_2 to do the further shifting.

State S2 :-

on this state, Register R is shifted right by 1 bit so that LSB goes to R and MSB is loaded with 0.

State S3 :-

On this state, the value of R is checked. If it is '0', machine is transferred to the state S_2. If it is '1', the machine is transferred to state S_1. and W is incremented by 1.

System Design :-

The system (Machine) consists of 2 subsystems :- data processor subsystem and control subsystem. The data processor subsystem performs the task prescribed for each state of the ASM block The control subsystem ensures the proper sequence of states and transitions depending on the I/Ps and the present state of the machine.

Datapath Subsystem :-

Fig 14. shows the datapath subsystem for the weighing machine. It shows the registers, F/F's, Control Subsystem ckt for zero checking and connections of the control signals which controls the operation of these.

The control subsystem produces the signals T_0, T_1, T_2 and T_3 when the machine is on state S_0, S_1, S_2 & S_3 respectively. (Ex:- $T_0 = 1$ when the machine is on the state S_0; $T_1 = 1$ when the machine is in the state S_2 & S_0 on). These signals are used to activate desired operations on their respective states.

Initially, the machine will be on state S_0, i.e $T_0 = 1$. For starting the process, make the i/p S equal to 1. The clock pulse occurring while in S_0 loads the i/p word parallelly into register R and sets the register W to all 1s at the same time. The same pulse takes the machine to state S_1 and T_1 becomes 1.

While in the state S_1, the clock pulse increments W and makes it 0 to begin with. The same pulse senses Z. If Z=1, then it would mean that the word in R contains all 0s and so the same clock pulse takes

(∵ ex:-
$$\begin{array}{r} 1111 \\ +\quad\ 1 \\ \hline 1\,0000 \end{array}$$
)

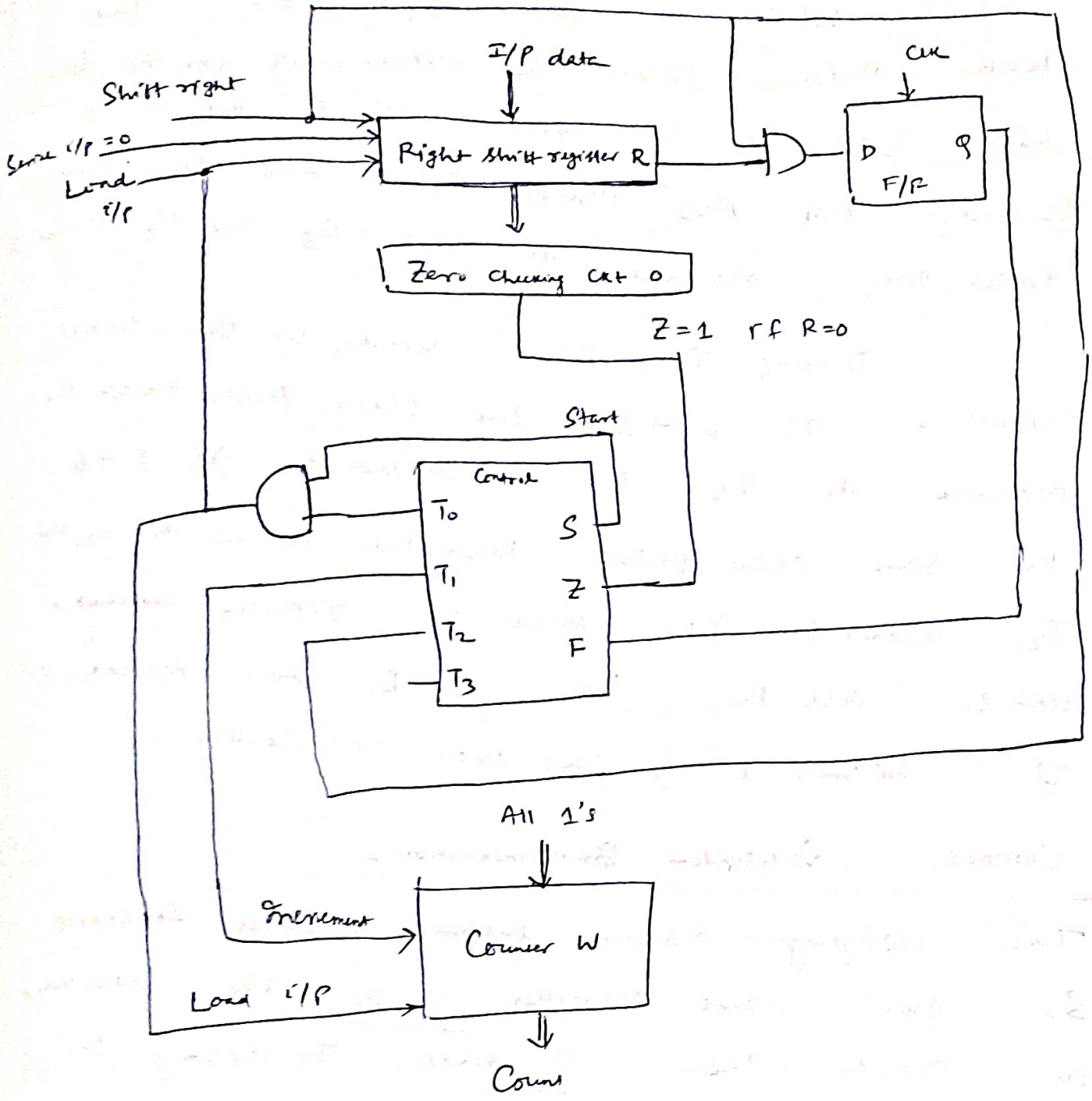


Fig 14:- Datapath Subsystem for weighing machine

the machine to the initial state 'S0' as the weight of the given word already indicated by W. and process ends.

If Z is sensed '0' during T_1, then it means that binary word contains one or more 1s in it. So the process has to continue and the Control ensures that the same clock pulse takes the machine to state S_2 and T_2 becomes 1.

While in state S_2, ($T_2 = 1$), the clock pulse shifts 'R' right by one bit so that LSB of R appears in F and '0' is shifted into MSB place. The same clock pulse takes the machine to state S_3 and T_3 becomes 1.

During T_3, F is sensed by the control circuit. If $F = 1$, the clock pulse takes the machine to S_1 and T_1 becomes 1. If $F = 0$, the same clock pulse takes the machine to S_2 and T_2 becomes 1. The process is repeated until all the bits of R are sensed or Z becomes 1, whichever is earlier.

Control Subsystem Implementation

The weighing machine system requires 4 states. So two state variables y_1, y_0 are required to provide these 4 states. By decoding the state variables, one can obtain the control timing signals T_0, T_1, T_2 & T_3.

Let Y_1 & Y_0 be the next state variables which depends on the I/Ps S, Z, F and present state variables y_1, y_0. Let us use D F/Fs. Then the excitations D_1 & D_0 are $D_1 = Y_1$ and $D_0 = Y_0$. To design the control subsystem, we have to obtain minimal expression for D_1 & D_0

in terms of y_1, y_0, S, Z, F.

Table 2:- State Table for Weighing Machine

Present State	I/Ps			N. State		F/F I/Ps	
						$D_1 = Y_1$	$D_0 = Y_0$
y_1	y_0	S	Z	F	Y_1	Y_0	
0	0	0	–	–	0	0	0
		1	–	–	0	1	S
0	1	–	1	–	0	0	$\bar{Z}$
		–	0	–	1	0	0
1	0	–	–	–	1	1	1
1	1	–	–	1	0	1	$\bar{F}$
				0	1	0	F

Row 0: Row 0 shows that in the next state, the O/P of (F/F 1) is always 0, for any value of S. So we can put 0 in the column D_1 for row 0. Similarly, for (F/F 0) the next state Y_0 follows S i.e when S = 0, Y_0 = 0, when S = 1, Y_0 = 1. that is why D_0 is written 'S'.

Row 1:- Row 1 shows that in the next state, the O/P of (F/F 1) follows the complement of Z. and (F/F 0) is always '0' independent of value of Z. so $D_0 = 0$.

Row 2 :- Row 2 shows that, in the next state, the o/p of FF1 & FF2 are logic 1. Hence we can put 1 in the column of D_1 & D_0.

Row 3 :- Row 3 shows that, in the next state, the o/p of FF1, $= \bar{F}$, & o/p of FF0 = F. <follows the complement of F> <follows F>

Since the control signals T_0, T_1, T_2 & T_3 corresponds to state S_0, S_1, S_2, S_3 i.e row 0, row 1, row 2 & row 3, the expression for the I/P of FF1 & FF0 are given by

$$D_1 = T_0 \cdot 0 + T_1 \cdot \bar{Z} + T_2 \cdot 1 + T_3 \cdot \bar{F}$$

$$D_1 = T_1 \cdot \bar{Z} + T_2 + T_3 \bar{F} \quad \text{---------- (1)}$$

$$D_0 = T_0 \cdot S + T_1 \cdot 0 + T_2 \cdot 1 + T_3 \cdot F$$

$$D_0 = T_0 S + T_2 + T_3 F \quad \text{---------- (2)}$$

Based on the above expressions for D_1 & D_0 a logic diagram can be drawn as shown in figure 15.

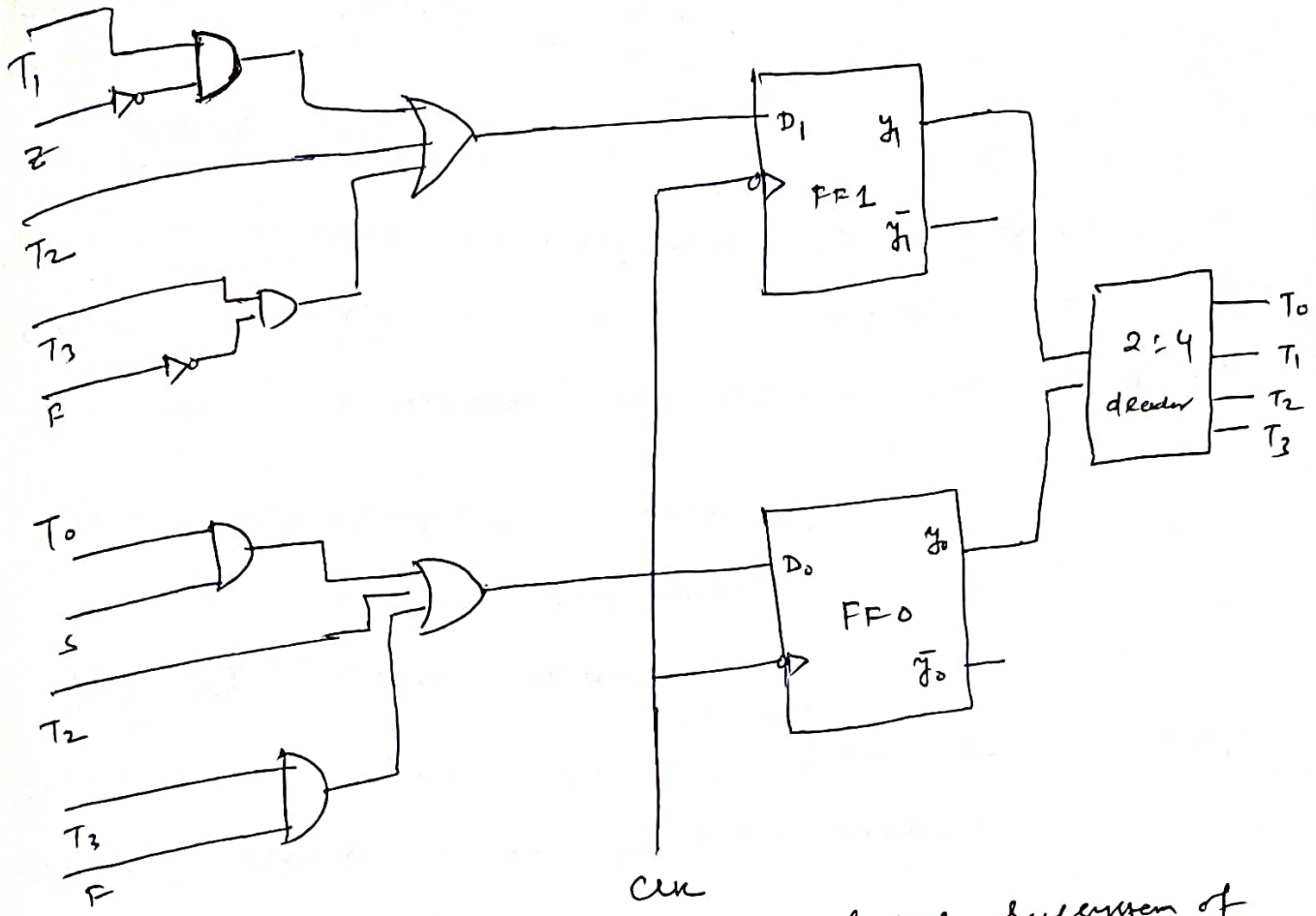


Fig. 15:- Logic ckt of Weighing machine control subsystem

— 0 —