24-Hour Clock

My clock consists of 10,6,3 mod up counters and 7 segment display (decoder)

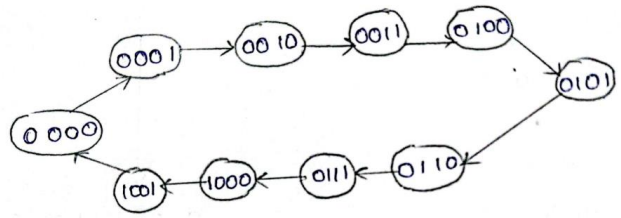
H_{0-2}	H_{0-9}	M_{0-5}	M_{0-9}	S_{0-5}	S_{0-9}

As we can see each 2nd digit in seconds and minutes and hours can be counted by 10mod counter

1st digit of minutes and hours by mod 6 counter and hour by mod 3 counter

Designing mod 10 counter

State Diagram:



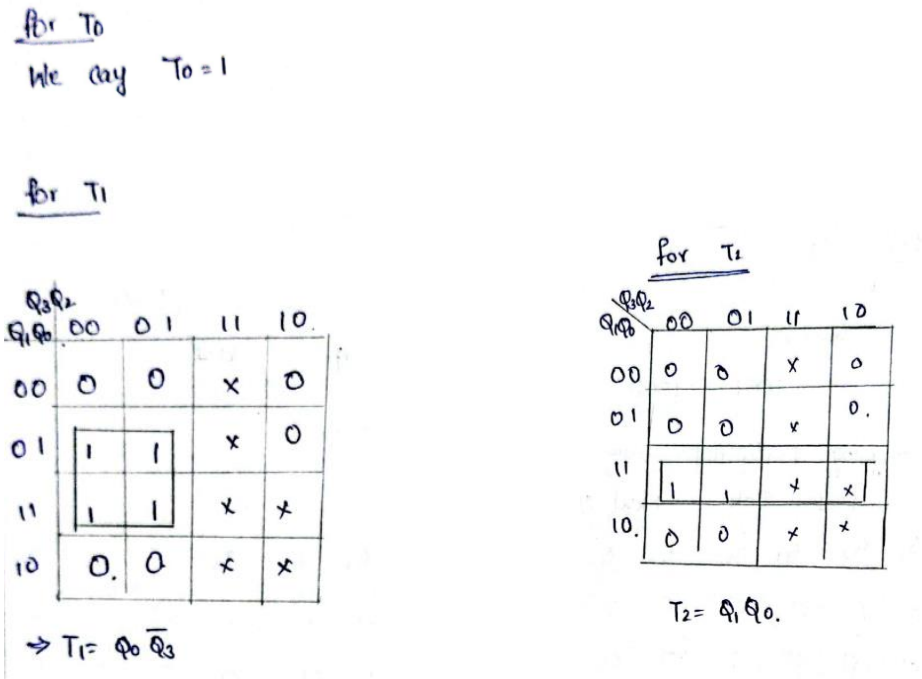
State Table:

Designing T-flip flops using state table

Present State Next state.

Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	T_3	T_2	T_1	T_0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	0	0	0	0	1	0	0	1

Solving for T_3,T_2,T_1,T_0 Using K-maps:



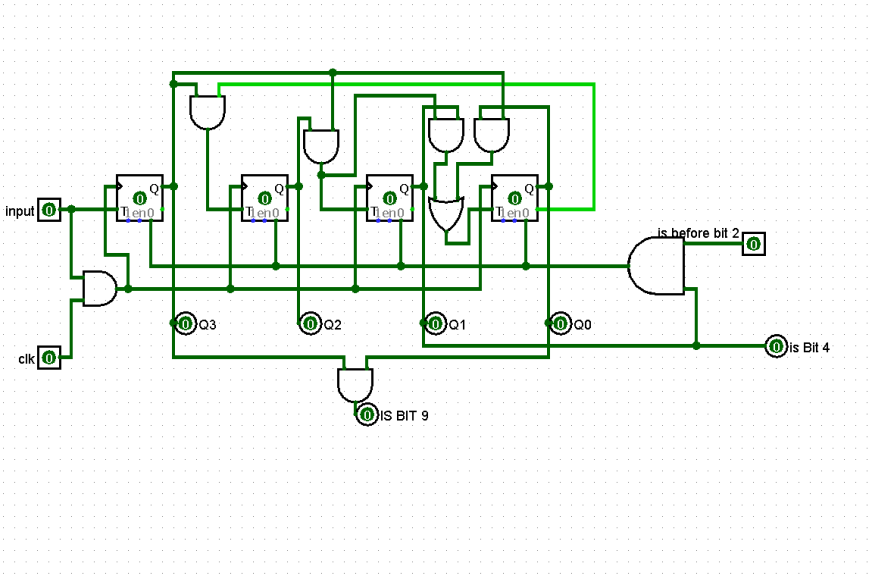
On solving We get

$T_0 = 1$ $T_1 = Q_0\overline{Q}_3$ $T2 = Q_1Q_0$ $T_3 = Q_0Q_1Q_2 + Q_0Q_3$

Designing mod 10 Counter & making use of pattern recognition

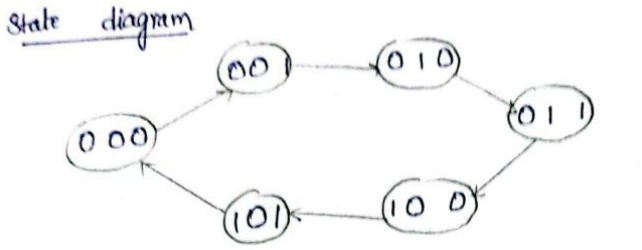
Now we make mod 10 counter using the above equations also we check is the current state is 4 (in hours part ,it makes mod 3 counter to reset so that hours don't exceed 23)

If current state is 9 then make the counter connected to it to count and reset the counter in hours part if the mod 3 counter is in state 2



Designing mod 6 counter

State Diagram:



State Table:

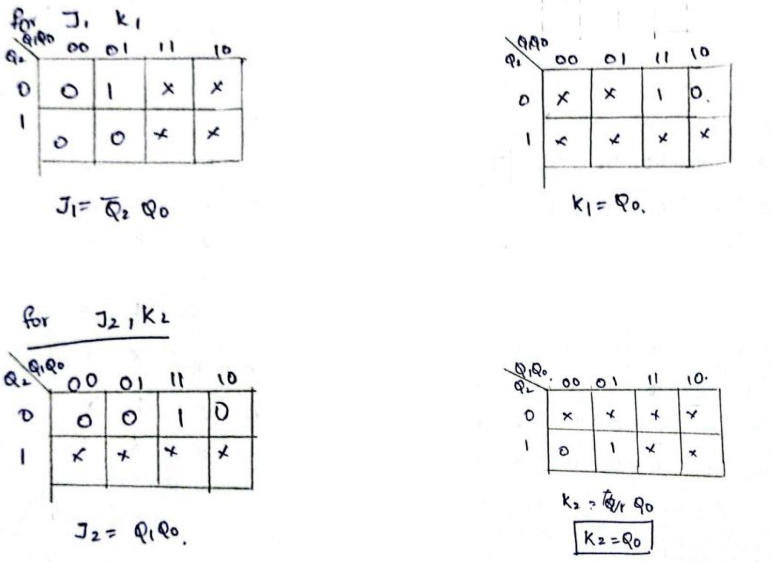
State table

Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	0	0	0	x	1	0	x	x	1

J-K exitation table

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

Solving for J_3,J_2,J_1,J_0, K_3,K_2,K_1,K_0 Using K-maps:



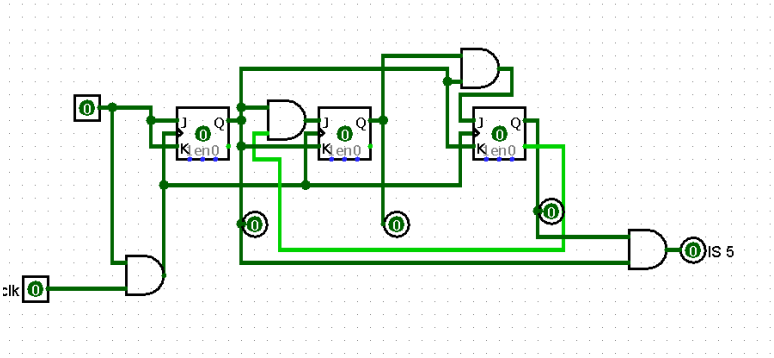
Thus we get

$J_0=K_0=1$

$J_1=\overline{Q}_2\, Q_0$, $K_1=Q_0$

$J_2 =Q_1\, Q_2$, $K_2=Q_0$

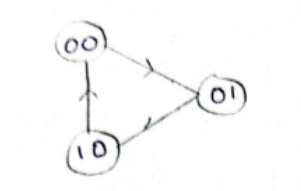
Designing mod 6 Counter & making use of pattern recognition



Now we make mod 6 counter using the above equations also we check is the current state is 5 (so that 1st bit of hours and minute part counts when the current bit is 5)

Designing mod 3 counter

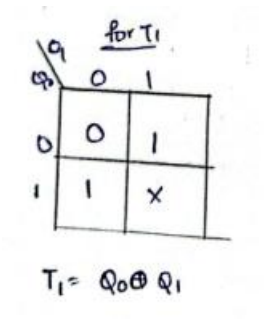
State Diagram:

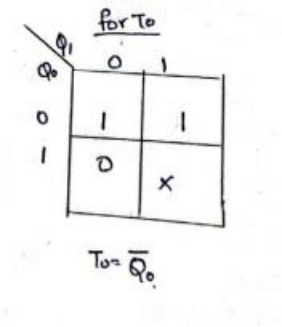


State Table:

Q_1	Q_0	Q_1	Q_0	T_1	T_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	x	x	x	x

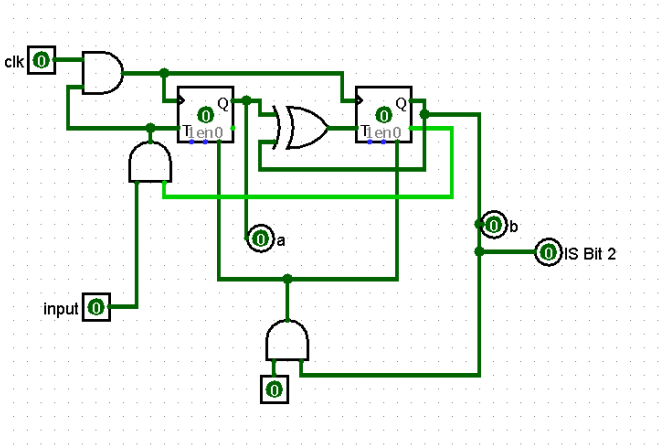
Solving for T_1,T_0 Using K-maps:





$T_0 = \overline{Q}_0$ $T_1 = Q_0 \oplus Q_1$

Designing mod 3 Counter & making use of pattern recognition



Now we make mod 6 counter using the above equations also we check is the current state is 2 and if 2nd bit of hr part is 3 then we reset both so they become 00 from 23

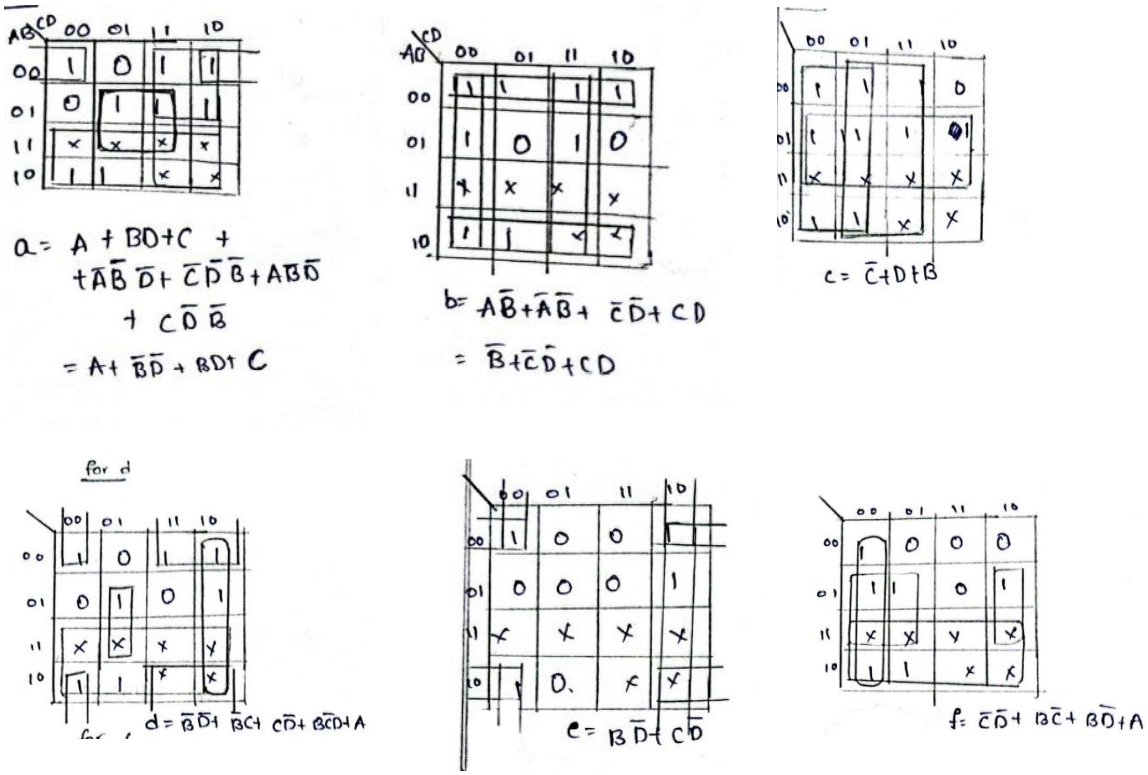
This makes out 24 Hr clock Complete.

Now we need to represent them using a 7-segment display

Truth Table of the combinational circuit of 7 segment Display :

A	B	C	D	a	b	c	d	e	f	g	h
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	0
0	0	1	0	1	1	0	1	1	0	0	0
0	0	1	1	1	1	1	1	0	0	1	0
0	1	0	0	0	1	1	0	0	1	1	0
0	1	0	1	1	0	1	1	0	1	1	0
0	1	1	0	1	0	1	1	1	1	1	0
0	1	1	1	1	1	1	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	0	1	1	0

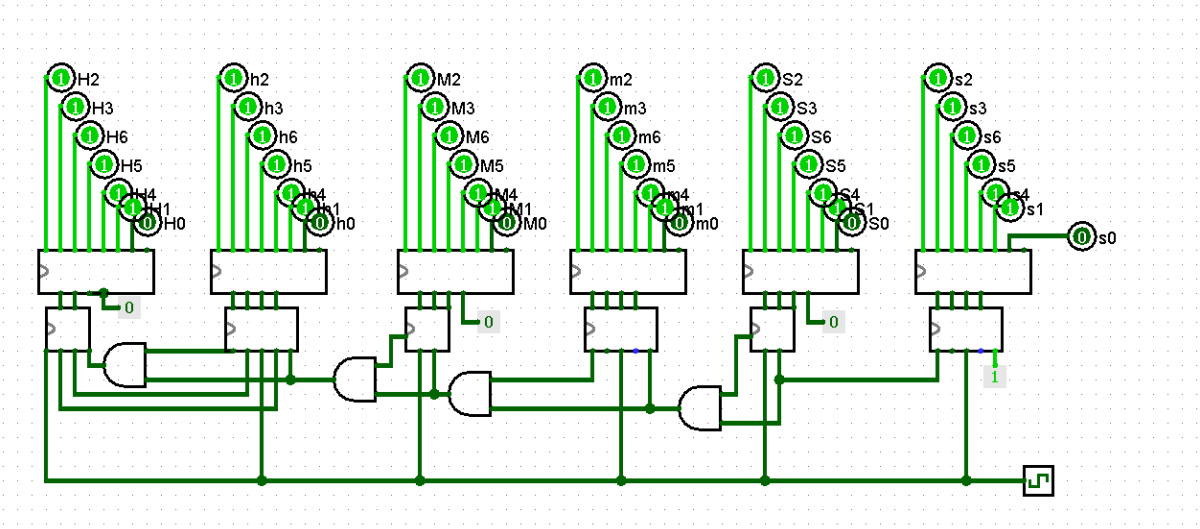
Solving It Using K-maps:



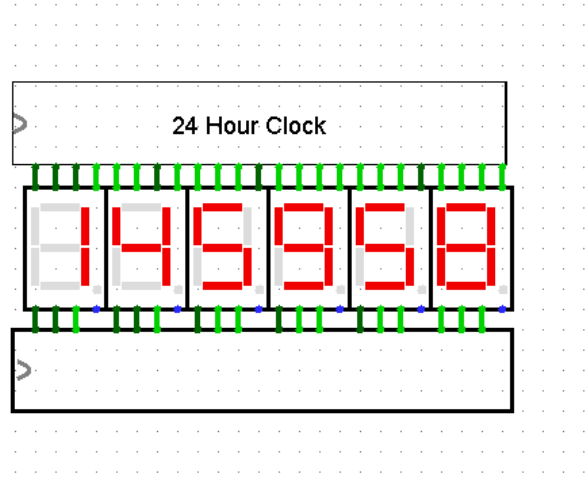
This completes our project and lets look into our implementation

Implementation:

Design of the core logic:



Making use of this lets make a Our clock with neat Design without many wires:



(A screenshot while clock is working)

Thank You

CS21B040 PAVAN