Pratyush Saini 2019CS10444

COL215 Assignment 2.

- Overview of the design of digital clock.

Modes of Digital Clock.

i) HH:MM

ii) MM:SS

where HH denotes the hours in decimal representation (0-23)
MM denotes the minutes in decimal representation (0-59)
SS denotes the seconds in decimal representation (0-59).

$H_1 H_2 : M_1 M_2 : S_1 S_2$
(i) (ii)

- Role of Push buttons

b_0 : Change the mode of clock from (i) to (ii) or (ii) to (i)
b_1 : Changing the value of first digit (H_1 in (i) and M_1 in (ii))
b_2 : Altering second digit (H_2 in (i) and M_2 in (ii))
b_3 : Altering third digit (M_1 in (i) and S_1 in (ii))
b_4 : Altering fourth digit (M_2 in (ii) and S_2 in (ii))

- We keep signals for all six digits and depending on the current mode, we use four of these six digits to be displayed.

- Timing Considerations

Assuming that the master clock has frequency 10MHz, we derive the following clocks using it.

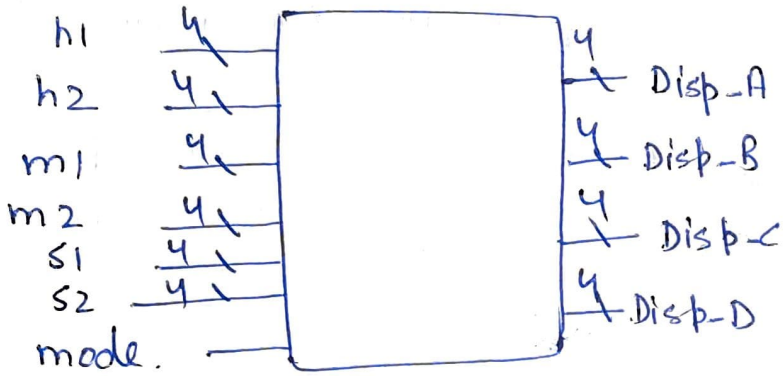
i) ENTITY DgtCLKEN

which takes 10MHz clock as input and outputs 1 at regular intervals of 1 second.

To accomplish this, we maintain a counter which goes upto 10×10^6, then outputs 1 ~~indicing~~ indicating that 1 second has been passed and then resets to 0.

ii) Refresh Clock: Discussed next ⟹

ENTITY TODISPLAYEN

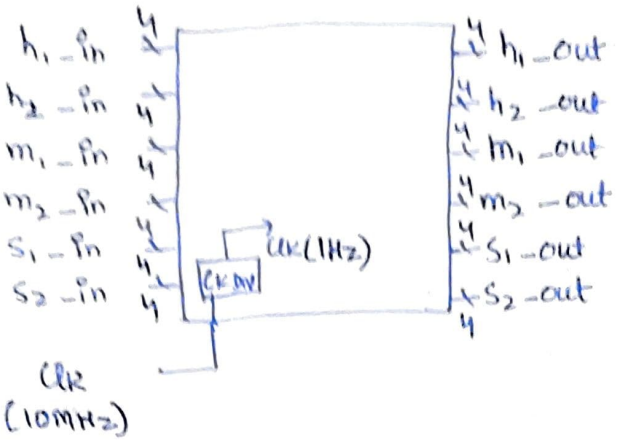


Selects which 4 digits ($h_1h_2m_1m_2$ or $m_1m_2s_1s_2$) are to be displayed depending whether mode=1 or mode=0 respectively.

i)

ENTITY

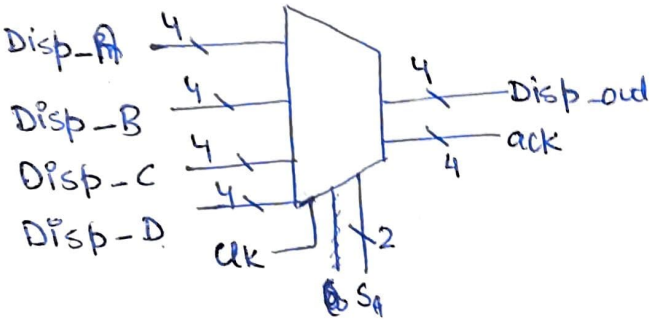
DgtCLKEN



Used the Increment the values of $h_1, h_2, m_1, m_2, S_1, S_2$ at regular intervals of 1second.

We used the Component CLK DIVEN to reduce down the frequency from 10MHz to 1Hz.

ii) DeMuxEN



Note: clk (1Hz) and S (2 bit Vector) are output from clk_div_refreshEN discussed Next.

The clock inputted in this circuits has refresh period of approximately 3.26 seconds.

This part of the circuit is used to figure out which of the four display inputs are to be displayed on the final display unit, at intervals of 3.2 second.

ack is a 4 bit vector for figuring out which Index out of 4 is to be displayed.

if S = '00', Disp_out = Disp-A ack = '1000'

S = '01', —11— = Disp-B ack = '0100'

S = '10', Disp-C ack = '0010'

S = '11', Disp-D ack = '0001'

8

iii) ENTITY clk_div_refresh EN

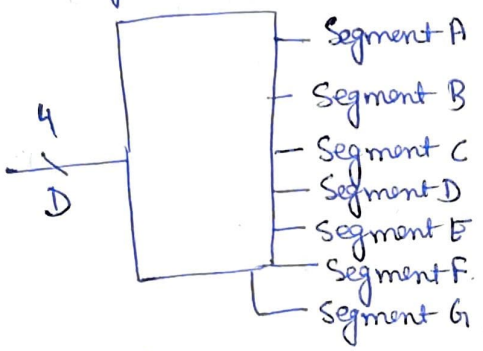
Input is 10 MHz and Current value of Select (Indexing, as discussed back).

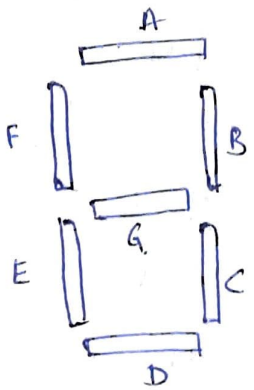
Output => clk_ref (ON after refresh period else off).

dig_out => Based on Current value of select, we increment it after refresh period to indicate at what position to display next digit.

Select	dig_out
00	01
01	10
10	11
11	00

(iv) BCD to Seven Segment Converter.





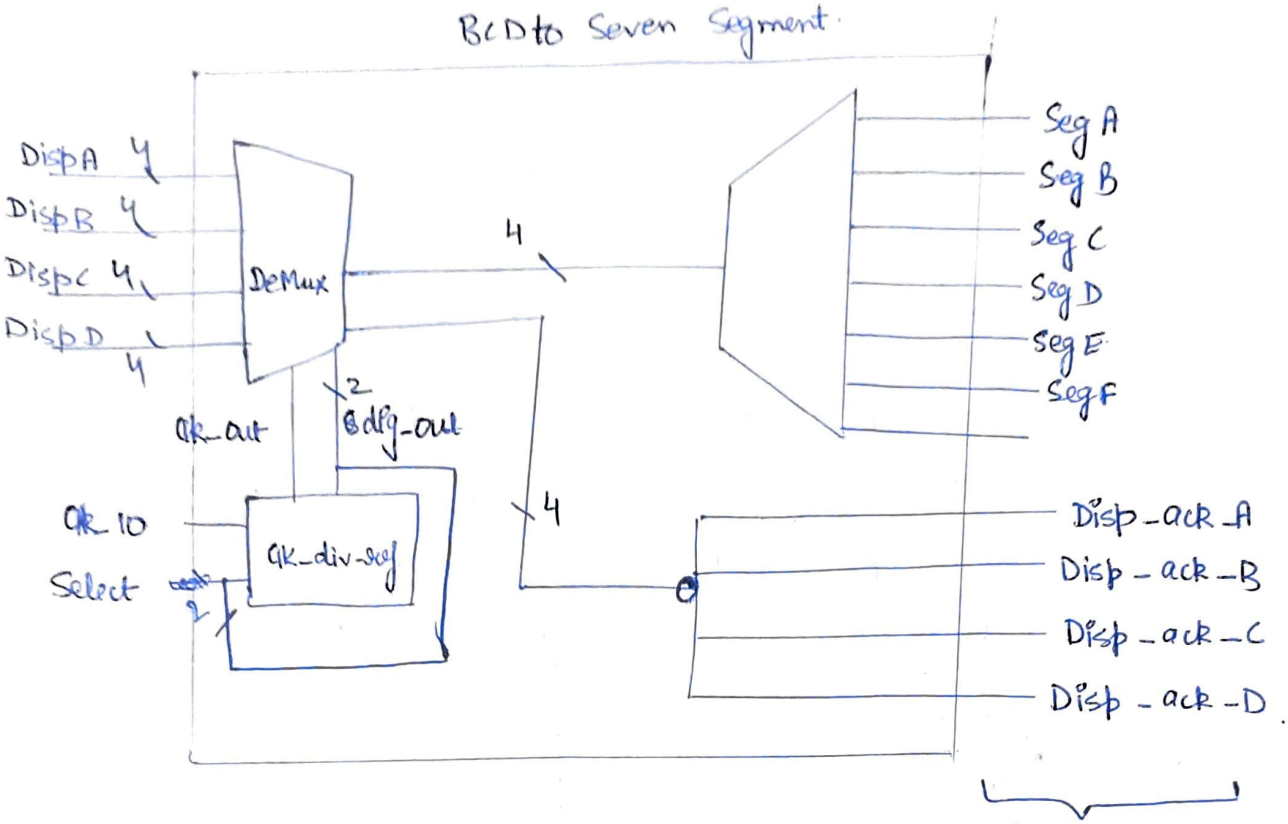
Determines which of seven segments to light up.

0000 — 1111110
0001 — 0110000
0010 — 1101101
0011 — 1111001
0100 — 0110011
0101 — 1011011
0110 — 1011111
0111 — 1110000
1000 — 1111111
1001 — 1111011
else — 0000000

2

v) ENTITY Segment Display

used Components CLK_div_refresh
DeMux EN
BCD to Seven Segment



Disp_ack

0111 => Lighting first digits and others off because anode is to be kept low for lighting to occur.

1011 => Second

1101 => third

1110 => fourth.

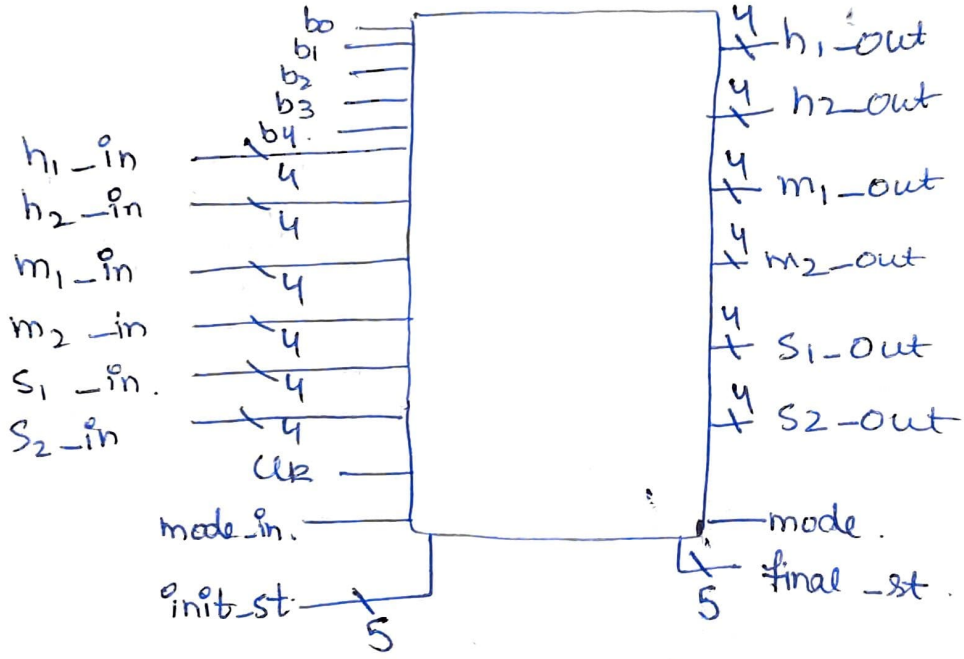
Note: As per the clock design, to reset the clock, we have to manually set all digits to 0.

ENTITY Push Buttons

Assumption : Took the asumption that we are pressing only a single button at some instant.

Priority assigned : $b_0 > b_1 > b_2 > b_3 > b_4$.

We store the initial state of the clock to overcome the problem of mis-interpretation as multiple pressing since. clock has time period of 100 ns and button is pressed for around some few milliseconds.



init_st ~~b0~~ is a binary vector of size 5 containing a single '1' to depict the initial state. If the value corresponding to current button pressed in init_st is '1', we do nothing, else change the final state and perform corresponding operation]

mode 1 : $h_1 h_2 m_1 m_2$ mode 0 : $m_1 m_2 S_1 S_2$

b_0 : Change mode of display 1 => 0 or 0 => 1

b_1 : If mode = 1, Increment h_1 else Increment m_1

b_2 : —"— h_2 —"— m_2

b_3 : —"— m_1 —"— S_1

b_4 : —"— m_2 —"— S_2

Overall Clock Design.