RISHI SHAH
2019CS10394 COL-215 Assignment

- Objective: To make a digital clock

- Input Taken: 10MHz Clock, button push ~~(=)~~ it is vector containing 3 buttons. Use of each button:
~~button(0) ⇒~~
When pressed:
  button(0): Mode of display can be changed [HH:MM ⇄ MM:SS]
  button(1): Causes change of state from display to allow change of time settings.
  button(2): Used for incrementing opertion. Considering a normal push comes in ~~is of~~ 0.1 sec and more ~~than~~ of that used in fast increament.
Logic of fast increment: For first 0.5 sec just normally increment. If button is pressed ~~more~~ more than for every 0.2 sec increment by 1.

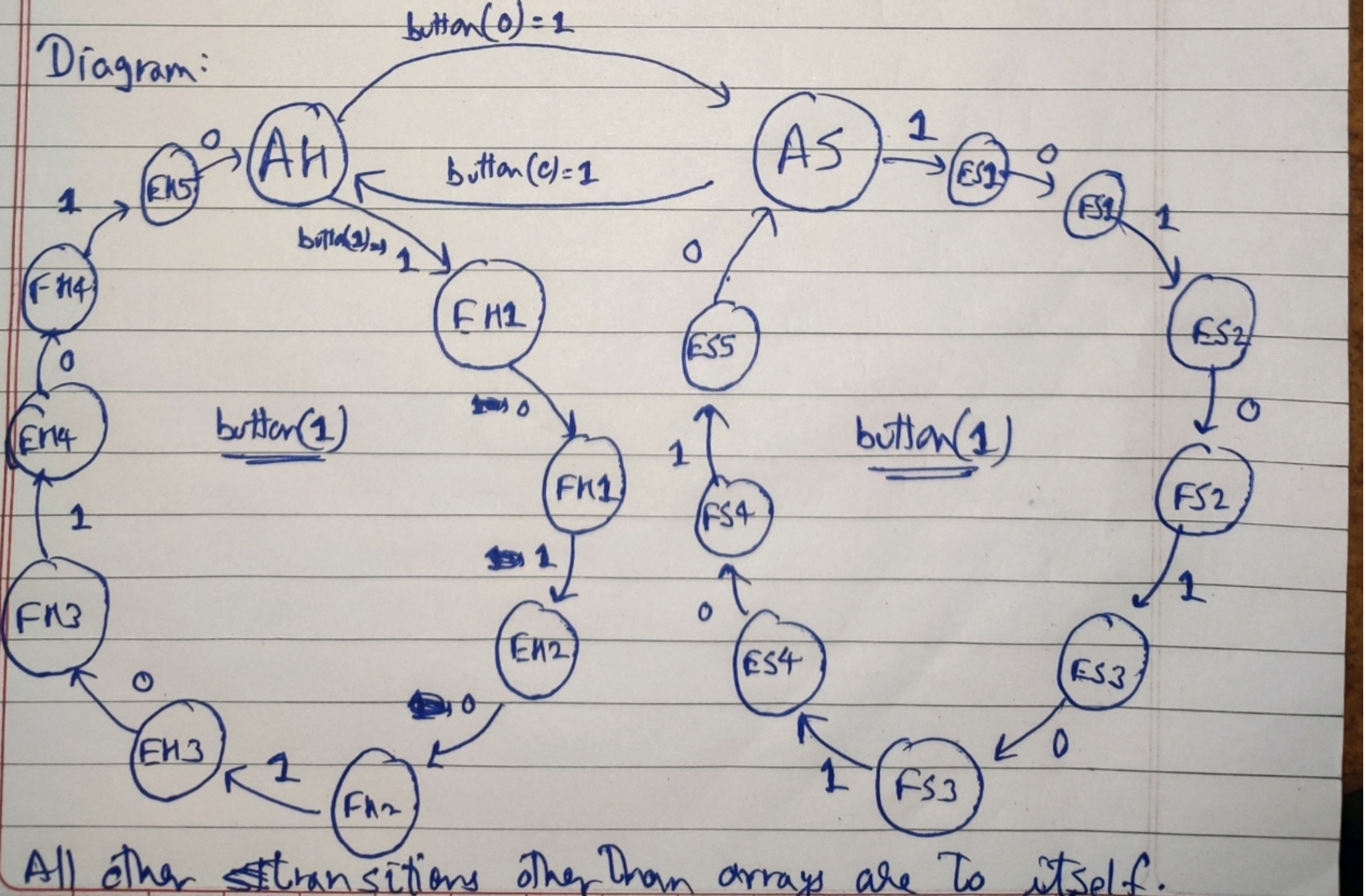
- Output:
i) anode_activate: There are 4 anodes. Each anode decides which number should be refreshed. As at any instant we can just refresh a single digit from the 4. Refresh rate should be between 1ms-16ms so as that human eyes doesnt see a difference between the refreshing. In our circuit, refresh rate is 4ms. ~~E~~ i.e. whole cycle takes 4ms (1ms for each digit.)
ii) led: It is the 7 segment display which is seen on the board. Each bit describes a part of the seven lines. Bit set to '0' means to light the led line.

- States:

AH → Normal display HH: MM
AS → Normal display MM: SS
~~EH1, EH~~
FH1 → State for increment of unit digits of ~~second~~ minute
FH2 → State for increament of 10s digit of minute.
FH3 → state for increment of unit digit of hour
FH4 → State for increment of ~~1~~10s digit of hour
FS1 → State for increment of unit digit of second
FS2 → State for increment of 10s digit of second
FS3 → state for increment of unit digit of minute
FS4 → state for increment of 10s digit of minute.

EH1, EH2, EH3, EH4, ES1, ES2, ES3, ES4 are the intermediate states so that we don't skip the incrementation of any digit. (more clear in state diagram)

Diagram:



All other ~~st~~transitions other than arrays are to itself.

State Transition:

	Present	button_push 0	button_push 1	button_push 2	Next state
Default 1	AH	1	-	-	AS
		0	1	-	EH1
2	AS	1	-	-	AH
		0	1	-	ES1
3	EH1	-	0	-	FH1
4	FH1	-	1	-	EH2
5	EH2	-	0	-	FH2
6	FH2	-	1	-	EH2
7	EH3	-	0	-	FH3
8	FH3	-	1	-	EH3
9	EH4	-	0	-	FH4
10	FH4	-	1	-	EH4
11	EH5	-	0	-	AH
12	ES1	-	0	-	FS1
13	FS1	-	1	-	ES2
14	ES2	-	0	-	FS2
15	FS2	-	1	-	ES3
16	ES3	-	0	-	FS3
17	FS3	-	1	-	ES4
18	ES4	-	0	-	FS4
19	FS4	-	1	-	ES5
20	ES5	-	0	-	AS

- Design:

i) Display entity and its architecture: Its input are 10MHz clock and button_push. The outputs are given to board to display. anode_activate & led. It ~~used~~ uses instance of entity assign, to get the 4 digit number to be displayed. It ~~used~~ uses instance of entity clk_part1ms to get 1ms clock which is used in the process for refresh rate.

ii) Assign entity and its architecture: It gets clk and buttons as input and gives displayed number as output. It uses all the states describe before to generate the desired output. It ~~used~~ uses clk_part1 entity to get 1sec clock for normal ticking of the clock. Bit vectors → S0, S1, M0, M1, H0, H1 are used for working which form the displayed number depending on state.

iii) clk_part1 entity and ~~design~~ arch: It gets the 10MHz clock as input and gives 1Hz clock as output. We start a counter and when counter reaches 4999999, the 1Hz clock changes its value (low becomes high, high becomes low). As 0 to 4999999 means 5×10^6 rising edges so half period.

iv) clk_part0_1s entity & design: It gets 10MHz clock as in and gives 10Hz i.e 0.1sec clock as output. Same working as clk_part1, just counter reaches 499999.

v) clk_part1ms entity & arch: It gets 10MHz clock as input and gives 10^3 Hz i.e. 1ms clock as output. Same working. Just counter reaches 4999.