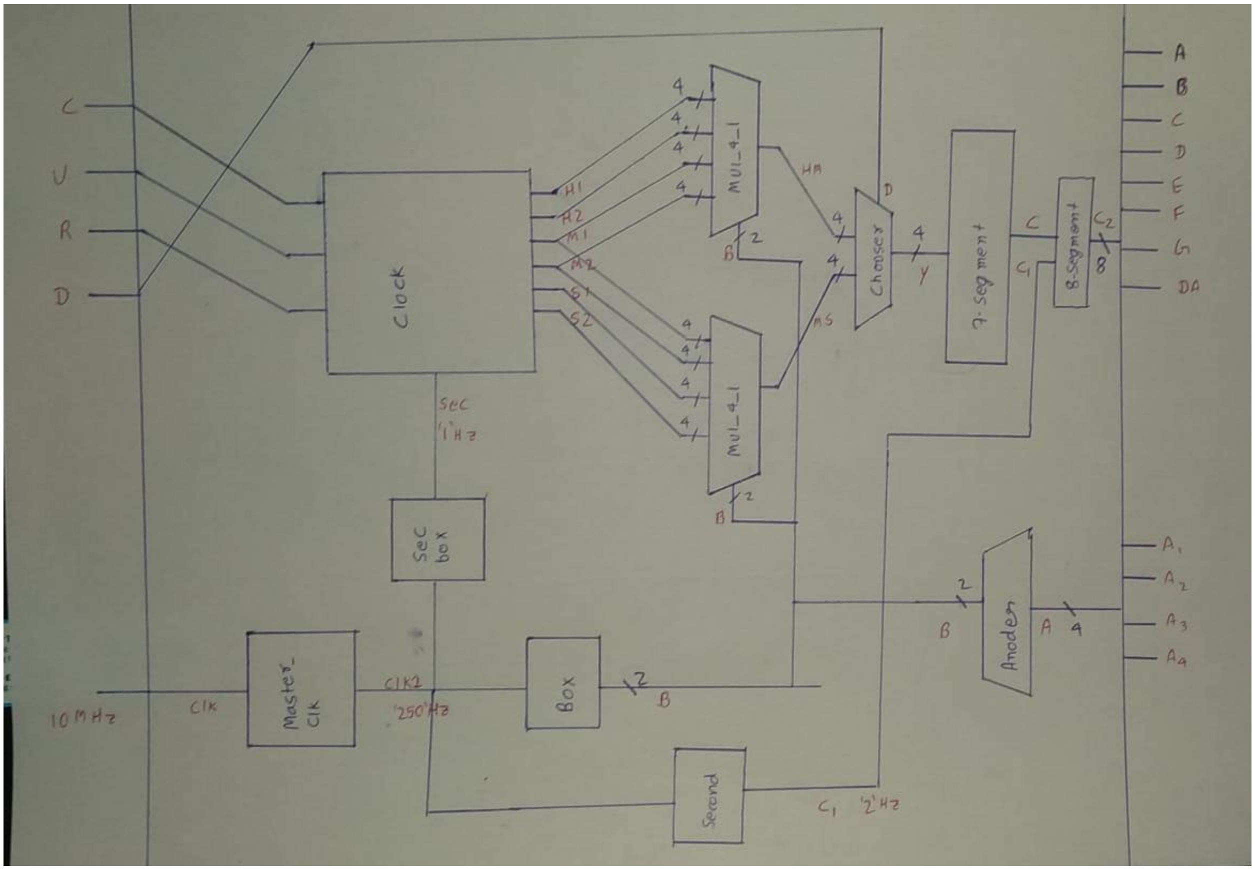
DIGITAL CLOCK

-Vivek Choudhary (2019CS10413)

Introduction

We have to built a digital clock which have specification like

1) It will show time in format HH:MM in normal mode (D-button=0) and MM:SS in another mode (D- button=1).
2) It should be able to change time when C-button=1 with constraint HH=[0,23], MM=[0,59], SS=[0,59].



-

Input and Output in functional block:

INPUT

- 10MHz clock (clk)
- Mode button (C) when C=0 display normal clock and when C=1 change time as user sets it.

- Value increment button (U) when U=1 increment value at that position by one.
- Position button(R*) when R=1 shift position in right by one position and it's default value is position zero(H1)
- Display mode (D*) when D=0 display HH:MM format and when D=1 display MM:SS.

*Remember button U and R will only work when button C=1.

OUTPUT

- Cathode side A, B, C, D, E, F, G, DA all in 8-bit output.
- Anode side A1, A2, A3, A4 single bit output.

Detailed description of all module used in functional block.

Master clock: It takes 10MHz(clk) as input and gives output 250Hz(clk1). It is slowing clock by doing counting method when counter reaches 20,000 it gives output '1' and when counter reaches 40,000 it gives output'0' and this process will continue in cycle.

Sec box: It takes 250Hz(clk1) as input and gives output of 1Hz(sec). It is also working on same principle as above but here counter inflection points are 125 and 250.

Box: It takes 250Hz(clk1) as input and 2bit_vector(B) as output. And 2bit_vector increases by one in every 4mili seconds.

Second: It takes 250Hz(clk1) as input and 2Hz(C1) as output.

Clock: It take 1Hz(sec), button(C,U,R) as input and six 4bit number(H1,H2,M1,M2,S1,S2) as output. If C=0 just display time and update sec, min, hour every second and modulo for H1 is 3, H2 is 10, M1 is 7, M2 is 10, S1 is 7 and S2 is 10. And if C=1 then sets all values to zero as 00:00:00 and default position of changing digit is leftmost position(H1) so if R=1 then shift position to right by one place. Now if U=1 then increase value of digit at that position by one with their respective modulo.

Mul_1_4: It take four 4-bit_vector and one 2-bit_vector(B) as input and give one 4-bit_vector as output. And choosing one out of four depend upon values of 2-bit_vector.

Chooser: It takes one bit(D) and two 4-bit_vector as input 'which are the output of Mul_1_4' and give one 4-bit_vector as output. And the selection is done by D. If D=0 display HH:MM and if D=1 display MM:SS.

Anoder: It takes 2-bit_vector(B) as input and four one bit(Ai) as output. Values of Ai depends upon B and also remember only one of Ai's are '1' at a time.

Seven_segment: It takes a 4-bit_vector(Y) as input and 8-bit_vector(C) as output and in output 'dot' position is always off. Basically it converts binary number to decimal number in range 0 to 9 by some complex circuit in it.

Eight_segment: It takes one 8-bit_vector(C) and bit(C1) as input and gives one 8-bit_vector as output. In this output all dot is blinking at rate of 2Hz.