**Prakhar Aggarwal**

**2019CS50441**

**Explanation of VHDL code**

**1ST ENTITY**

1st entity's role is to use clk which is of 100Mhz frequency and give me 2 clocks of frequency 1Hz and 2Hz.

cnt1 and cnt2 are incremented at each rising edge of clk. If cnt1 reaches 108/2, I will start it from 1 and flip the value of clk\_on. Clk1\_on stores whether my clk1 is at 1 or 0. Clk1\_on will be 0 for ½ second and 1 for remaining ½ second giving me a clk of 1Hz. Similarly, we can get a clock of 2Hz.

**2ND ENTITY**

2nd Entity’s role is to ensure proper functioning of my Digital Clock i.e timely increment in seconds, then minutes and then hours.

**INPUT:** We have all 5 buttons, clk1, clk2 from 1st Entity, dot\_show is for the HH:MM display mode in which last decimal point will be flashing at 1Hz frequency. Final\_hr, final\_min, final\_sec are the output vectors which store the output value of hour, min and second respectively. Final\_display\_mode is to determine in which display mode am I (HH:MM or MM:SS).

**SIGNAL:** temp\_hr, temp\_min, temp\_sec stores the temporary value of hour, minute and second respectively, edit\_mode\_value checks if I am in edit mode or not, left\_value has significance only if edit\_mod\_value is 1 and it keeps count of at which LED am I currently. temp\_display\_mode\_hour tells me whether I need to display HH:MM or MM:SS. Dot stores whether decimal point should be on or off and will passed to dot\_show.

**ARCHITECTURE:** On pressing reset button, every value be reset to 0 and we will be directed to default display mode (HH:MM).First process block flips value of dot with rising\_edge of clk2 (2Hz frequency). temp\_display\_mode\_hour is assigned current value of display mode. 2nd Process block flips temp\_display\_mode\_hour value on pressing of display\_mode button. 3rd Process block flips edit\_mode\_value value on pressing of edit\_mode\_input button. In 4th Process block, if edit\_mode\_value is 1, then I will look for left button and increment button and accordingly change value of temp\_hr, temp\_min and temp\_sec. After this, I will change temp\_hr, temp\_min and temp\_sec with positive edge of clk1 (1Hz clock). In last, I will assign all temporary signal values to the output values mentioned in entity.

**3RD ENTITY**

3rd Entity is responsible for displaying numbers on LEDs.

**INPUT:** final\_display\_mode is the current display mode in which I need to display my numbers. Final\_hr, final\_min, final\_sec are the outputs from 2nd entity and are the value of hour, minute and second to display. Dot\_show is the output of 2nd entity. Anode, cathode are the outputs which will determine which led is to be kept on and which segments are to be kept on.

**SIGNAL:** Hr\_, min\_, sec\_ are the integer values gained after converting vector to integers. Hr1, min1, sec1 stores the first digit of hour, minute and second, similarly, hr2, min2, sec2 are used for second digit. Led\_number keeps track of which led to display, anode\_on will on that anode as directed by led\_number. cnt is used to change between different LEDs (digit period). Blink is used for flashing decimal point. Display\_digit stores which digit to be displayed on current led. 1st process block will equally divide time between all 4 LEDs. 2nd Process evaluate values of hr1, hr2, min1, min2, sec1, sec2 and will ON anode according to led\_number and will store it in anode\_on. It will also evaluate display\_digit and will ON the cathodes required to display that digit.

**4TH ENTITY**

4th Entity is the final entity which combines all 3 entities.

**INPUT:** All buttons which are controlled by user are in the input of 4th Entity i.e. reset, display\_mode, edit\_mode\_input, left, increment button. Anode and Cathode vectors are the output final vectors which can be used to display numbers on BASYS3 FPGA Board.

**FINAL\_OUTPUT:** Its architecture includes calling all 3 entities with appropriate inputs and outputs.