Assignment 1 COL215P

Shreyansh Singh Aman Verma

September 2022

1 Components

- ClockDivider: Divides the 100MHz clock into 2 clocks:
  1. 10Hz: This clock is used for keeping time when the clock is not paused.
  2. 200Hz: This is used for refreshing the digits on the seven segment display(5 ms refresh rate means 50*4 changes every second with requires a 200Hz clock).
- TimeKeeper: Used for setting the m, s1, s2, t values to maintain time as the stopwatch runs.
- BCD_to_SSD: Converts the digit(including with or without decimal point) to be currently displayed on the seven segment from binary coded decimal to an 8 bit representation(7 for A-G and 1 for DP).
- Stopwatch: It brings together all the components together to make a fully functional stopwatch. Also includes functionality for debouncing button inputs - start, stop and reset.

2 Description

2.1 Seven Segment Display

The BCD_to_SSD component receives as input the BCD encoding of the digit to be displayed and also if the decimal point should be turned on or off. It then converts this value to the SSD encoding and returns the 8-bit value to the stopwatch.

2.2 Timing circuit for driving LEDs

The ClockDivider divides the given clock into two frequencies as stated earlier - 10Hz and 200Hz. This done by simply flipping a bit using a counter which repeatedly counts from 0 to $\frac{100Mhz}{2 \cdot desired\_frequency} - 1$ in every clock cycle and flips the bit every time the counter is reset to 0.The 200Hz clock is used to drive the seven segment display output in the Stopwatch module.

The digit to be displayed is selected in a round robin fashion just as the anode using 4 states which are represented as follows:

1. 0111: Selects m as the BCD encoding for which we need to compute the SSD encoding. In this case we display the digit at the first place.

2. 1011: Selects s1 as the BCD encoding for which we need to compute the SSD encoding. In this case we display the digit at the second place.

3. 1101: Selects s2 as the BCD encoding for which we need to compute the SSD encoding. In this case we display the digit at the third place.

4. 1110: Selects t as the BCD encoding for which we need to compute the SSD encoding. In this case we display the digit at the fourth place.

The states are named in a fashion which is also directly used as input to the anode of the seven segment display. In this manner the values of m, s1, s2, t are displayed in a suitable manner on the display.

2.3 Modulo counter

Using the 10Hz clock, in the Timekeeper component, we keep track of the values of the 4 counters -

- m: keeps track of the minutes passed. Values in range 0 to 9, increment of 1 when s1 is reset to 0.

- s1: keeps track of the tens place of the seconds passed. Values in range 0 to 5, increments of 1 when s2 is reset to 0.

- s2: keeps track of the ones place of the seconds passed. Values in range 0 to 9, increments of 1 when t is reset to 0.

- t: keeps track of one-tenths of seconds passed. Values in range 0 to 9, increments of 1 in every clock cycle of 10Hz clock.

We also have two additional signals - paused and reset which are inputs to the Timekeeper component from the stopwatch:

- paused: when this signal is true then the counter is not incremented.

- reset: when this signal is true the m, s1, s2, t values are reset to 0.

In this manner, the values are maintained correctly along with maintaining the state with the button inputs.

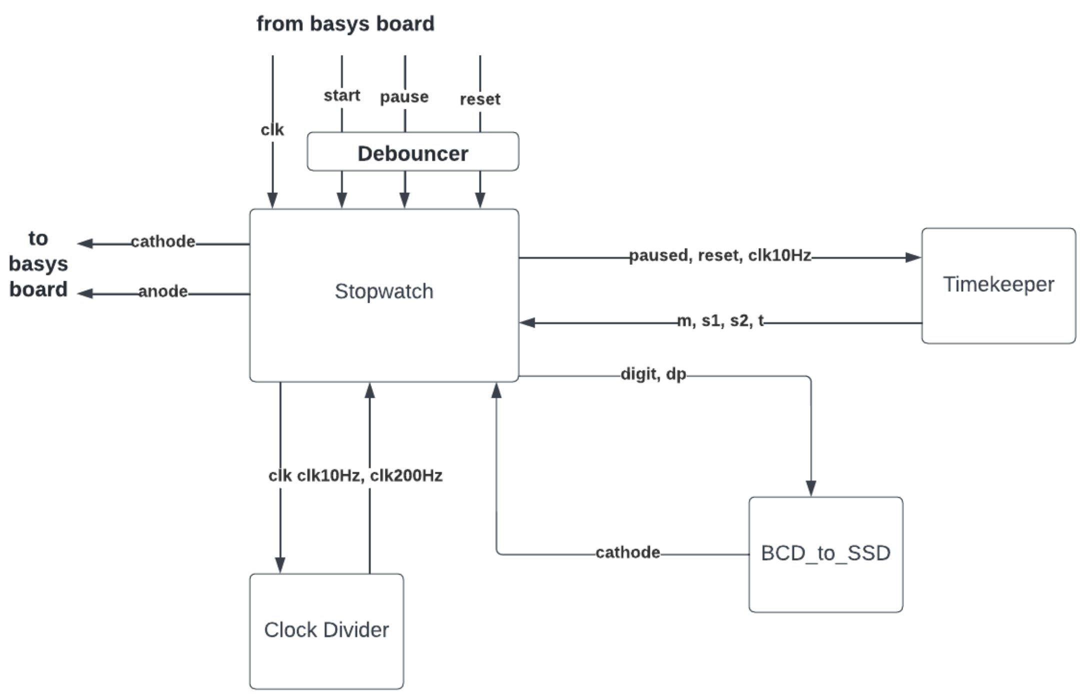
2.4 Buttons

We also have three button inputs -

- start - sets the paused state(which is passed to timekeeper) to false.
- stop - sets the paused state(which is passed to timekeeper) to true.
- reset - sets the reset state(which is passed to timekeeper) to true.

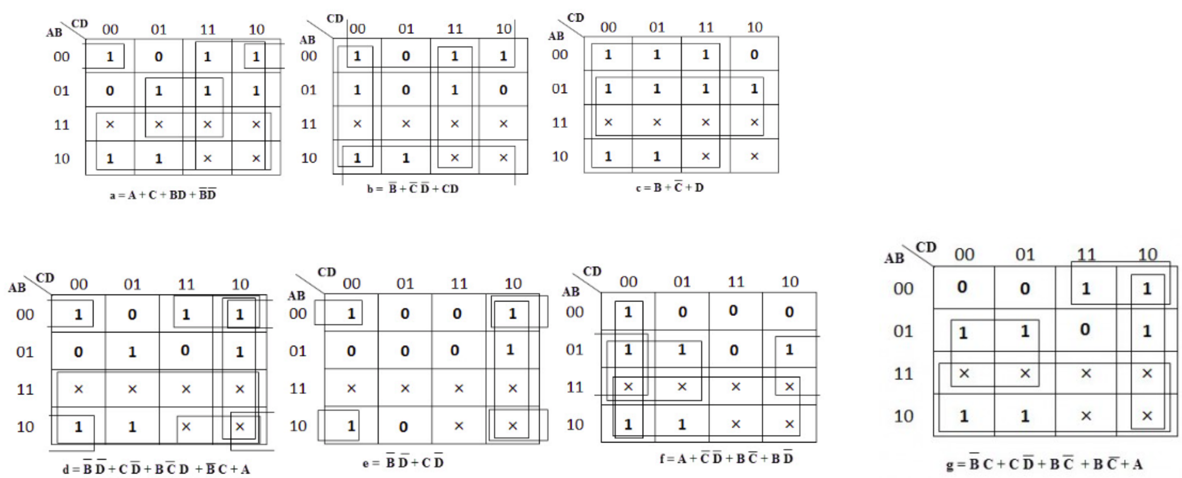
However, the input signals aren't used directly due to a bounce which is produced when the button is pressed as the input takes time to stabilize. To eliminate this, we refine the input signals by setting it to 1 only when the previous 3 are 1. This eliminates the bounce in the input signal.

3 Block Diagram



4 Truth Table

Decimal Digit	Input lines				Output lines						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1



Kmap used to solve the truth table.
(We need not of all the values since ACTIVE is denoted by 0 in basys board)