Lab 6: Stopwatch

Shrey Patel
2019CS10400

Aayush Goyal
2019CS10452

May 2022

Contents

1 Implementation

1.1 Display

- The display has been handled using the code developed in Assignment 3. All the four digits are displayed in turn with each digit being displayed for 2^{18} clock cycles (or roughly 2.62 ms) which is less than the human persistence of vision due to which all the digits look as if they are all on at the same time.
- The leftmost digit corresponds to the tenth of a second, 3rd digit corresponds to the units digit of second, 2nd digit corresponds to the tens digit of the second and the first digit corresponds to minute.
- To separate the minutes, seconds and deciseconds, we have also displayed a decimal digit after the first and the third digit.

1.2 Updating the Time

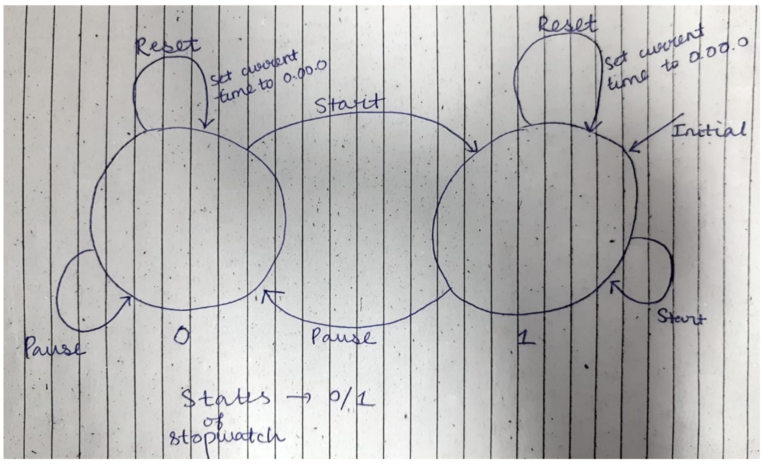
- We need to obtain a clock of roughly 10 Hz using the original clock since our last digit must update 10 times every second to be comparable to a real stopwatch. For that, we have obtained a slow cycle which updates every 2^{22} cycles of the original clock. So, the frequency is roughly 12Hz.
- The last digit is now updated at the rising edge of this 12Hz clock. It is updated modulo 10. The 1st and 3rd digit of the display are updated modulo 10, while the 2nd digit is updated modulo 6.

1.3 Handling Input

- There are three inputs which the user can give through push buttons: start, pause and reset. Start is mapped to left button (BTNL, W19), Pause is mapped to centre button (BTNC, U18) while Reset is mapped to the top button (BTNU, T18).

- The stopwatch can be in two states: ’0’(paused) and ’1’(running). Pause causes the stopwatch to move to state ’0’ from ’1’ from which no update to time occurs (no change if already paused). Start causes a paused stopwatch in state ’0’ to move to state ’1’ and the clock resumes updating from the current time(again, no change if already running). On reset, the value of current time is set to zero regardless of whether the clock is paused or running.

This can be summarised in the following FSM:


Reset
set current time to 0:00.0
Start
Reset
set current time to 0.00.0
Initial
Pause
0
Pause
1
Start
States of stopwatch → 0/1

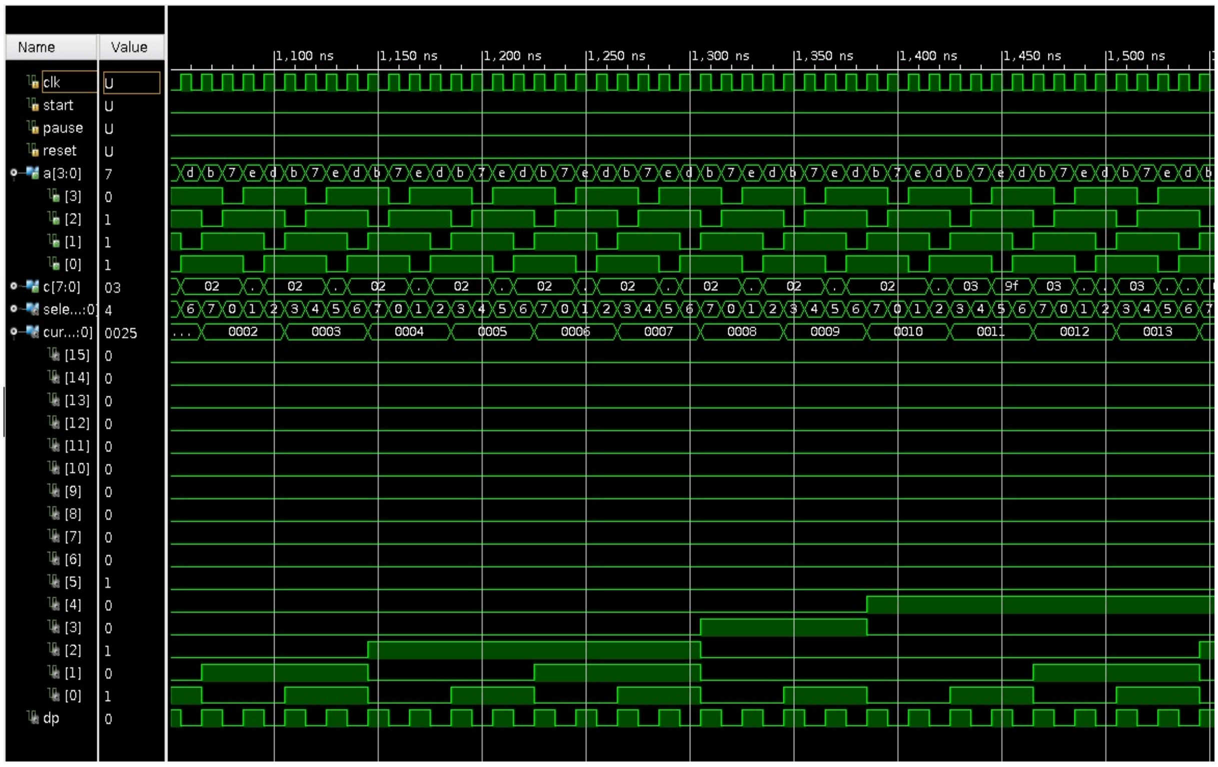
1.4 Debouncing

- Since we are using push buttons, there is naturally an issue of fluctuations in input due to irregularities in pushing the button by the user. So, a single button push is interpreted by the device as a series of fluctuations starting from 0 and finally ending at 1 till the time the user has kept the button pressed. And since the user typically presses the button for some milliseconds which is thousands of clock cycles, the high number of fluctuations in the device ports and registers may damage the device. This can be solved using debouncing.
- We discussed that the main problem is that the fluctuations are massive as seen by our original clock. So, we slow down the clock, so that the probability of those perturbations coinciding with the rising edge of this slow cycle is very low. So, even if the button input changes from 0 to 1 and back to 0 multiple times before fixing at 0, the device interprets this as a single transition from 0 to 1, and therefore only one assignment to a register/port. In this assignment, we have used a slow clock which updates every 2^{21} cycles of the original clock. So, the time period of this slower clock is roughly 42 ms which is much larger than the duration of those perturbations, but still smaller than the amount of time for which the user keeps the button pressed. Thus, any button pressed is satisfactorily detected without any damage to the device.

2 Simulation

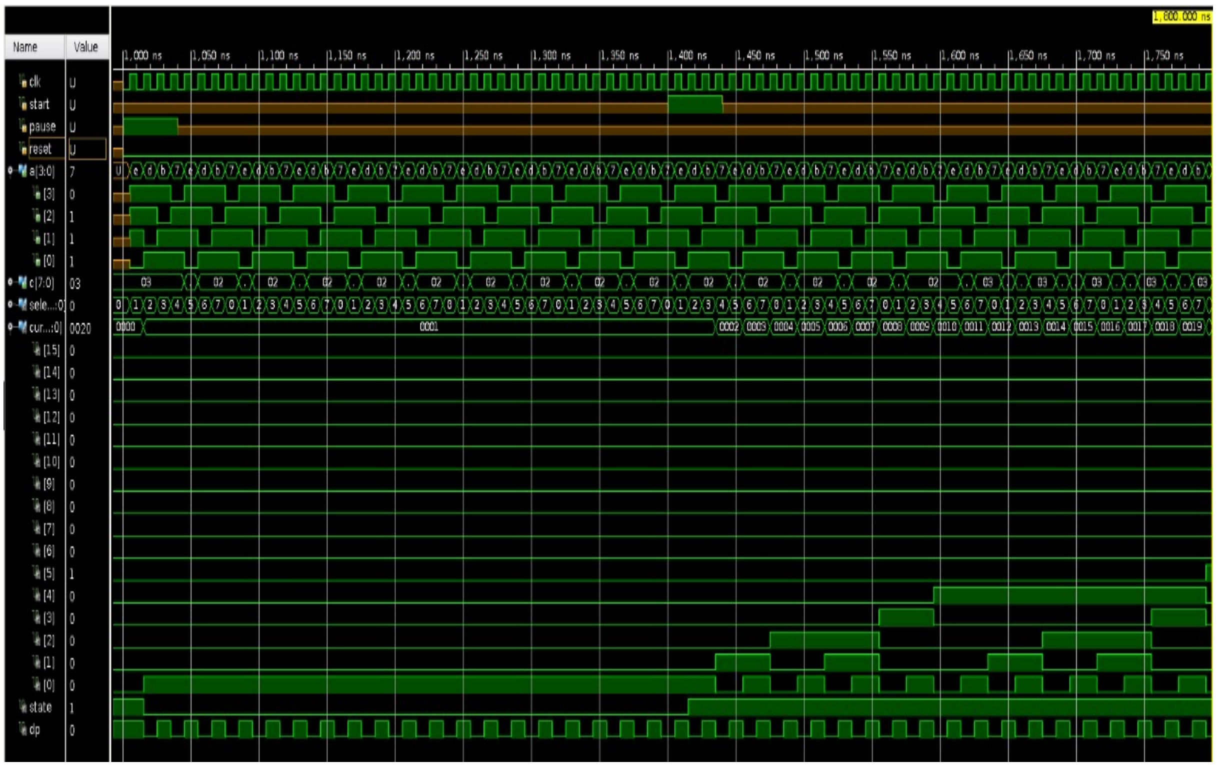
2.1 Functioning of Clock

- The display of stopwatch uses the seven-segment display from Assignment 4, in which every digit is displayed turn by turn for certain number of clock cycles(which is 4 cycles in the simulation below). This is also evident from the anode positions which update every cycle. So, the digit to be displayed changes every 10ns.
- The time counter updates every 40ns, which can also be seen in the below simulation. We can also see that the last digit goes upto a maximum of 9 before changing back to zero, in which case the previous digit increases by one. (The same logic applies to every digit, but showing all such transitions cannot fit in the scale of this image).
- Additionally, we can see that the decimal point is displayed for the first and the third digits (so it appears to be alternating between anode positions 0111 and 1101).



2.2 Handling Buttons

2.2.1 Pause/Start

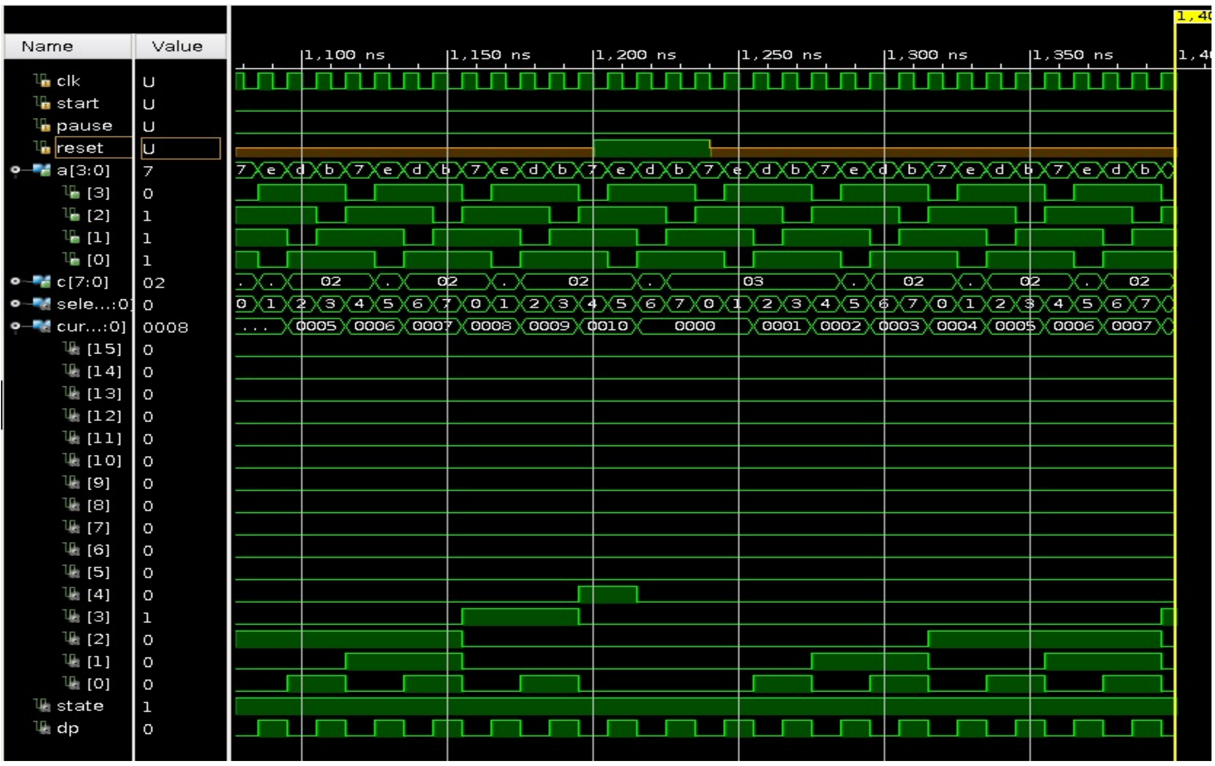


t

- In the above simulation, the pause button is pressed between 1000 and 1040ns, and this causes the stopwatch to reach state '0' due to which the current time freezes at "0001".
- The start button is pressed between 1400 and 1440ns, and this causes the stopwatch to reach state '1' due to which the time counter resumes updating from 0001.
- This show that both the start and pause buttons work as intended.

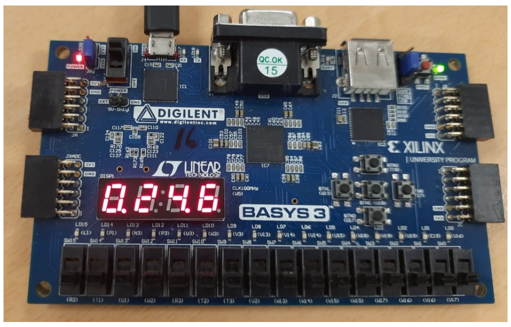
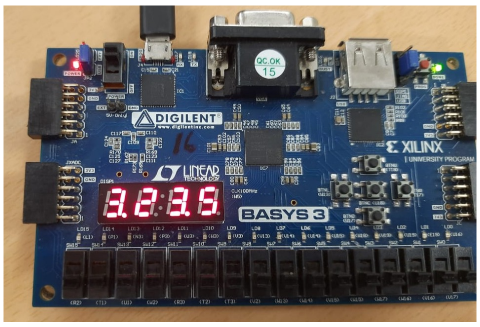
2.2.2 Reset

- The reset button is pressed between 1200ns and 1240ns, which causes the time counter to reset its value to zero in the next clock cycle(around 1210 ns). This shows that the reset button works as intended.

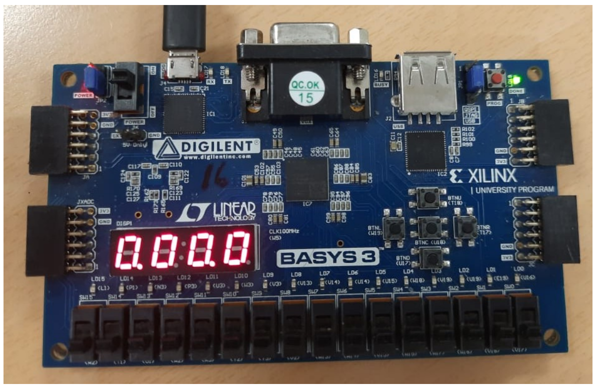


3 Output

- Some examples of the clock

- On reset(while paused)



4 Resource Utilization

- **LUTs:** 29
- **Flip Flops:** 51
- **Block RAMs:** 0
- **DSPs:** 0