A Design of Digital Stopwatch Circuit with Chip Implementation

Yu Han SHEN
School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom
ys5g15@ecs.soton.ac.uk

Abstract. This paper outlines the design, simulation, and testing of a stopwatch circuit using Tanner S-Edit and L-Edit design environment. The configuration of this chip includes button synchronizer, main sequencer unit and a seven-segment decoder. Schematic and layout of the chip was developed using Tanner design suite. After fabrication, test vectors were generated, and applied in simulation and on the physical chip. Improvements were made to create a golden design version of the chip that passed testing. An attempt was made to implement this design, including the missing modules, onto a Custom Programmable Logic Device (CPLD).

1. Introduction
Digital stopwatch is an important timing device widely used in both industrial and daily life. Compare with conventional mechanical stopwatch, the main property of digital stopwatch includes low cost, high precision and high reliability. In this paper, the construction and performance of a digital stopwatch circuit will be presented from both simulation and testing. This design provide function of Go, Stop and Reset, output via seven-segment display, providing a timing resolution of 0.1 seconds.

The main component in digital stopwatch is main sequencer. A 4-digit main sequencer was designed to reduce space and complexity, compared to a 5-digit design. The main sequencer was split up into smaller modules, consisting of counters of various sizes and an output control unit (OCU). What follows is the chip design, simulation, and layout. Schematic was developed using Tanner S-Edit, and tested via ModelSim simulation. When the simulation result proves the design behave as expected, the layout carries on in L-Edit. For the post fabrication testing, different approach was applied including test vectors, simulation, physical chip testing, and system testing.

The design of digital timing devices has been studied for years. Many design were achieved using FPGA or CPLD [1]-[6]. Design and implementation of digital stopwatch based on FPGA. [1] proposed a digital stopwatch with VHDL and schematic input based on Quartus II software platform. The designed circuit had been implemented on FPGA, and downloaded to Cyclone EP2C5Q208C8 device to test. Results showed that [1] had designed a circuit to achieve time display, start and stop, rest, and overflow alarm. The author of [2] was proposed a digital clock calendar including components of comparator, counter, multiplexer and decoder. The design was implemented using VHDL and performed with the Spartan®-6 FPGA SP605 Evaluation Kit.

The rest of the paper is organized as follows. Section II briefly introduces the design simulation and layout. Section III describes integration. Post fabrication testing is described in Section IV. Finally, conclusions are drawn in Section V.
2. Design Simulation and Layout

2.1. Button synchronizer
The button synchronizer was designed using sequential logic. A logic truth table was made with the inputs “Go” and “Stop”, the output “Run”, and the present and next states “Q0” and “QN”. Equations for “Run” and “QN” in AND and OR logic were derived from the truth table. De Morgan’s law was applied to these equations so that NAND logic was used. This reduced the gate count and complexity, as less inverters were required in the layout.

2.2. Main Sequencer
To simplify the design of the main sequencer, the overall functionality of the module was analyzed and separated into several blocks. This methodology also made designing the layout of the main sequencer easier. The blocks used to design the main sequencer consisted of: a 2-digit counter, a 10-digit counter, a 5-digit counter, and an output control unit (OCU). All modules were designed using sequential logic to produce synchronous designs, so the entire main sequencer was a synchronous design.

The three counter blocks used in the main sequencer design were designed in order to reduce the input clock frequency to the correct timings so the output value of each of the four digits was updated at the correct time, to ensure an accurate output of the main sequencer, whilst maintaining a fully synchronous design. Both the 10-digit and 5-digit counters have 4-bit binary outputs that change and output the current value for its corresponding digit every time the counter receives a logic high on its ‘input increment’ pin. The 10-digit counter outputs binary 0-9 and the 5-digit counter outputs binary 0-5. When the counters reach their highest value, they output a logic high on the ‘output increment’ that is connected to the next counter. This method was chosen to allow the correct sequence of numbers to be outputted by allowing each counter to increment the next one. The block methodology used to design the main sequencer unit was successful here, as it allowed the 10-digit counter design to be reused with the 2-digit counter, so a 20-digit counter was generated, reducing schematic and layout design time.

The OCU was designed to take in all of the counter output values, output the right signal to the display to power the correct digit, and send the corresponding digit value to the decoder. The OCU was designed to update each digit value in turn, with an operating frequency of 200 Hz the refresh rate of the 4-digit time value was 50 Hz. This is fast enough for the delay between updating the first and last digit of the number to be unnoticeable. The sequential logic was designed by creating an algorithmic state machine (ASM) that had four states, one for each digit that is updated. This was made into two flip-flops, with the outputs controlling multiplexers (MUXs) and combinational logic, which send the correct counter values to the decoder and power the correct display digit respectively. The MUXs were designed using combinational logic, and then converted into NAND in a similar fashion to the button synchronizer.

Both the main sequencer and button synchronizer functionality was validated using System Verilog test benches, as this allowed us to observe the output waveforms and improve our design accordingly.

2.3. Seven Segment Decoder
The seven-segment decoder was designed by analyzing the connections required to properly light up the seven-segment display. Since the seven-segment display chosen is common-cathode, it lights up when the input to the segments is at logic 0. A logic truth table was made to show all the bit sequences for the outputs and the equation used in the seven-segment decoder for each output was derived by using the K-map method. In the final design, NAND logic was used as this reduced the total gate count and reduced the amount wiring work required for the later layout design. In the final version of the decoder schematic as shown in Figure 1, there were only 17 gates in total. When the seven-segment decoder was simulated in T-spice, the output of the module correctly followed the logic truth table and outputted all the correct bits to control the seven-segment display.
In the original layout of the seven-segment decoder, the whole circuit was divided into two parts with connecting wires in the middle. However, it was discovered during the full chip integration that the design could not fit in due to lack of available space. As a result, this design was revised to reduce the size. The circuit was modified to a one-piece configuration in which all the logic gates were placed in a straight line. All the output wires were moved to the bottom of the circuit while input wires stayed on top. This reduced the difficulty of wiring as well as minimized the size of the circuit. Also, the arrangement of logic gates has been optimized, ensuring less amount of wires used. The seven-segment decoder layout is shown in Figure 2.

3. Integration
Once the schematic and layout were finished, the next step was to integrate the separate circuits onto one design so it could be fabricated on the finished chip. There was limited space to work with on the chip, therefore all the designed modules’ layouts were first reviewed and revised in order to reduce
their size. The major problem encountered during integration was the lack of space available to place correctly spaced tracks, which connected the modules to their corresponding I/O connections. Despite trying alternate approaches using both layers of the chip, a solution could not be found.

To obtain more available space, the main sequencer was reduced from a 4-digit design to a 3-digit design. However, there still wasn’t enough space for wiring. This change also meant that the logic in the main sequencer that controls the decimal point had to be edited to suit a 3-digit design. For the final integration, the main sequencer was removed from the chip design to allow space for the rest of the modules to be correctly wired up.

4. Post Fabrication Testing

4.1 Test Vector

As the seven-segment can only display the number from 0 to 9, test vectors were developed to test the functionality of the module. The test vectors checked how the system responded to stepping through the binary values from 0 to 9 and observing the output. By stepping through the binary values, all the input and output pins are checked for stuck at 0 and stuck at 1 faults.

Testing the button synchronizer was relatively easy. The first version focused on a functional test of the module, which was done by referring to the truth table and making a vector for each row. Once these were made, it was found that a lot of the stuck-at faults were also tested through these vectors. As the synchronizer contained a single flip flop and a small number of logic gates, only 13 vectors were required for an exhaustive test for every stuck-at-1 and stuck-at-0 fault on the inputs and outputs.

The main sequencer has inputs A13, A14, A18 and outputs Q7 to Q16. A13 is clock for main sequencer, A14 is reset and A18 is enable signal. Q7 to Q11 are the outputs which control the digit to update on the seven-segment display from digit 1 to digit 5, in this design Q11, the digit 5, is not used and should be 0 at all. The first digit shows the minute of the stop watch, the second and third digit indicate seconds, and the fourth digit gives one tenth of a second. Q12 is the decimal point signal for seven-segment display. As the requirement, the first and third decimal points should always light up during counting to indicate the separation of different units. Q13 to Q16 are the binary outputs of the number wanted on the seven-segment display, and these signals are also the input for the seven-segment decoder. The test vector is a functional test vector, as it covers any stuck at 1 and 0 faults on both the input and the output, reducing the need for a fully comprehensive test. The first set of test vector is used to reset the main sequencer, with no “Clock” or “RunIn” signal applied. The rest of the test vectors focus on observing the output of the 4-digit main sequencer when the input “RunIn” is applied.

4.2. Simulation

By exporting the S-Edit schematics into Verilog files we could test each module with its corresponding test bench in ModelSim. By utilizing ModelSim as our simulation platform, it allowed us to observe the errors generated by the test vectors when they are applied to the modules, along with the waveforms generated by the input and outputs of each module.

Using the exported Verilog test benches, all the modules were tested to observe any errors with either the module or the test vectors. When the test vectors were applied to the submitted design for the button synchronizer, errors in the output were found through the ModelSim simulation. It was discovered the button synchronizer did not work as expected. When the inputs Go and Stop were applied at the same time, the input Stop should overwrite the input Go, so the output Run would not be asserted. What was found however, was that the Run was being asserted when Go was asserted after Stop. Through more testing, the error was found to be that both inputs can overwrite each other. Once this error was discovered, the schematic of the button synchronizer was fixed and this change was carried through to the golden design. The rest of the vectors that tested the remaining modules worked as expected. The full unified set of 1236 vectors, after including the fixes previously explained, also passed as expected.
4.3. Chip Test

To test the stopwatch, three push buttons with pull-up or pull-down resistors were used for Reset, Go and Start respectively and connected to the corresponding input pins on the chip. The seven-segment display was powered and connected with the output pins of the chip. This test setup for the stopwatch is shown in Figure 3. The A to G and Di ports of the two 7-segments were linked on the breadboard to test the 5-digit 320 Hz stopwatch from other chip. The main sequencer in other chip was used to test the function of the circuit. The 4th digit of the first seven-segment was used as the first digit of the stopwatch. In that case, the remaining 2nd to 5th digit for the stopwatch used the 4 digits on the second seven-segment display. Since there was no inner connection between the seven-segment decoder and the main sequencer, they were wired together manually. Therefore, the stopwatch began to run as the Go button was pressed. The stop button, when pressed, successfully held the value on the seven-segment displays and paused the stopwatch and the Reset button started over the stopwatch from “00.00.0”.

Another test was done using the 3-digit version of other chip. The first 7-segment was removed since there were only 3 digits that needed to be displayed. After turning the site switch to the corresponding number and set the clock frequency to 160 Hz (required frequency of 3-digit stopwatch), the stopwatch worked when the Go button was pressed. It was noticed that overflow occurs when reaches maximum value of 99.9 seconds and the displayed number would return to 00.0 and continue counting. Both stopwatches were observed to have accurate timing, within 1% accuracy. Since this design was lack of main sequencer on the chip, a CPLD with the total Verilog code generated from our design was needed for the purpose to see if the stopwatch is working.

The CPLD that used for testing was the Il Bagatto Max V 5M240Z100CT. Through the software Quartus, the Verilog code of the total design which generated from S-edit was transferred to a .svf format file so that it could be loaded to the CPLD [2]. The pin arrangement on the Il Bagatto were set

![Figure 3  Setup of test circuit](image-url)
randomly after compiling. Some of the pins need to be rearranged in the pin-configuration step as shown in Appendices E. After rearranged, the whole Verilog file was compiled, and downloaded onto the CPLD. This circuit works properly.

5. Conclusion
This paper describes a complete digital stopwatch circuit with chip implementation. The design schematics of the chip were good, with only one error in the button synchronizer that was corrected in the final golden design. However, during implementing the physical layout of the various circuits, it was found that it was not possible to put all the modules on the allocated space on the chip. The final design for fabrication excluded the main sequencer module of the stopwatch circuit. The results from the post fabrication testing were more successful. The unified test vectors identified a single error with the chip to do with the response of the button synchronizer when both Go and Stop are both pressed, and this error was fixed in the final golden design submission. An attempt was made to implement our golden design onto a CPLD. This would verified that our stopwatch can physically work, instead of just working in simulation. Overall, this design is a reliable and cost-effective solution for digital stopwatch devices. For future improvement, this design can be refined to extend the maximum timing length or improve the precision to 0.001 second.

References
[1] Wang Y. Design and implementation of digital stopwatch based on FPGA[J]. Microcomputer & Its Applications, 2014.
[2] Senthil S M, Prabu R T, Jayanandan I. Design of Digital Clock Calendar Using FPGA[C]// IETE 45th Mid Term Symposium on Broadband-Technologies and Services for Rural India MTS 14, 2014.
[3] Yang Y, Zhao C, Lei J. Design of Digital Stopwatch Based on FPGA[J]. Modern Electronics Technique, 2008.
[4] Han Z X. Design of digital stopwatch based on FPGA[J]. Journal of Shaanxi Institute of Technology, 2004.
[5] Wang X, University S. The Design of the Electronic Stopwatch Based on FPGA[J]. Journal of Taiyuan Normal University, 2014.
[6] Singh B, Chodha A, Sharma B, et al. Design of Low Power Digital Clock on FPGA using Different IO Standards[J]. Indian Journal of Science & Technology, 2016, 9(21).