Peter Minzicu
CSC 34300
3/25/2014

Lab 6 Memory and a Timer

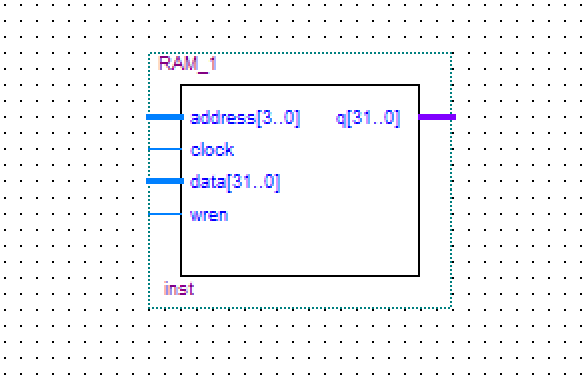
Introduction

The purpose of this lab is to demonstrate how a timer functions and how random access memory works. A timer is an integral part of society today and it is important to know how it functions. With the help of the DE2 board, we will build a timer from scratch. We will use the internal clock of the DE2 board to keep time, and counters and comparators to measure it. Finally the DE2 board will be programmed to store a current time and display it.

Process

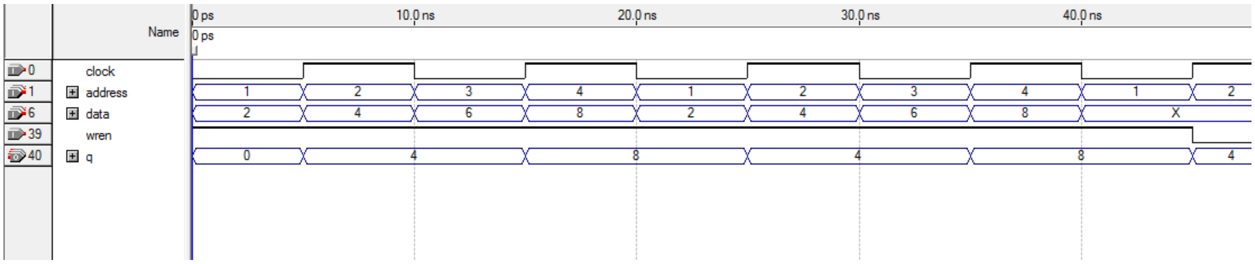
The first step is to become familiar with Quartus' megafunction wizard. The wizard is a tool used to automatically make logical or mathematical systems. When setting up the timer, we can use these functions to streamline the making of the timer. The first component that will be made is a LPM memory module. The purpose of this system is to store a 32 bit integer. The 32 bit integer in our case is a snapshot of the current time. Because a megafunction is being used, Quartus will generate the block symbol file, the VHDL file and a waveform. In this lab however, the waveform will be created manually.

Here is the block diagram for the memory module.



The inputs are an address, a clock signal, a data signal and a write enable signal. The only output is a 32 bit binary output. The address input is a way to select the different memory blocks in the module. The LPM has 4 different memory locations. Depending on the input, a different location will be chosen. The data input is the 32 bit data input that is stored. The WREN pin stands for Write Enable. When this input is high, it will enable the module. When it is low, the module will be disabled. Finally the output will be connected to multiplexer which will control when the 7 segment display is activated.

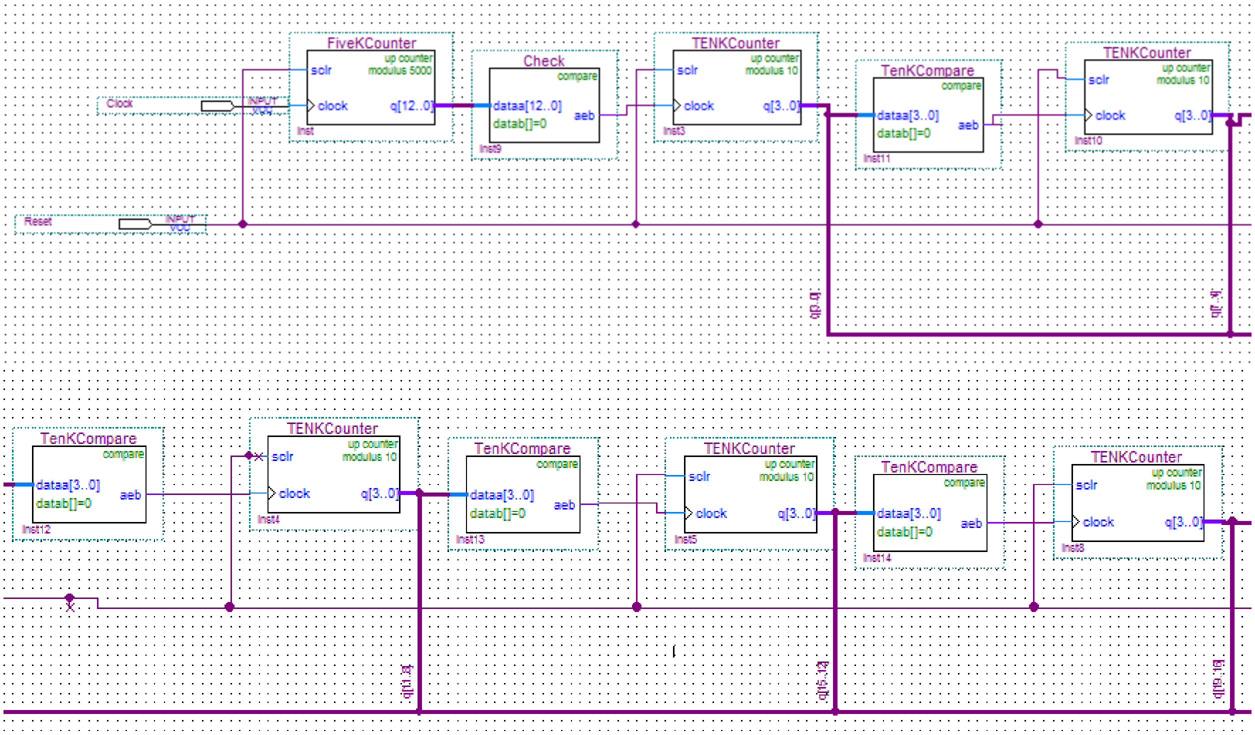
Here is the test waveform of the file.

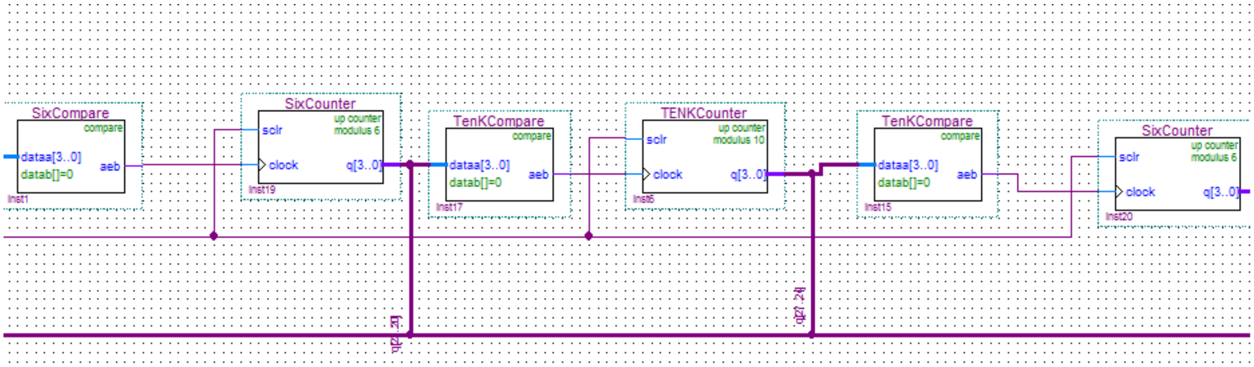


Depending on the address selected, the output will show the value stored. In this simulation, the write enable is always high.

In Part II of this lab, a counter had to be designed. A counter or stopwatch is a device that measures time that has passed. It starts at 00:00:00 when the start button is pressed and ends when the start button is released. The stopwatch that will be designed will show, milliseconds, seconds and minutes. These will be displayed on the 7 segment displays on the DE2 board. To create this, certain components must be made.

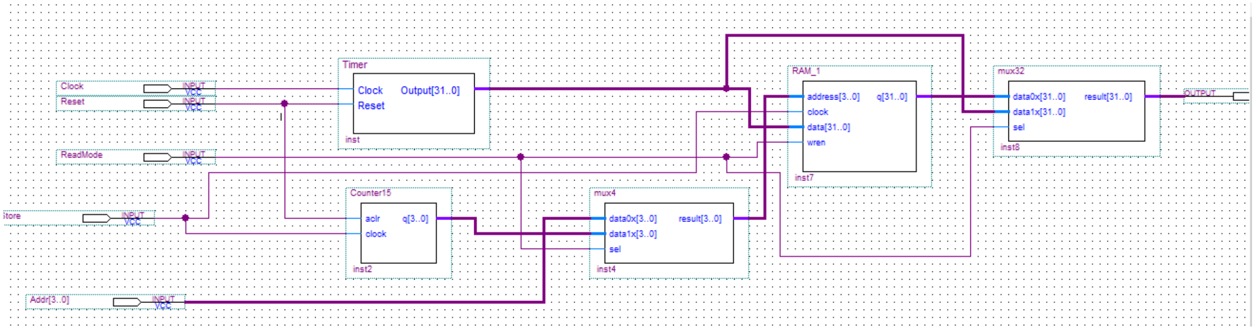
The first is a timer circuit.





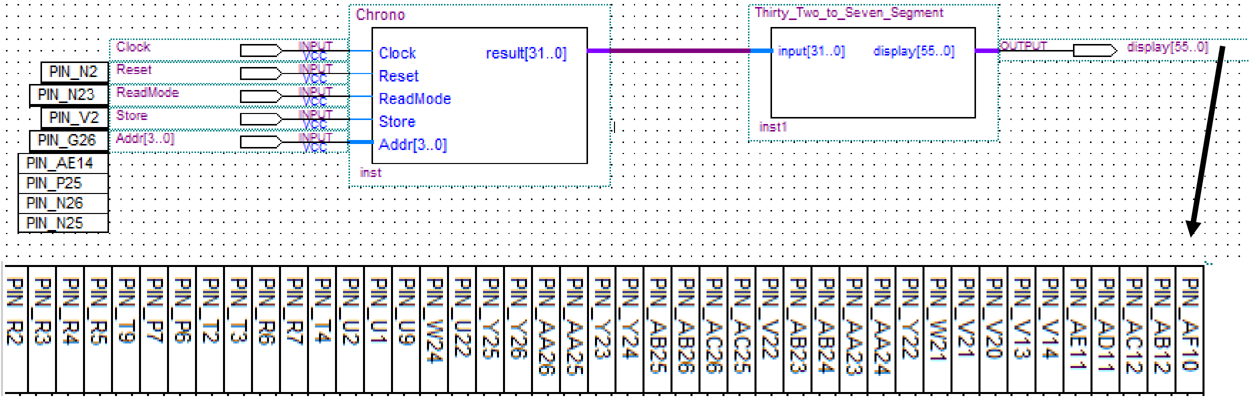
Each screenshot is connected together. Different modulo counters and comparators are daisy chained together to make the full timer. This circuit is necessary because of the DE2's internal clock speed. The DE2 uses a 50mhz clock signal. This means it can perform 50,000,000 clock rises per second. To control the first display, we need to scale the clock so that a value will increment once every 10,000 times. This represents a millisecond. When a millisecond passes, the display on the board will increment by 1. To control when the display increments, we use a 5000 modulus counter. When the number 5000 is counted to, it will start back from zero. The next component is a comparator. It checks to see if a value is equal to 0. If it is, it will output a high signal and activate the next component. If the input is a 0, it means the specific time amount has passed and the next unit must be incremented. The 5000 modulus counter is only used once. After that, the time is scaled to the correct speed. After this there will be a series of 10,000 modulo counters and modulo 6 counters. The 10,000 counter is used for scaling the next units of time. It is used from going to milliseconds to seconds. However, there are 60 seconds in a minute. Therefore when incrementing from seconds to minutes, the display must increment after 6. The 10,000 and 6 comparators work the same by activating the next component after their max value is reached.

Once the timer circuit is created, an additional circuit needs to be created to store selected times.



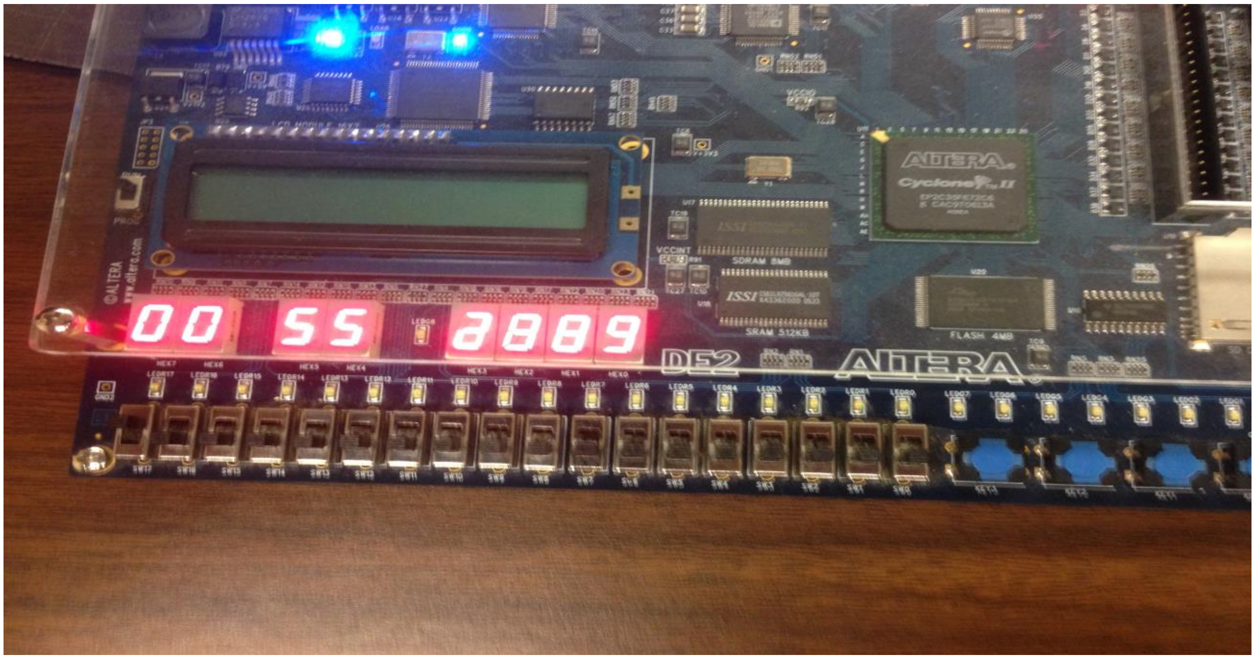
As part of the lab instructions, the stopwatch must also be able to store time when requested. Therefore a circuit must be made that has a memory component and modulo counters. The inputs are the clock signal from the DE2 board, a reset function to reset the time to 0; an address input to select a memory location, and, a read write switch. The read write switch will set the system to show elapsed time or show a stored time. The output size is 32 bits and will connect to a converter circuit to show values on the DE2 board. The multiplexers switch outputs to the display when the system is set to read or write.

Finally we combine all these systems together so we can show the time on the DE2 board.



The symbols are compiled and made into block files. Then their outputs are generated and assigned. The 32 to 7 segment display is given in the lab as a VHDL file. It is compiled by Quartus and made to a block symbol. The pin assignments from the symbol I/O ports to the DE2 board are also given. They are imported to Quartus in a text file. The reset and store pins are activated with blue pushbuttons on the DE2 board. The readmode and memory address inputs are created using toggle switches on the DE2 board. The output of the stopwatch (Chrono) is fed to the display converter. The output of the converter is 56 bits for the 7 bits for 8 displays.

Here is the final result.



Shown on the DE2 board are a total elapsed time of 55 seconds. The 4 displays on the right are blurry since they are incrementing very fast.

Conclusion

Overall the board functioned properly. When the system was set to write by the toggle switch, it began showing elapsed time just like on a stopwatch. The store button was then pressed 4 times. Right after, the board was switched to read mode. By selecting the toggle switches to the proper memory location, we were able to read the 4 different times stored when the store button was pressed. Finally, the reset switch allowed us reset the time to 00:00:00 when the button was pressed. One improvement I would make is an additional switch that slows down the millisecond displays. While it would inaccurately show the current time, it would allow us to see if the assigned displays are functioning properly.