Memory Transfer with Controller

Venkateswara Raju Datla#1, Suhani Vaishnav\*2

#Computer Engineering Department, San Jose State University  
San Jose California

1venkateswararaju.datla@sjsu.edu

3suhani.vaishnav@sjsu.edu

Abstract— This document delivers how to implement Memory Transfer with the help of a controller. This is a top module which is generated by combining the multiple modules in Verilog. The small modules which are used in this memory transfer block include Memory blocks, Arithmetic Logic Unit, Counters and Controller.

Keywords— Clock, D Flipflop, Address of the Register, Counter, ALU Unit, Comparator and Reset.

1. Introduction

The design block reads two eight bit packets from an 8x8 source memory (memory A), processes them in the ALU and stores the result in an 8x4 memory (memory B). The processing part depends on the relative contents of each data packet: if the contents of first data packet are greater than the second, then contents of data packets are added. Otherwise, they are subtracted from each other. The result is the transferred to memory B.

1. Modules

Several individual modules combine together to form a memory transfer (top module). These individual modules are as follows:

1. Controller
2. Counter A
3. Memory A
4. ALU
5. Counter B
6. Memory B

1. Controller:

The inputs of Controller include clock and reset. The outputs include INCA(increment), INCB, WEA(Write enable A), WEB. The controller timing diagram is shown in Fig 3. The controller is implemented using a counter-decoder. Fig-6.

1. Counter A

The inputs of Counter A include clock, reset and INCA. The output of Counter A is addrA which indicates the address of registers in memory A. Once the INCA is high, the counter starts counting from 0 to 7.

1. Memory A

The memory block A is an 8x8. The inputs of Memory A include AddrA, clock, WEA and datain (which is the input data). The output of Memory A include dataout1. Fig 4 shows the timing diagram of Memory A.

1. ALU (Arithmetic Logical Unit)

The inputs of this block are dataout1 (which is the output of memory A) and clock. It contains D flip flop with input dataout1 which is operated during positive edge of the clock, the output of d flip flop is dout2. The comparator with inputs as dout2 (A1) and dout1(A0) compares the values and performs addition or subtraction operation based on sign bit.

If A0 is less than A1 then the sign bit is 1 hence, addition. If the A0 is greater than A1 then sign bit is 0. The output of ALU is datainB. Fig 2 shows the timing diagram of ALU.

1. Counter B

The inputs of counter B include clock, reset and incB. The output include AddrB which is the address of registers in memory B. It is a 2 bit counter and it runs when INCB is high.

1. Memory B

The memory block B is 8x4. The output of ALU is the input for memory B datainB. Fig 5 shows the Timing diagram of Memory B.

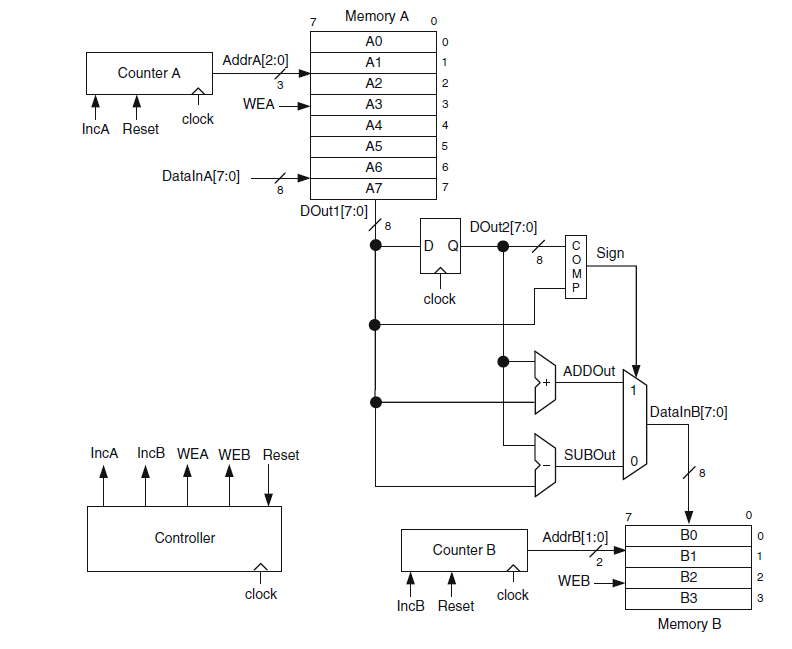


Fig : Block diagram of the Memory Transfer

1. Top Module

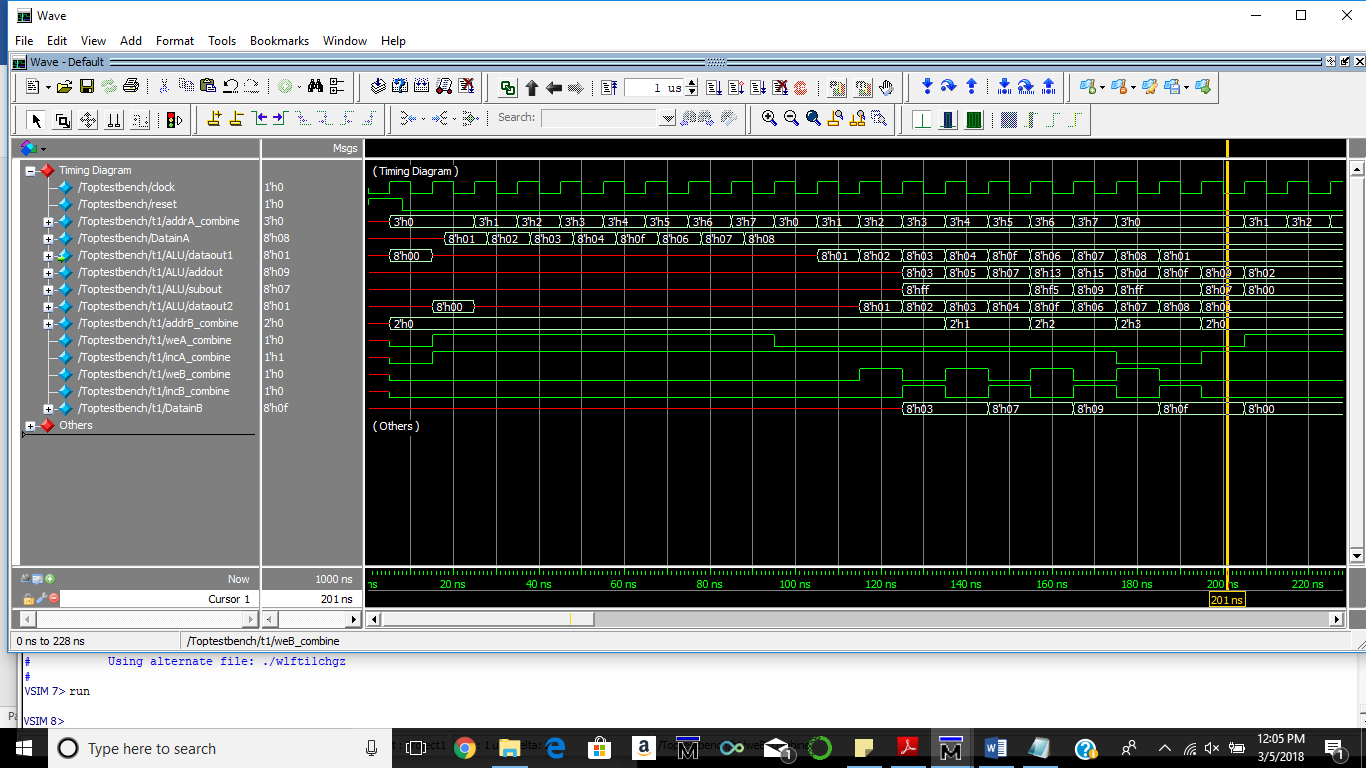
The Input of top module is clock, reset and datain. Initially, counter A generates the addresses, 0 to 7, for memory A and writes the data packets, A0 to A7, through DataInA[7:0] port. When this task is complete, counter A resets and reads the first data packet A0 from AddrA[2:0] = 0 in clock cycle 9. In the next clock cycle, A0 becomes available at DOut1, and the counter A increments by one. In cycle 11, AddrA [2:0] becomes 2, the data packet A1 is read from DOut1[7:0], and the data packet A0 transfers to DOut2[7:0]. In this cycle, the contents of the data packets A0 and A1 are compared with each other by subtracting A1 (at DOut1) from A0 (at DOut2). If the contents of A0 are less than A1, then the sign bit, Sign, of (A0 – A1) becomes negative. Sign = 1 selects (A0 + A1) and routes this value to DataInB[7:0]. However, if the contents of A0 are greater than A1, (A0 – A1) becomes positive. Sign = 0 selects (A0 – A1) and routes this value to DataInB[7:0]. The result at DataInB[7:0] is written at AddrB[1:0] = 0 of memory B at the positive edge of clock cycle. In the same cycle, A1 is transferred to DOut2[7:0], and A2 becomes available at DOut1[7:0]. A comparison between A1 and A2 takes place, and either (A1 + A2) or (A1 – A2) is prompted to be written to memory B depending on the value of the Sign node. Since A1 is used in an earlier comparison with A0, A1 cannot be used in a subsequent comparison with A2, and neither (A1 + A2) nor (A1 – A2) should be written to memory B. Fig 1 shows the Timing diagram of Top Module.

Conclusions

This design block combines the data-path and controller design concepts. It also introduces the use of important sequential logic blocks such as flip-flop, register, counter and memory in the same design. By combing multiple modules, the top module for memory transfer is implemented..

References

1. Ahmet Bindal, *Fundaments Of Computer Architecture and Design*, 5th  ed.



**Fig 1 Timing Diagram of Top Module without reset**

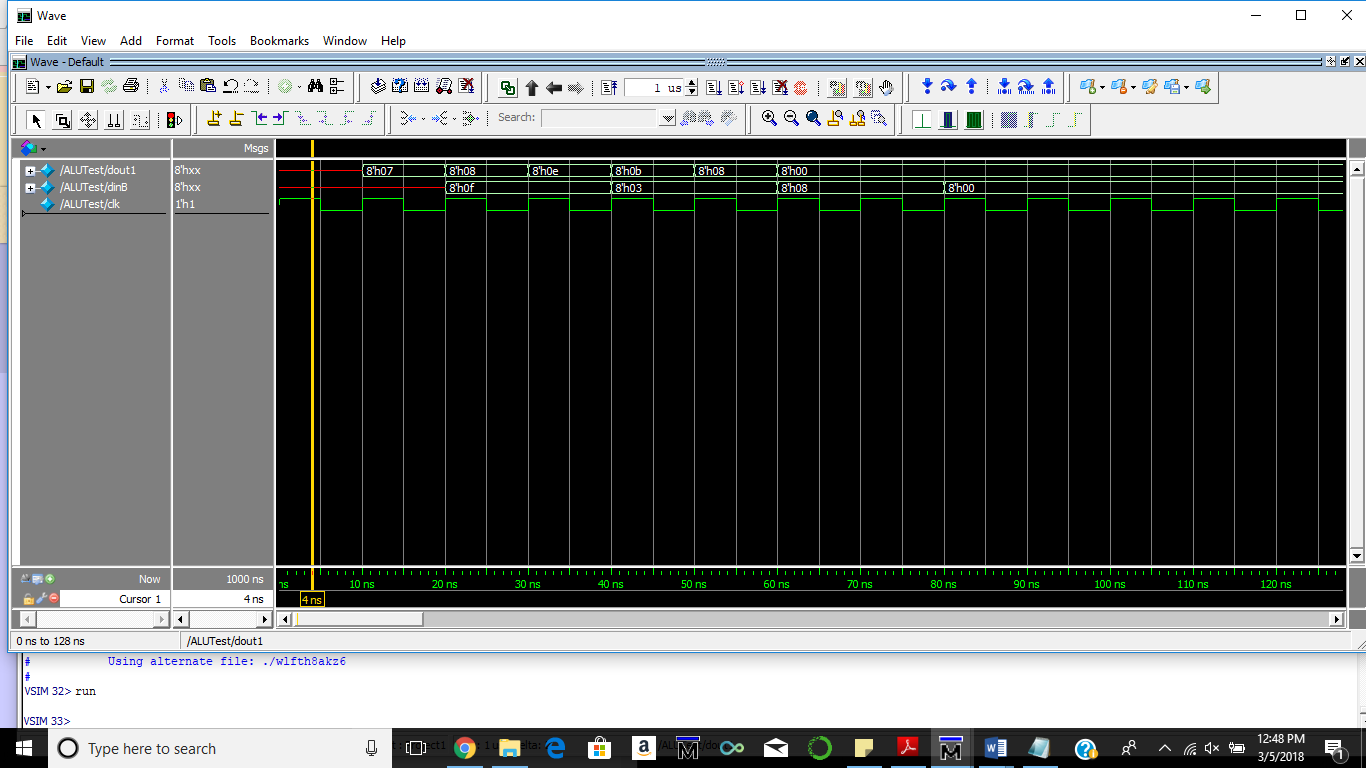


Fig 2 Timing Diagram of ALU.

Here, the ALU takes in data from the memory (dout1) and compares it with the next data(dout2) from the memory. If dout1<dout2, then addition is performed and if dout1>dout2 ,then subtraction is performed and the output is visible in figures as dinB.

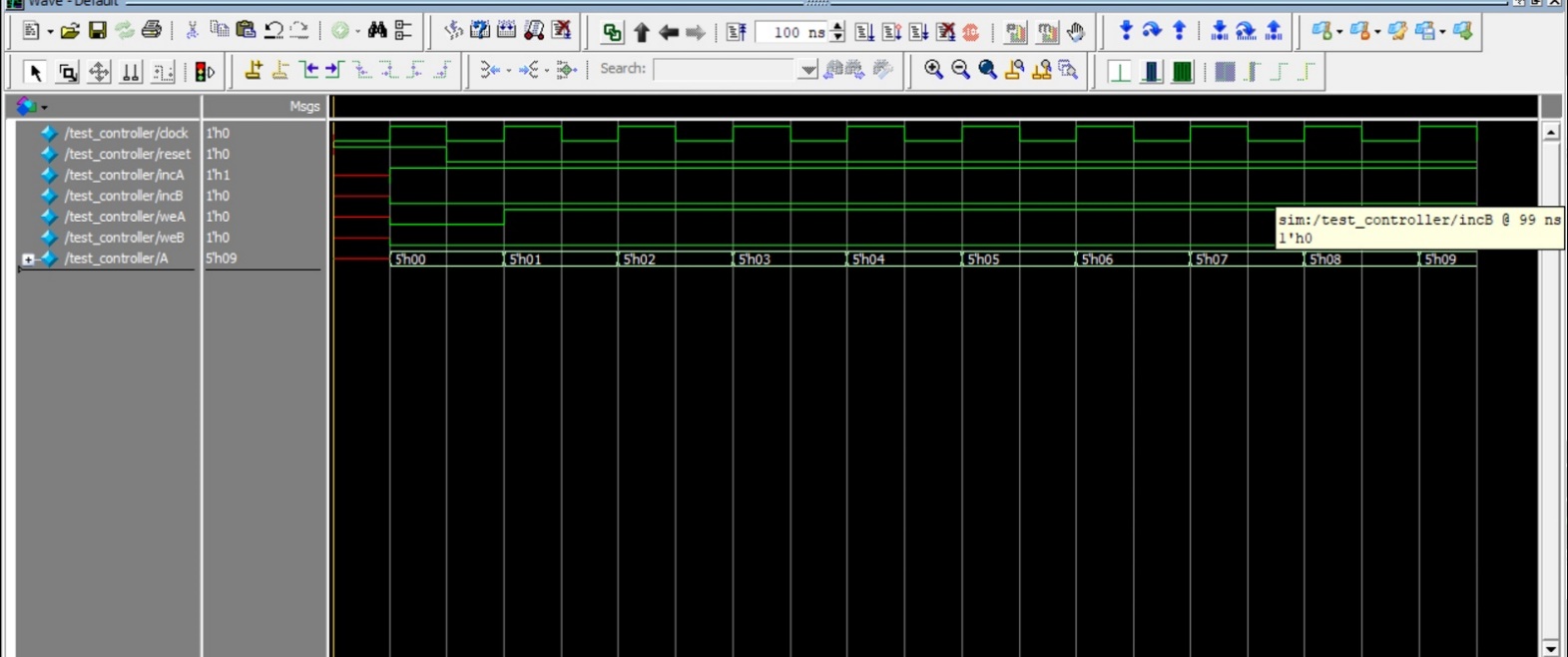


Fig 3 Timing Diagram of Test Controller.

Here depending upon the counter outputs, the values of IncA, IncB, WeA and WeB change.

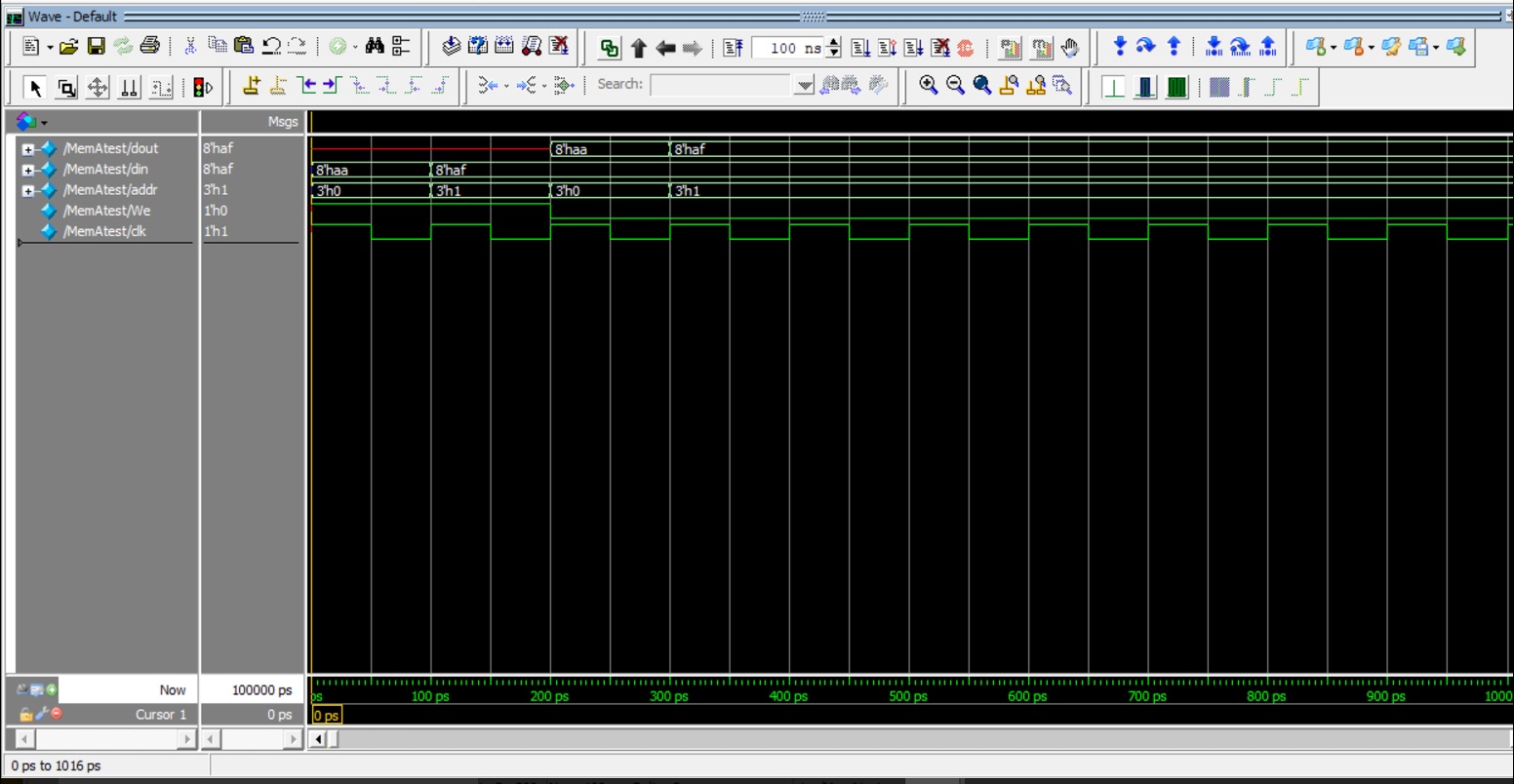


Fig 4 Timing Diagram of Memory A

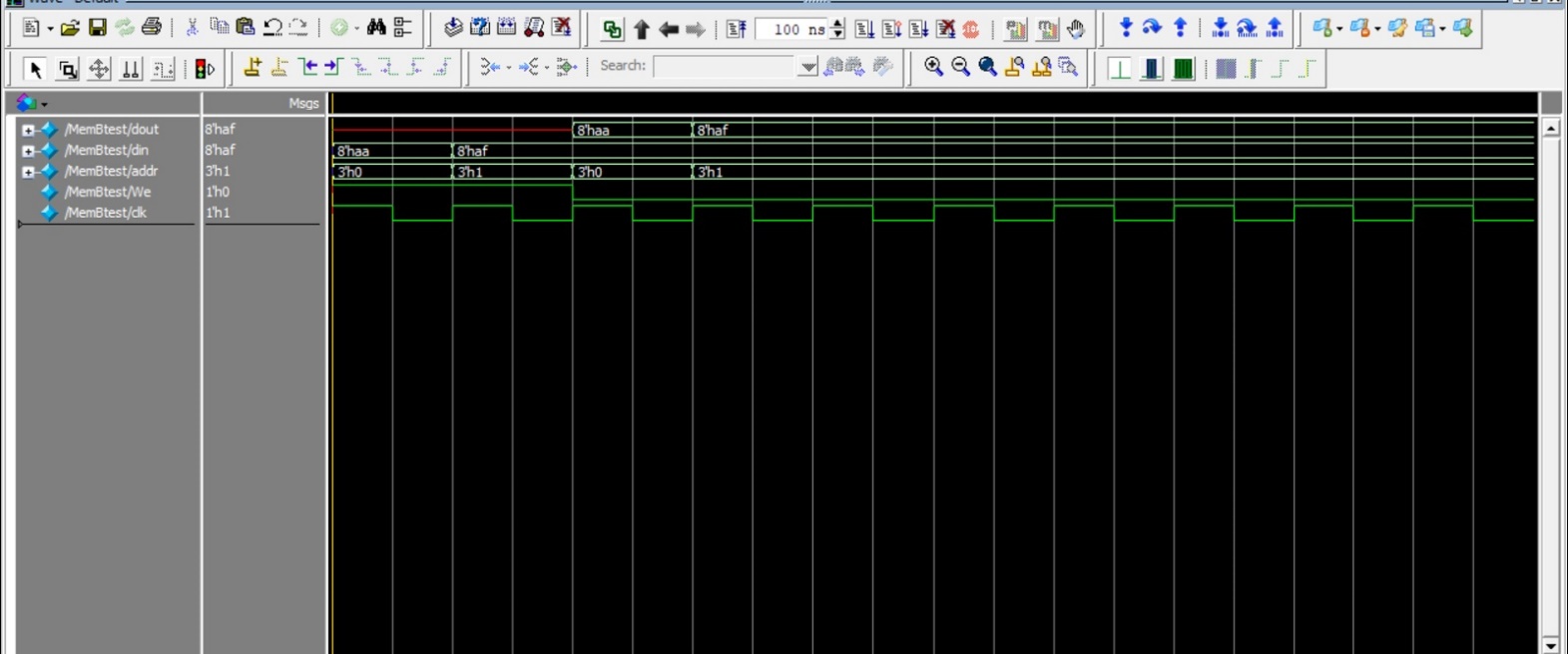


Fig 5 Timing diagram of Memory B

These two diagrams show working of the memory-A and memory-B based on the status of Write-enable signal (generated by counter-decoder).

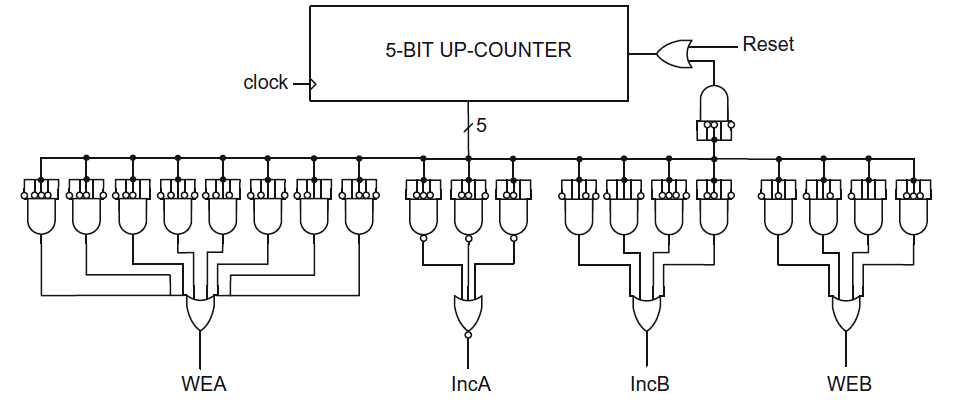


Fig 6: Counter Decoder implementation for Controller

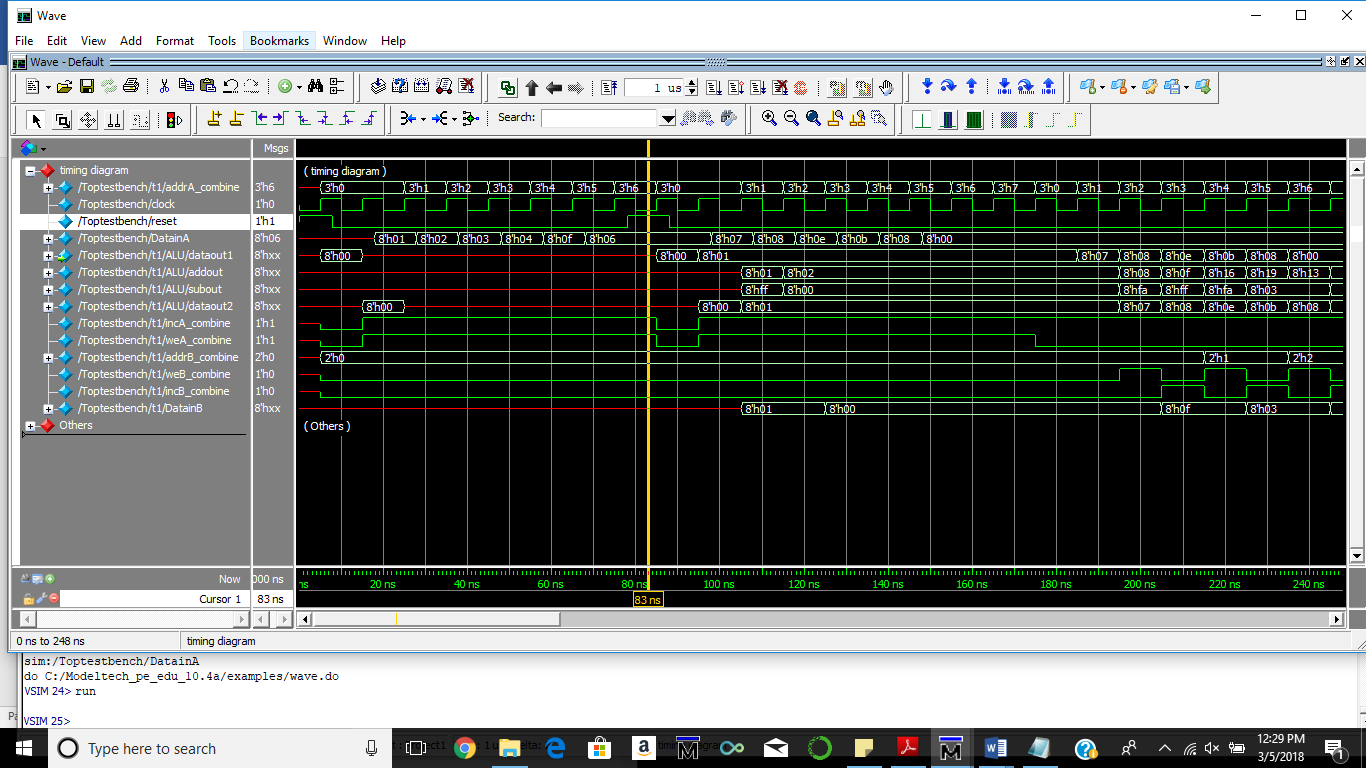


Fig 7: Timing diagram of the module with reset condition at 8th clock cycle.