**G19\_Barrel Shifter**

In this lab, we were asked to design a 14-bit Barrel Shifter which base on three 14-bit, 2input, Bus Multiplexer. This device will shift on input vector by a certain number of bit position according to the select signal. (1) The 14-bit Multiplexer were built by combining 14 2-bit multiplexers which is represented by the following equation and truth table.

|  |  |  |  |
| --- | --- | --- | --- |
| S | X | Y | Out |
| 0 | 1 | 1 | 1 |
| 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 0 |
| 0 | 1 | 1 |
| 0 | 0 |

**Table 1**

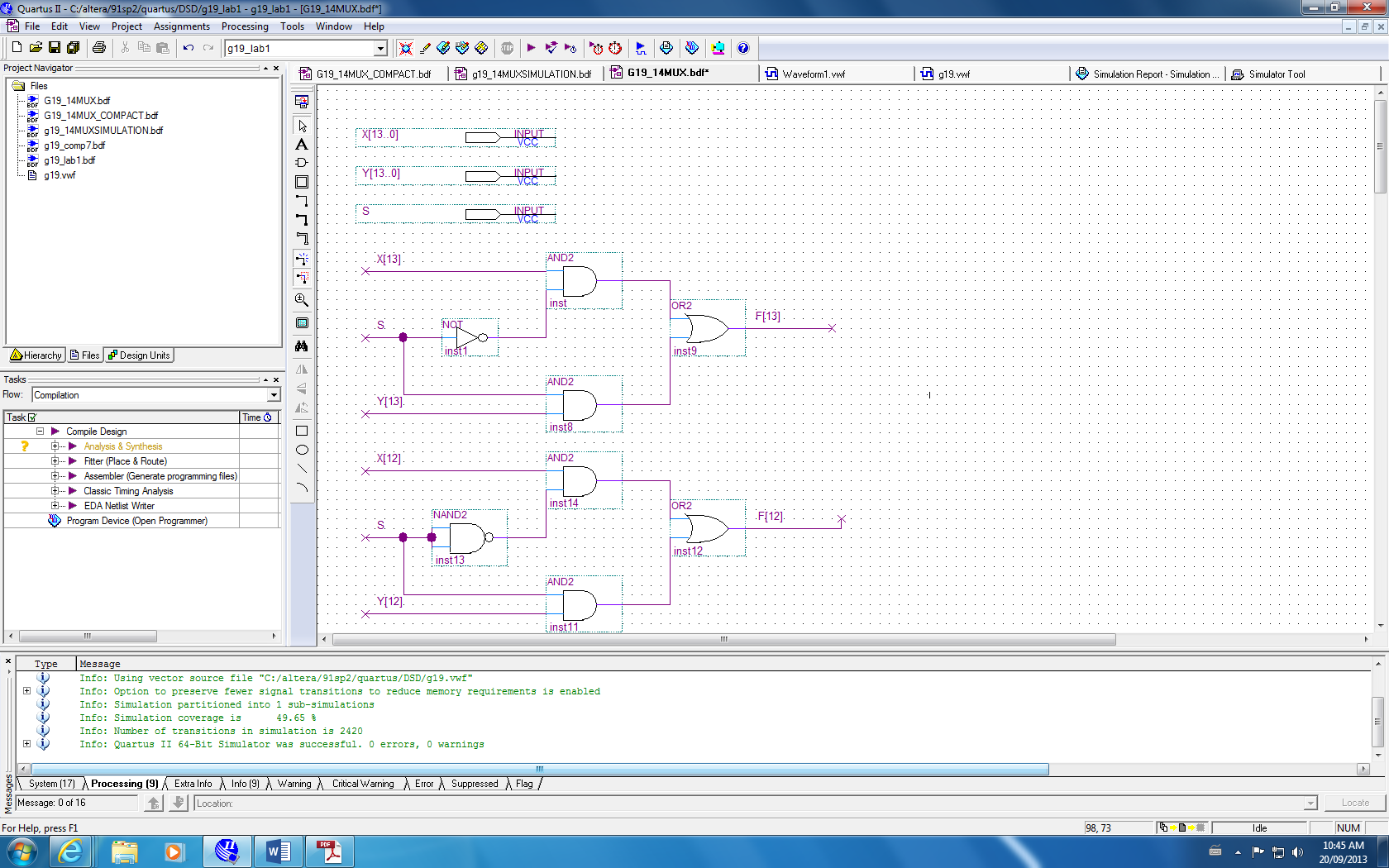
The 2-to-1 Multiplexer are formed by 2 AND gate, 1 OR gate and a NOT gate. It selects which input signal (X,Y) go through the output depending on the signal S (selector). The diagram below(Fig. 1) shows our design of the 2-to1 Multiplexer with input X, Y and S which are all represented by the truth table and equation above(Table 1).

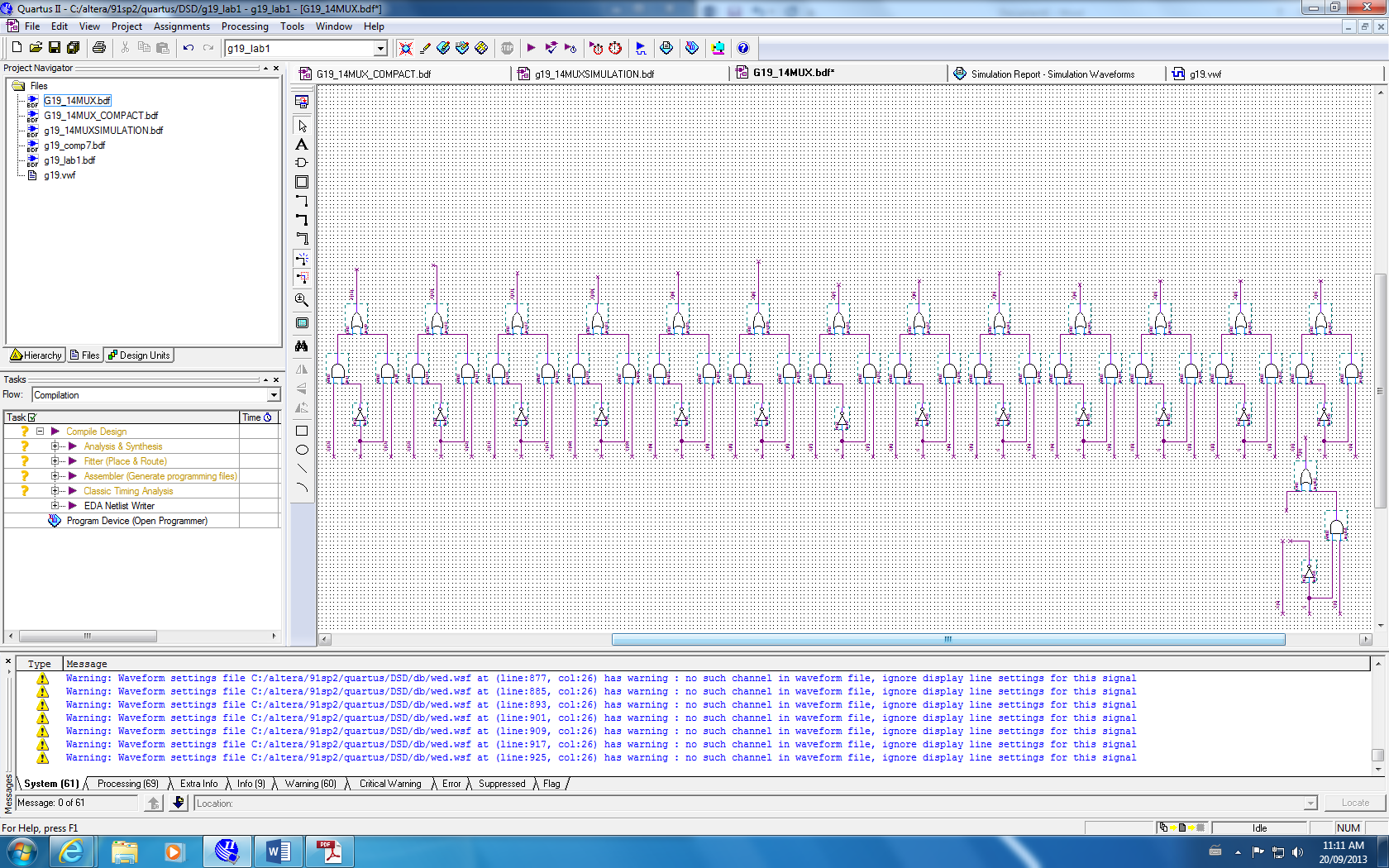
|  |  |
| --- | --- |
| Inputs | Output |
| X[13..0] | F[13..0]  F → X when S=0  Y when S=1 |
| Y[13..0] |
| S |

**Table 2**

Each MUX has 2 inputs (X and Y, where i=0 to 13) and a selector. The selector will choose whether the output (F, where i=0 to 13) is X or Y. In other terms, we observed that when S=0, then F=X, but when S=1 then F=Y (Table 2).

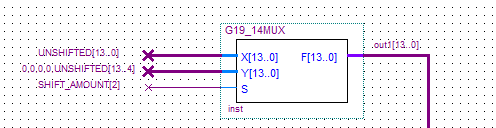
In order to build a 14-bit multiplexer, we connected 14 2-to-1 multiplexer with same design as below to the same output which means the new 14-bit multiplexer will take 2 14-bit inputs, 1 1-bit selector signal input and result in a 14-bit output(Fig. 1). Xi is AND’ed with the S’ and Yi is AND’ed with the S. Depending on the value of the selector, X and Y will be chosen as the output of the AND gates and are then OR’ed to give the selected input (X or Y). Also, the 2-to-1 MUX is connected 14 times in order to build 1 Bus Multiplexer of the Barrel Shifter (Fig. 2).



**Figure 1**

**Figure 2**

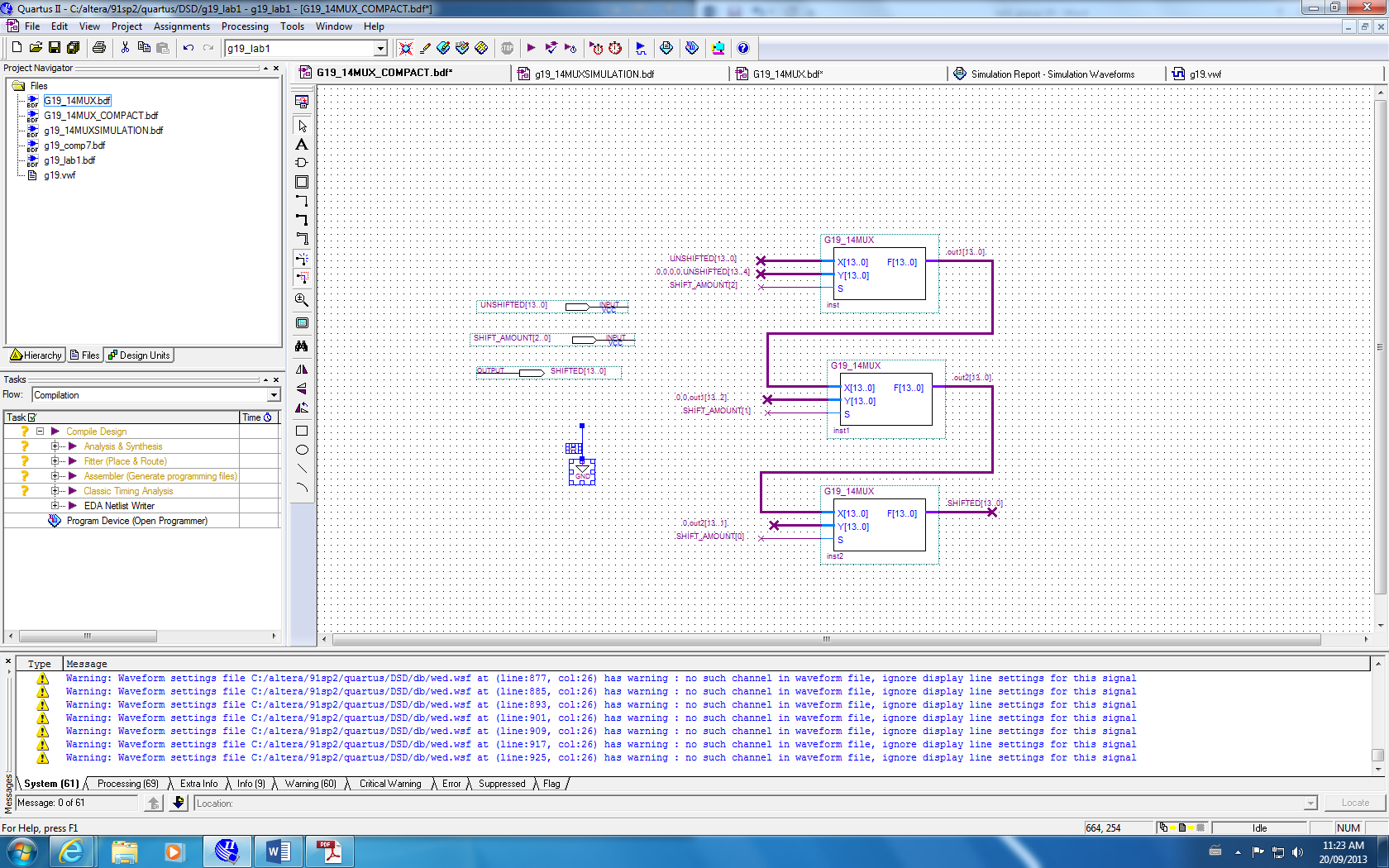
The requirement of this lab stated that the final design of our barrel shifter only needed to shift right, and the maximum number of bit places to shift were limited to 7.(2) This suggested that we need three 14-bit MUX (since the selector is 3-bit = 7) and each 14-bit MUX responsible for different bits of shifting(4,2 and 1).



**Figure 3**

The Barrel Shifter consists of 3 multiplexers and each of them has 14 2-to-1 multiplexers, each 14-bit MUX will take three inputs(Fig. 3):

* Un-shifted 14-bit input
* A “shifted” un-shifted 14-bit input
  + 1st MUX shifted amount = 4 bit (add 4 0s in the front and cut the last 4-bit off)
  + 2nd MUX shifted amount = 2 bit
  + 3rd MUX shifted amount = 1 bit
* And a single bit shift amount as a selector.
  + The selector for the 1st MUX will be from the most significant bit of the 3bit shifted amount, the 2nd MUX will take the second bit from the shifted amount as selector and the last MUX will take the last bit as the selector.

Therefore, we implemented as a sequence of multiplexers in a way that the output of one of the multiplexer is attached to the input of the next multiplexer hence output (shifted or not shifted) can be reused and shifted data will not be lost. **Figure 4**

­

The following table shows gives three examples and show how input changes on each stage of the multiplexer by having different SHIFT\_AMOUNT value.



For example 1, the shifted amount is 011. Therefore on the MUX1, the shift amount input will be 0 hence the un-shifted input will be chosen as the output. Then at the MUX2 and MUX3, the shift amount is 1 so 0, 0, Un-shifted (output shifted by 2) and 0, Un-shifted (output shifted by 1) will be chosen for out2 and out3 respectively. This shows that the amount will be shifted will be 3 which is the same as the Shift amount in the specifications.

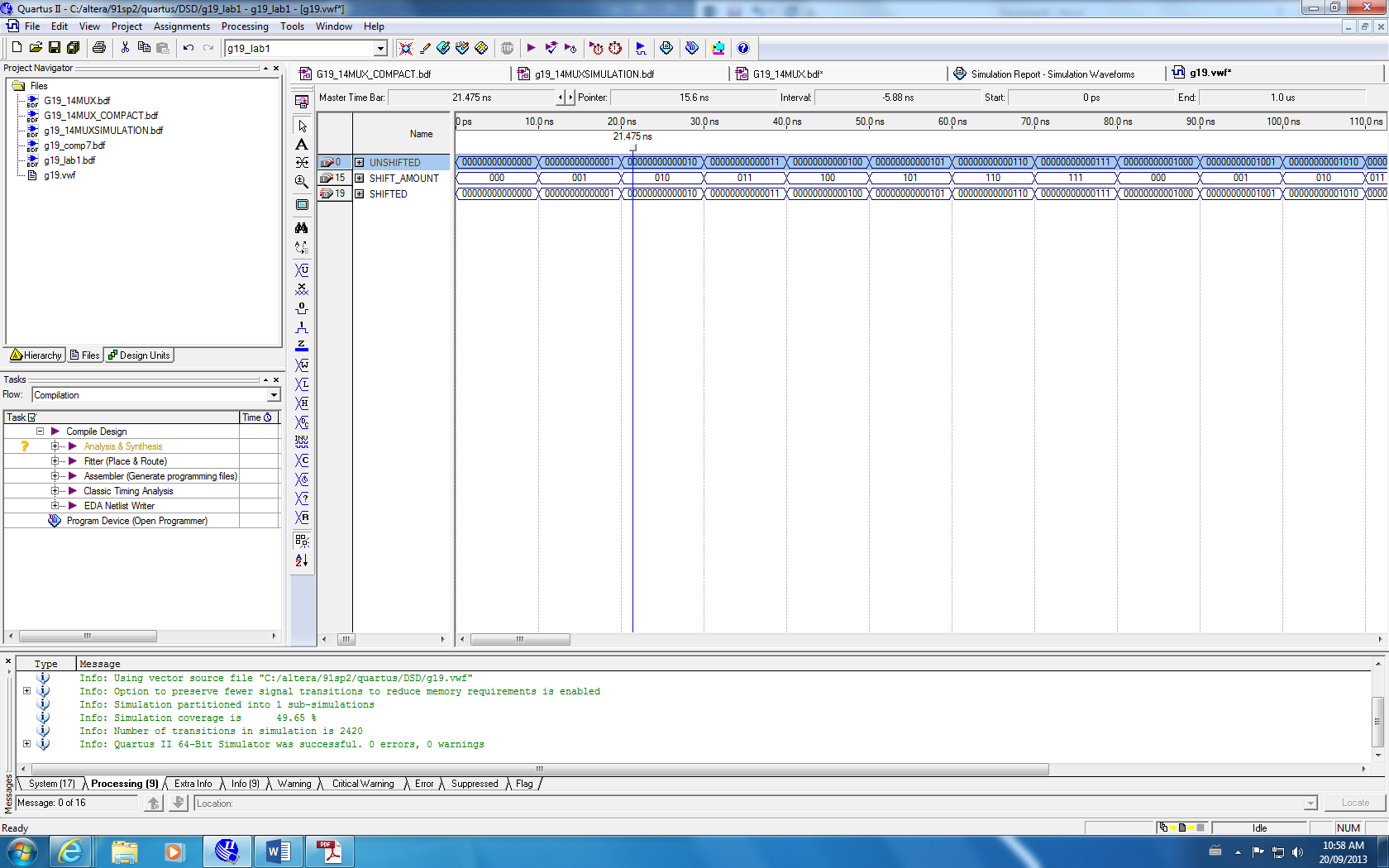


Example 2 and 3 go through the same process as example 1 but with different shifted amount which is lead to different MUX have different output (choose between shifted output or not) which is show above.

Our team created 3 Bus Multiplexer with one 14-bit input called UNSHIFTED, a 3-bit input called SHIFT\_AMOUNT and SHIFTED(Table 3). Also, a timing diagram was generated to test the Barrel Shifter design. The simulation demonstrates that our circuit is capable of shifting the 14-bit binary vector to the right in one clock cycle (Fig. 5):

|  |  |
| --- | --- |
| Inputs | Outputs |
| UNSHIFTED[13..0] | SHIFTED[13..0] |
| SHIFT\_AMOUNT[2..0] |

**Table 3**

**Figure 5**

Here is a truth table which demonstrates how three different inputs are related. The shifted amount will determine the amount that the UNSHIFTED input will be shifted.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UNSHIFTED** | **SHIFT AMOUNT** | | | **SHIFTED** |
| 10100100001010 | 0 | 0 | 0 | 10100100001010 |
| 10100100001010 | 0 | 0 | 1 | 01010010000101 |
| 01010010000101 | 0 | 1 | 0 | 00010100100001 |
| 00010100100001 | 0 | 1 | 1 | 00000010100100 |
| 00000010100100 | 1 | 0 | 0 | 00000000001010 |
| 11111111111111 | 1 | 0 | 1 | 00000111111111 |
| 00000111111111 | 1 | 1 | 0 | 00000000000111 |
| 10101010101010 | 1 | 1 | 1 | 00000001010101 |

**Table 4**

**Reference:**

1. McGill University ECSE-323 Lab 1 instruction page 62 by Prof. J. Clark.
2. McGill University ECSE-323 Lab 1 instruction page 63 by Prof. J. Clark.