**CEG2136 Computer Architecture I**

**Lab3**

**Arithmetic Logic Unit**

**Professor: Voicu Groza, P.Eng.**

**Dr. Fadi Malek**

**TA: Harman**

**Nayma**

**Sonal**

**University of Ottawa**

**Faculty of Engineering**

**2019 FALL**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Group Number: 8**

**Student Name: Jiajie Xu**

**Student Number:7881937**

**Student Name: Yilin Yang**

**Student Number: 8726747**

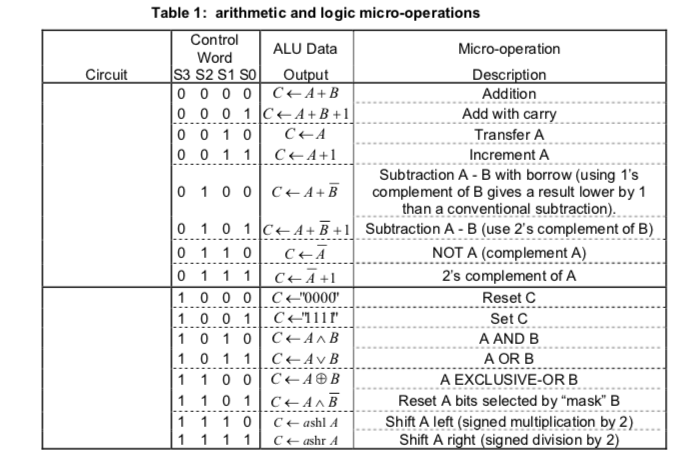
**Theoretical Part**

**1. introduction to the lab:**

Designing a basic 4 bits ALU was the task of this lab. The ALU was required to be able to do both Arithmetic and logic operations. The circuit of each was complemented separately according to the function table provided in lab instruction. Circuits were then integrated into blocks for final ALU circuit designing as a part. After signing each input and outputs of the final circuit and filling out the sequence simulation table, it was tested on DE2 -115 Board to see if the circuit works correctly by comparing the output data with the table. The designing structure played a significant role since the final circuit consists of a large number of logic components and connections. It would save the amount of work and be easy for debugging when a miss connection takes place.

**2. Discussion of the problem (specifications/requirements, diagrams, flowcharts)**:

A 4-bit ALU circuit was required to be implemented. The circuit should be able to do arithmetic and logic micro-operations. 4 control inputs were needed and the operation table is shown below.



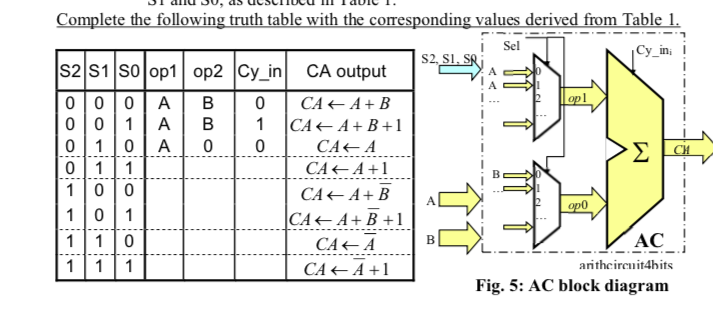
**3. Discussion of the algorithmic solution:**

Explain the used algorithm to solve the problem and the block components you are going to use.

The circuits were mean to be designed separately in a hierarchical manner (Shown below). Files on the lowest level are 1-bit full adder and 1-bit logic and shift circuit (LSC). The intermediate files will consist of a 4- bit register, a 4-bit arithmetic circuit (AC), a 4-bit logic and shift circuit (LSC), and a state circuit. Finally, the file at the highest level will cover the complete circuit. And, Or, Not gates and 8X1 multiplexers were used to implementing the lowest and intermediate files. 4 D flip flops were used to construct 4-bit registers. A quadruple multiplexer was connected with the outputs of arithmetic and logic circuit and the select input,S3 to chose the final outcome of the ALU circuit.

**Design Part**

**1. Presentation of the design:**



**Table 1 truth table COMPLETE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S2 | S1 | S0 | OP1 | OP2 | CY\_IN |
| 0 | 0 | 0 | A | B | 0 |
| 0 | 0 | 1 | A | B | 1 |
| 0 | 1 | 0 | A | 0 | 0 |
| 0 | 1 | 1 | A | 0 | 1 |
| 1 | 0 | 0 | A | B’ | 0 |
| 1 | 0 | 1 | A | B’ | 1 |
| 1 | 1 | 0 | A’ | 0 | 0 |
| 1 | 1 | 1 | A’ | 0 | 1 |

**conclusion: S0 is a carry and simplify the truth table**

|  |  |  |  |
| --- | --- | --- | --- |
| s2 | s1 | op1 | op2 |
| 0 | 0 | A | B |
| 0 | 1 | A | 0 |
| 1 | 0 | A | B’ |
| 1 | 1 | A’ | 0 |

**Design the diagrams:**

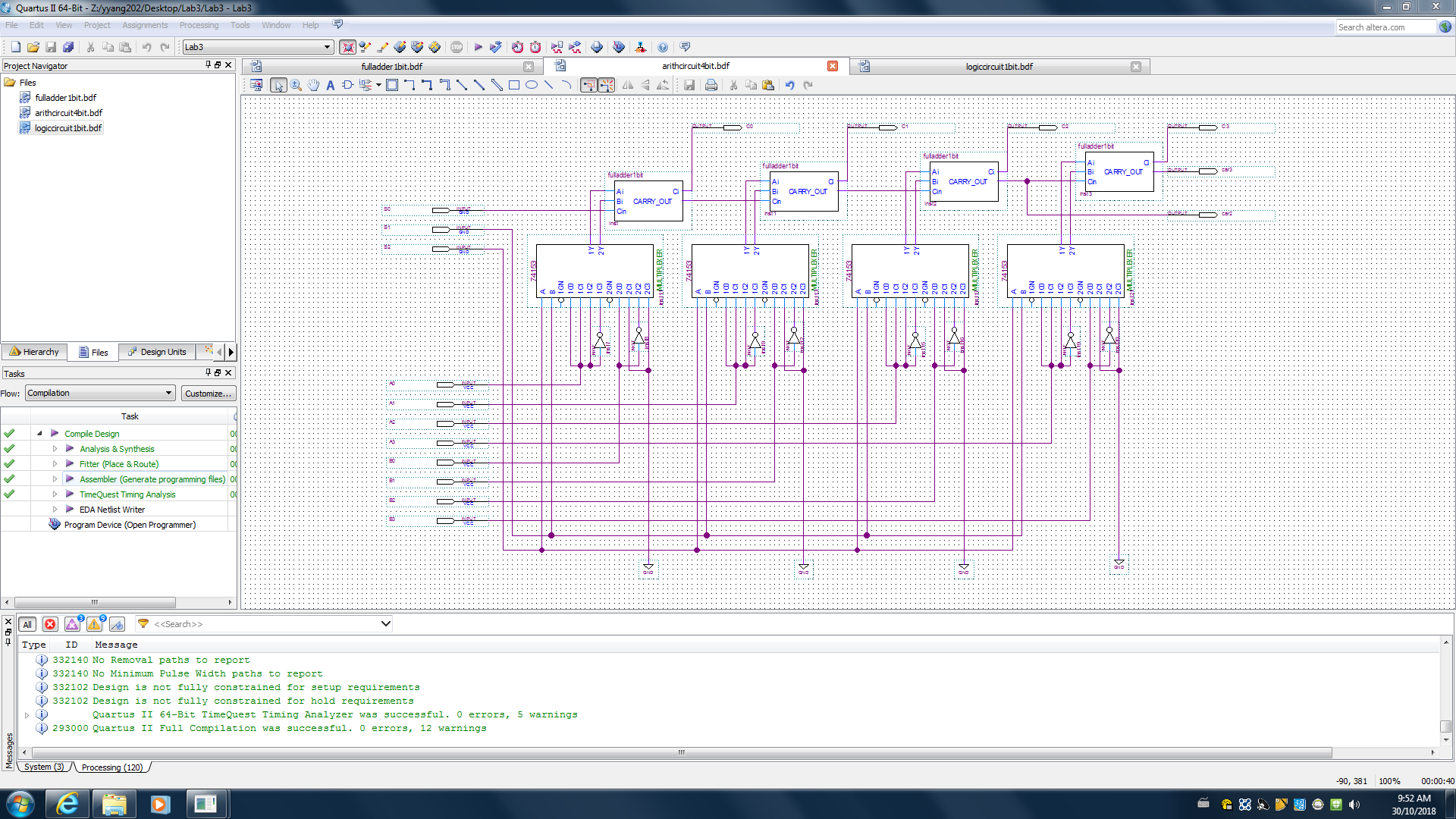


Figure 1: diagram of the arithmetic circuit 4 bit

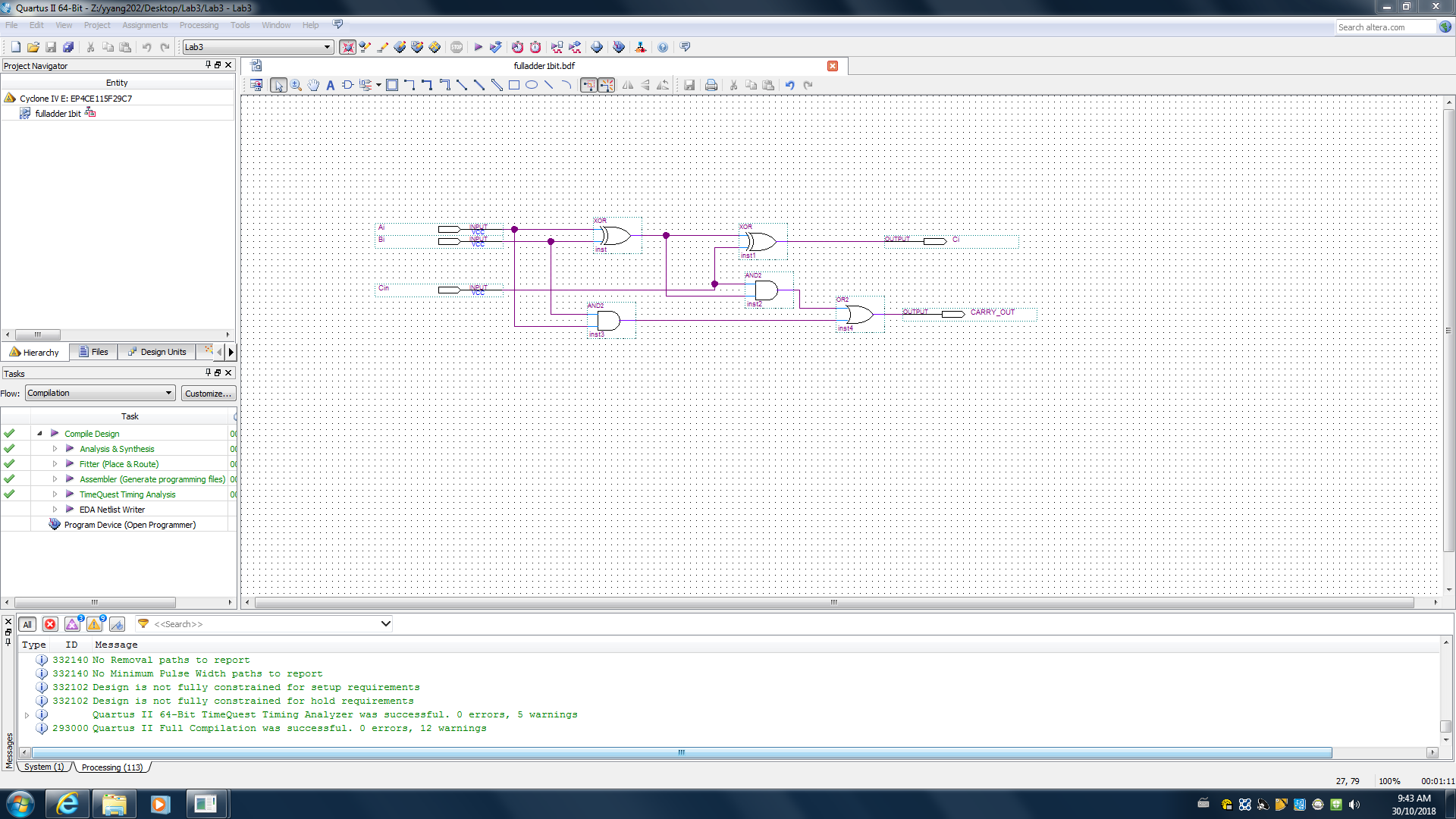


Figure 2: diagram of the full adder 1 bit

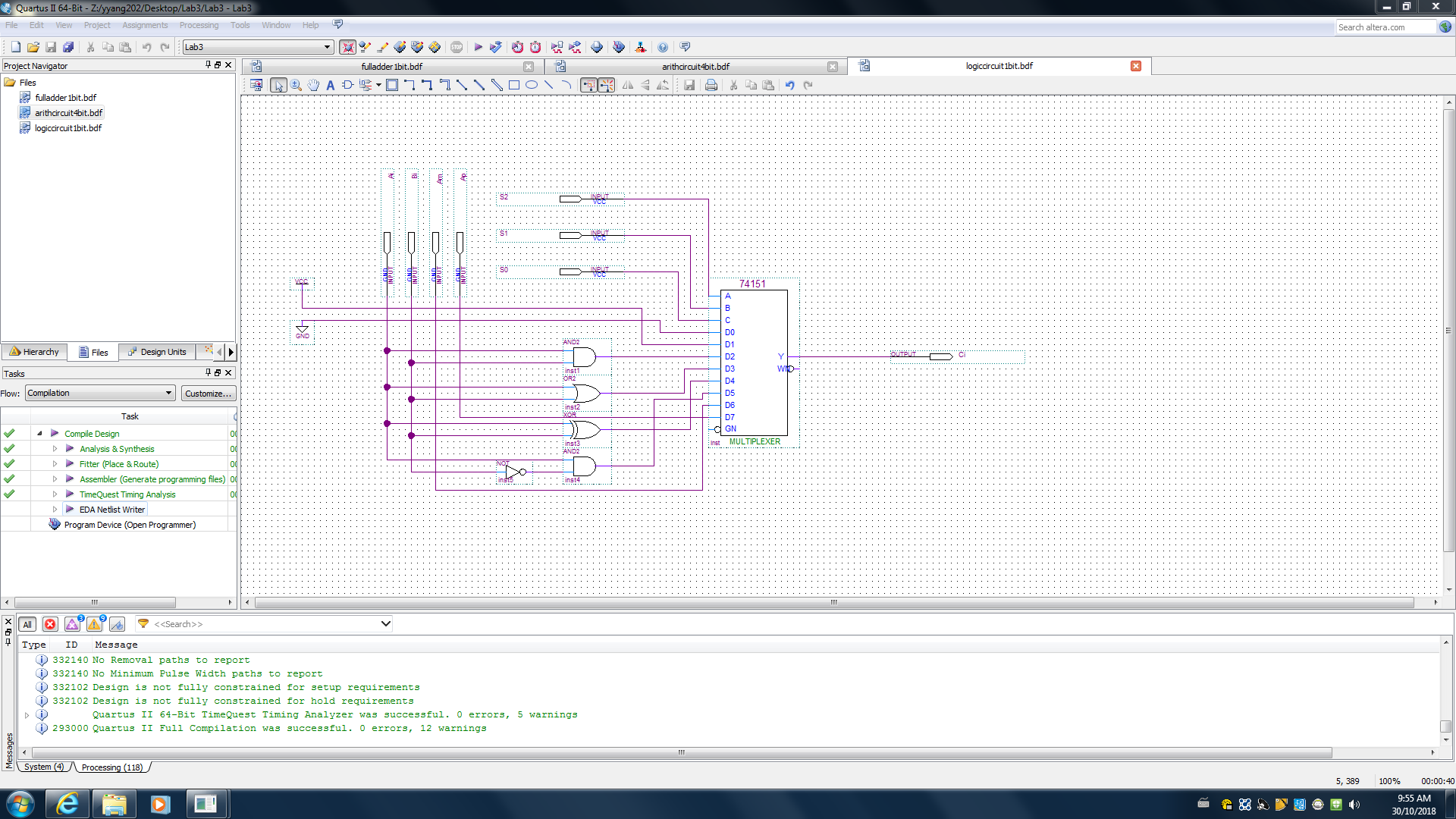


Figure 3: diagram of the logic circuit 1 bit

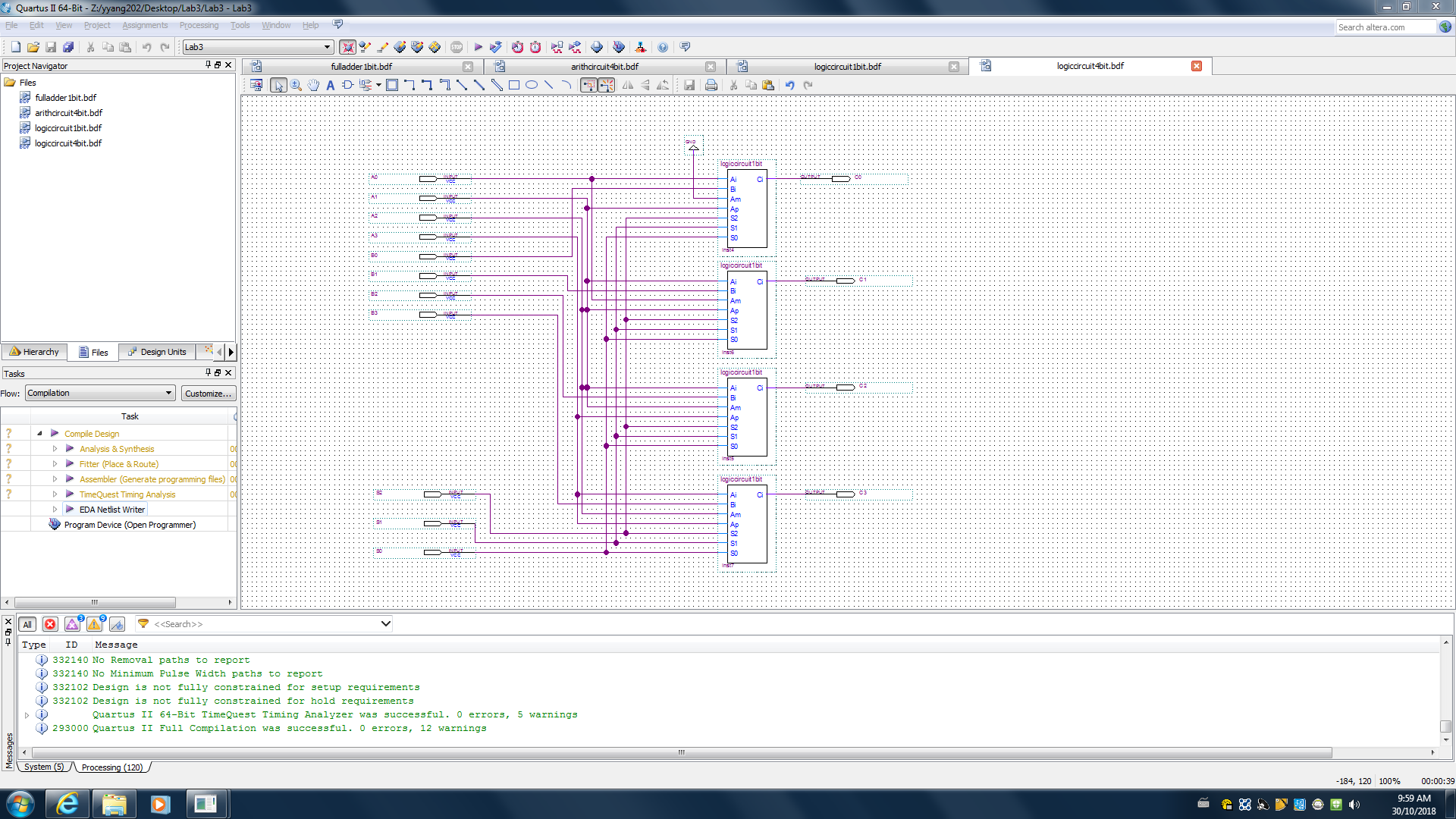


Figure 4: diagram of logic circuit 4 bit

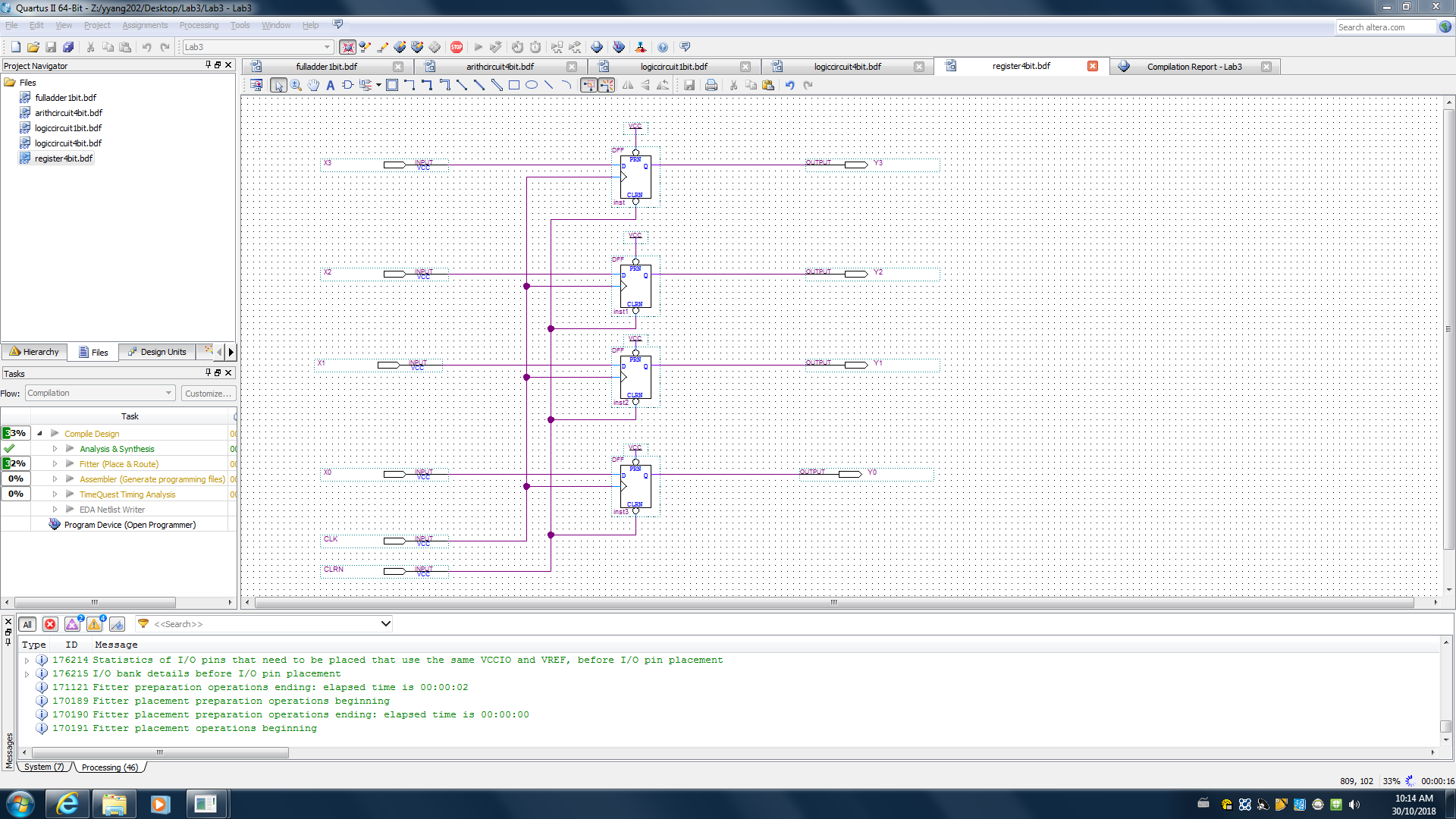
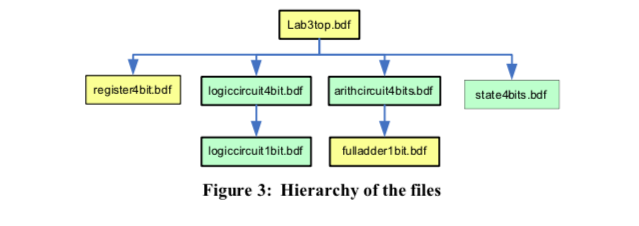


Figure 5: diagram of the register 4 bit



As the manual shows, we designed the lab 3 top as requested.

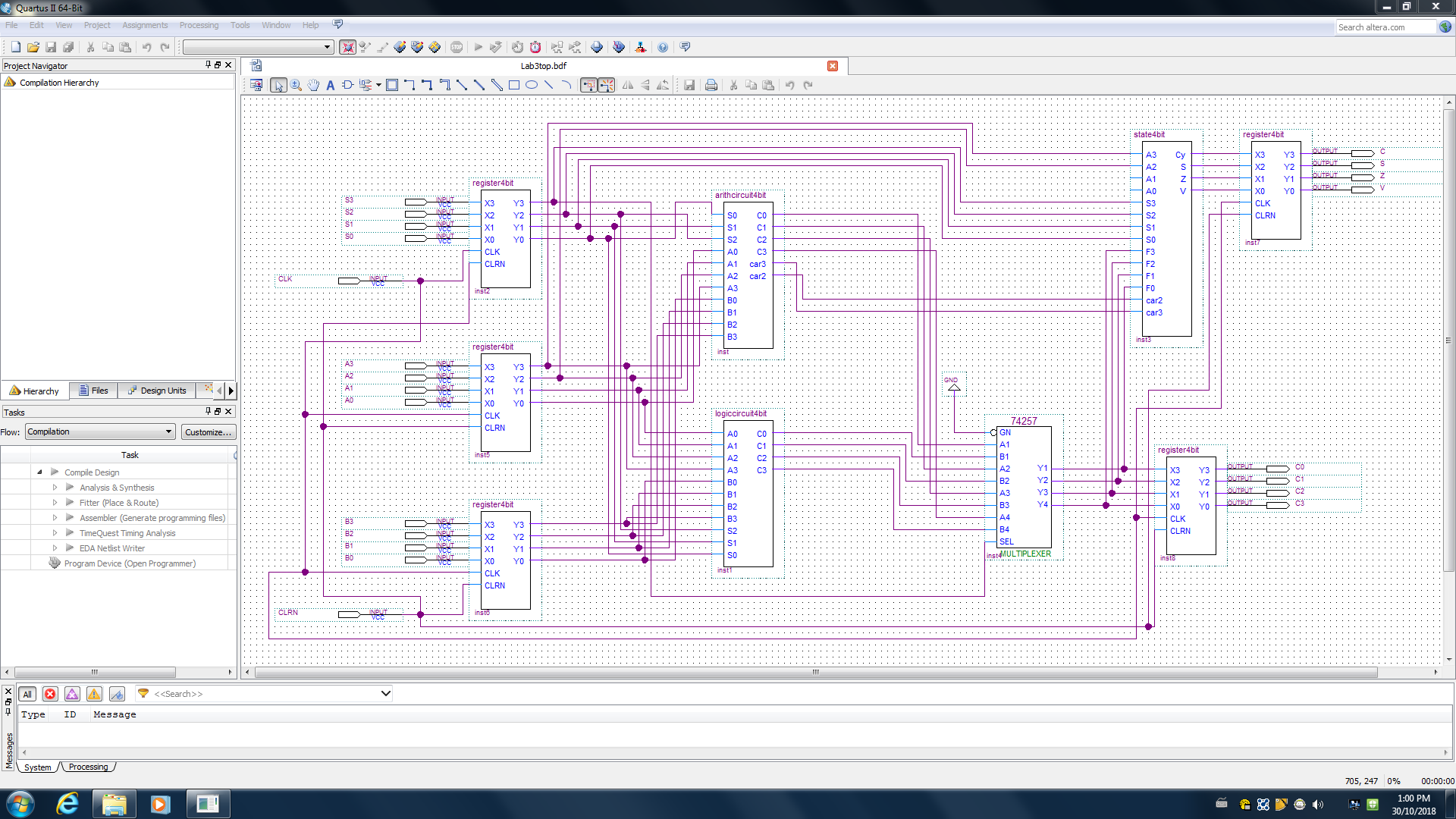


Figure 6: diagram of the lab 3 top

**2. Discussion of used components:**

1. Equipment and Supplies:
2. \* Quartus II (student edition or web edition) \* Altera DE2-115 board with
3. - USB-Blaster cable - Power supply 12 VDC, 2A

**3. Discussion of the actual solution:**

A 1-bit full adder was implemented and compile into a block symbol. It was then used for constructing the 4-bit arithmetic circuit with 4 8X1 multiplexers. Same mean was used for implementing a 4-bit register and logic circuit. After making sure each of the basic circuits was designed correctly and successfully compiled, a symbol of each were created and saved. They were used as components in the final circuit. Each procedure and implementation are shown above.

**4. Discussion of challenging problems (Bonus):**

Some of the logic operations give the incorrect outputs compared to the sequence table. Since the circuit had been identified by the TAs and compiled successfully, we proposed the problem was due to the connection error of the final circuit. By zooming into the block diagram and recheck all wires, it was confusing that no connection error was found. After communicating with TAs and comparing our circuit with others’, we finally realized the problem was due to the state of inputs. I was misguided by looking at the default design of the multiplexers in Quartus II where the inputs were set to be low. Therefore, when creating the block symbols for the arithmetic and logic circuit, It was set to be ground as default and cause the actual inputs the opposite as our expectation. After resetting them to VCC and recreating the block symbols, we replaced the old ones in ALU circuit. The outpt then shows the correct answer.

**Simulation and Verification Part**

**1. Shown simulation/synthesis results:**

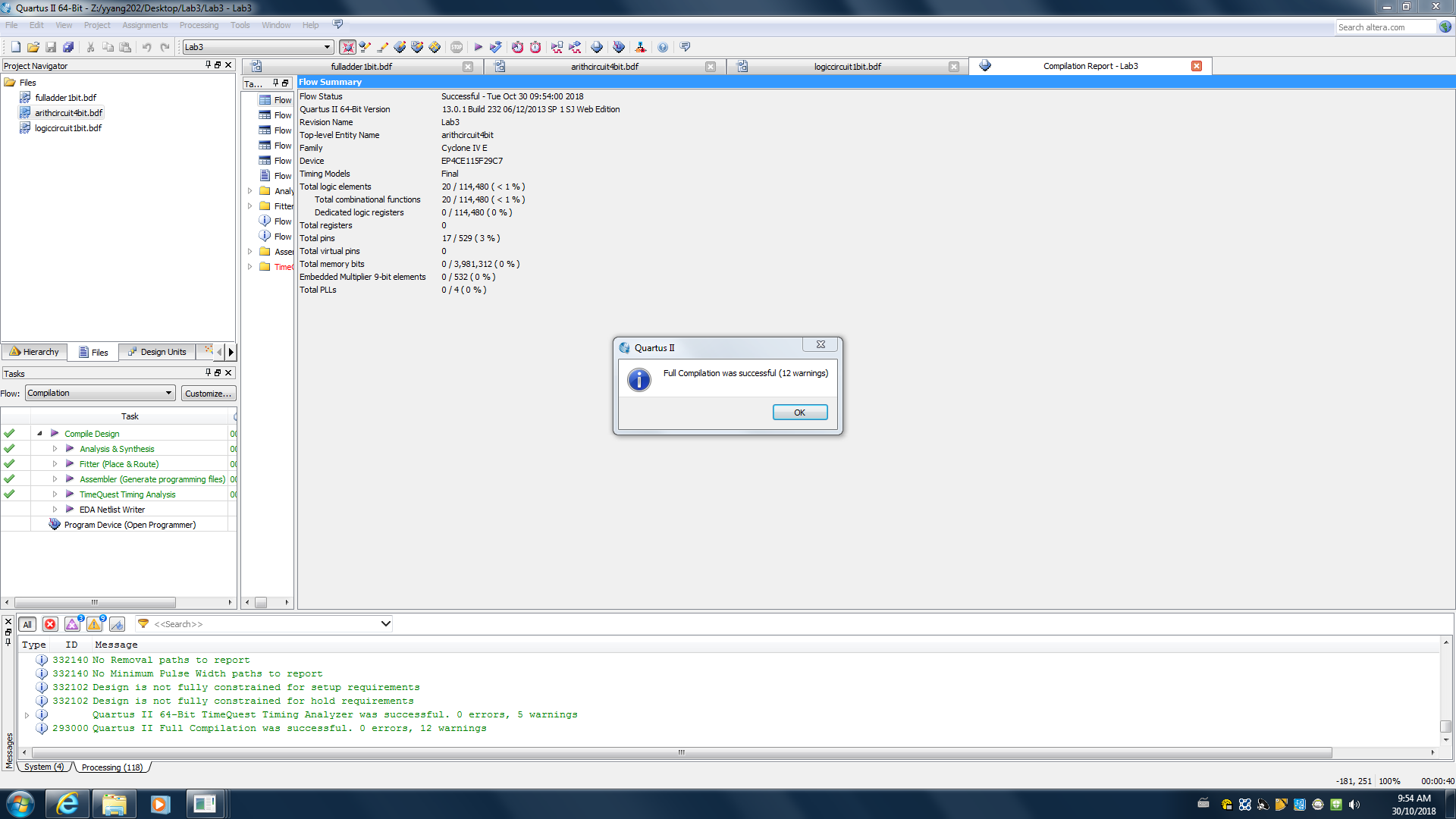


Figure 7: compilation result of the arithmetic circuit 4 bit

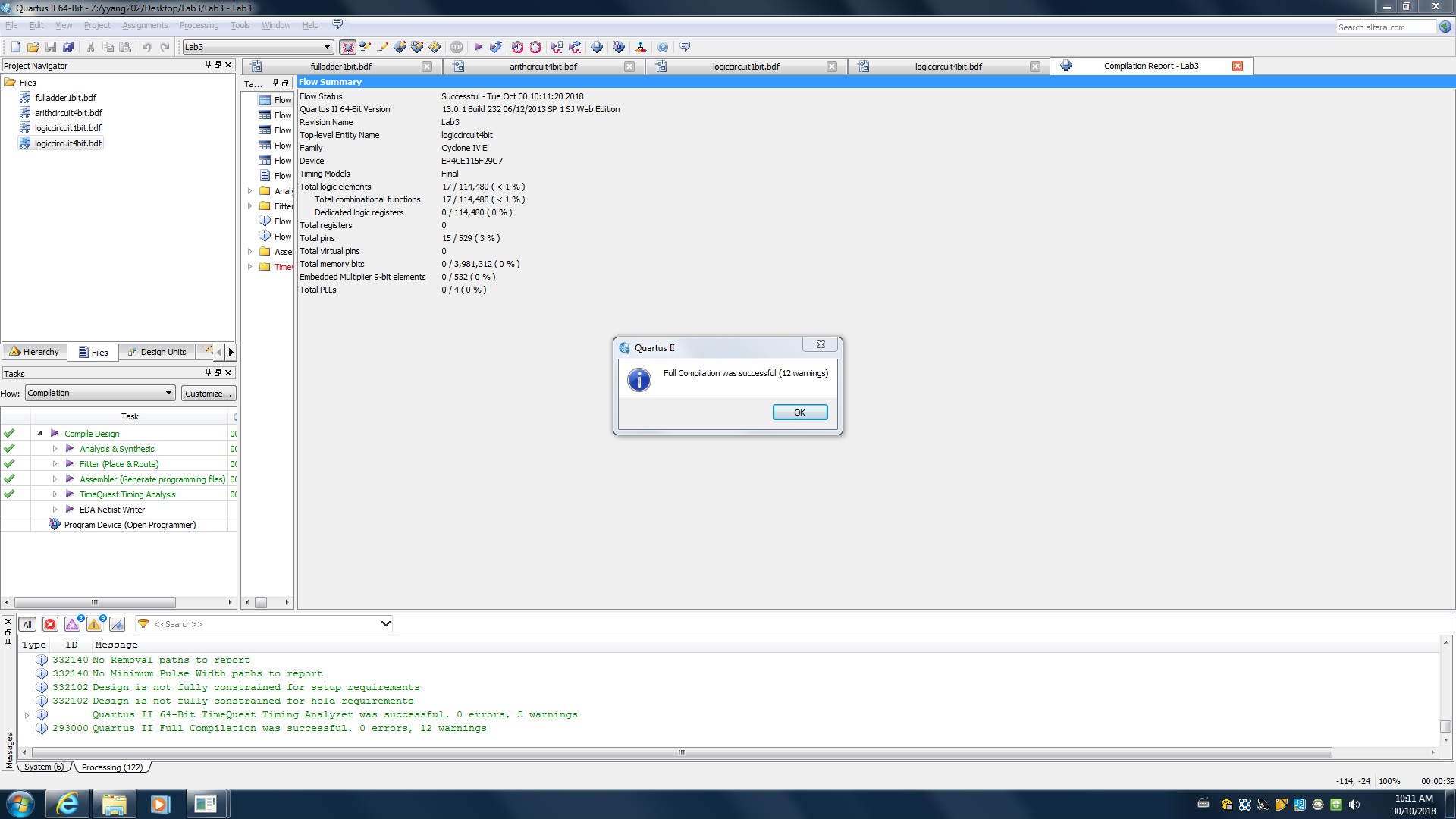


Figure 8: compilation result for the logic circuit 4 bit

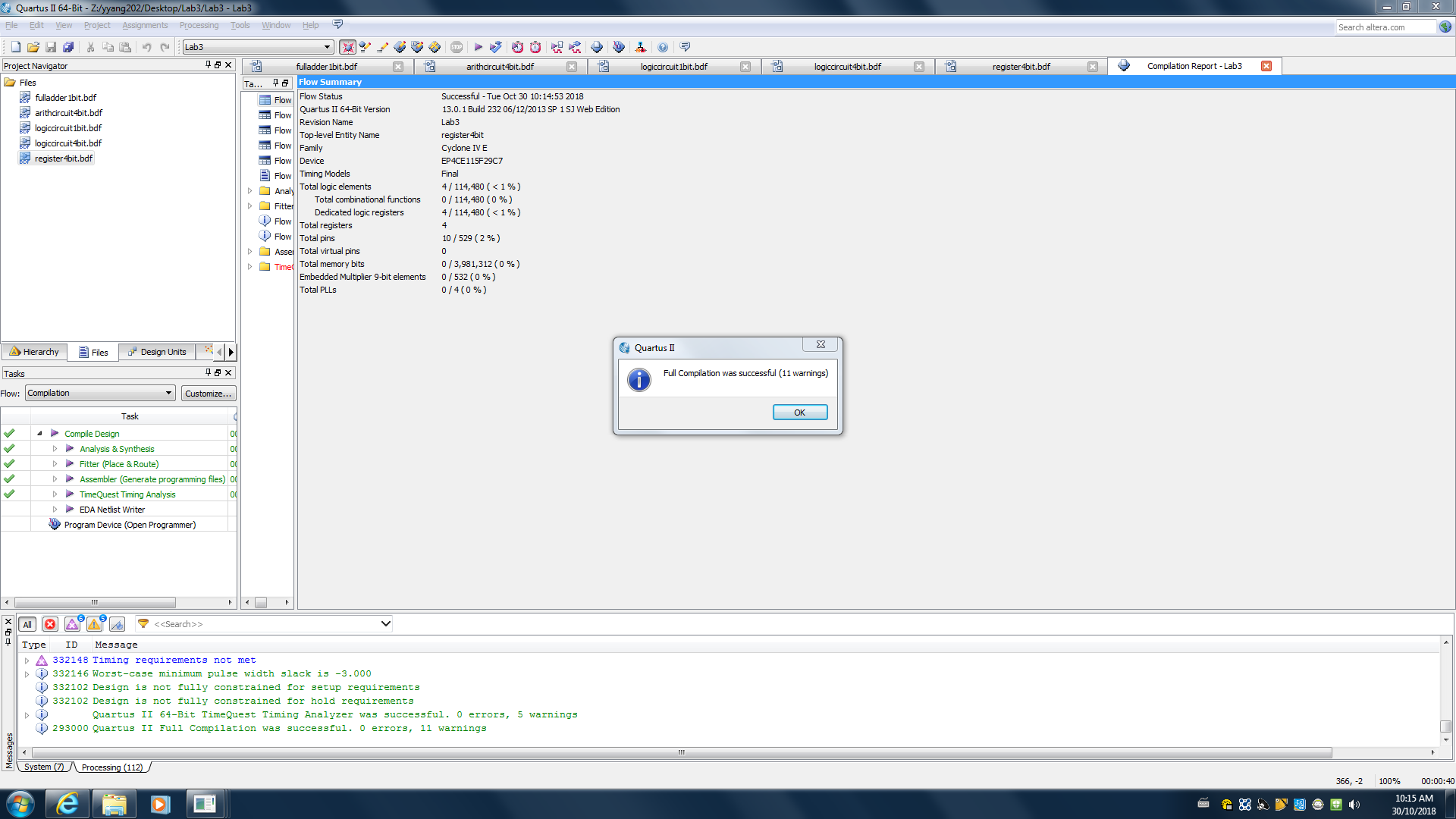


Figure 9: compilation result of the register 4 bit

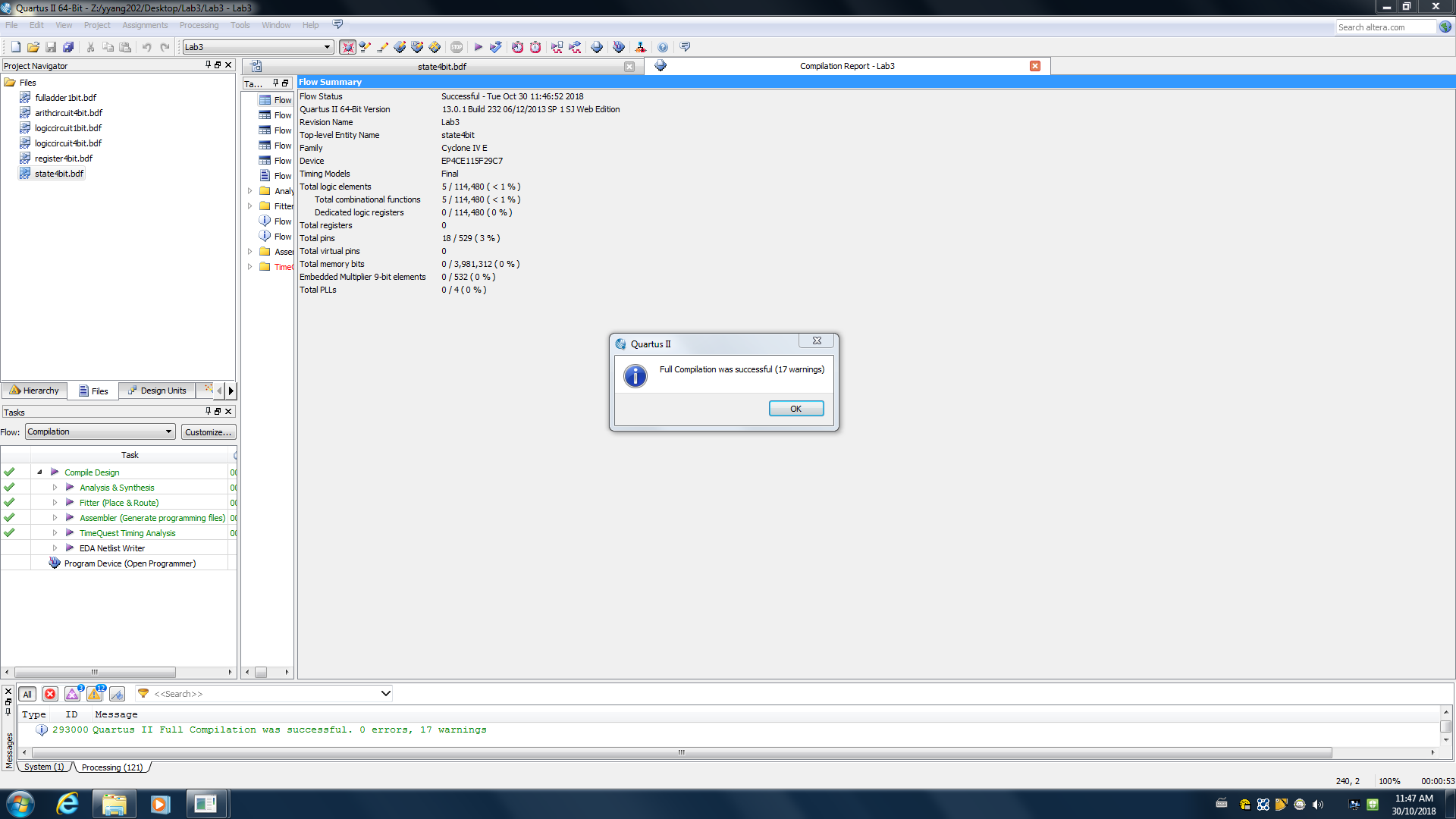
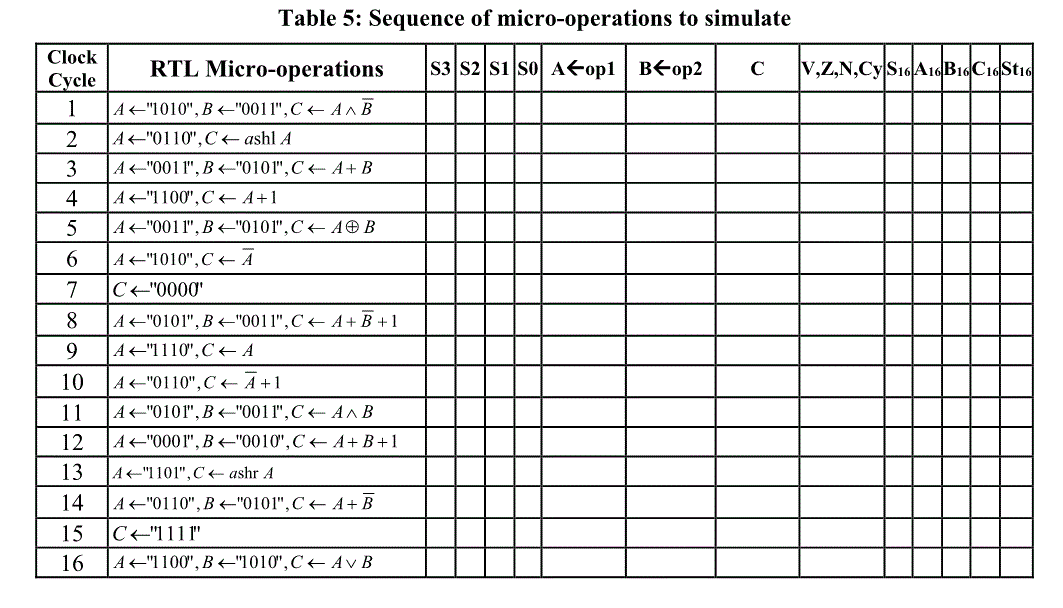


Figure 10 : compilation results for the state 4 bit



**Table 5: Sequence of micro-operations to simulate COMPLETE**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S3S2S1S0 | A<-op1 | B<-op2 | C | v,z,n,cy | S16 | A16 | B16 | C16 | St16 |
| 1101 | 1010 | 0011 | 1000 | 0010 | D | A | 3 | 8 | 2 |
| 1110 | 0110 | XXXX | 1100 | 0010 | E | 6 | 3 | C | 2 |
| 0000 | 0011 | 0101 | 1000 | 1010 | 0 | 3 | 5 | 8 | A |
| 0011 | 1100 | XXXX | 1101 | 0010 | 3 | C | 5 | D | 2 |
| 1000 | XXXX | XXXX | 0000 | 0100 | C | 3 | 5 | 6 | 0 |
| 0110 | 1010 | XXXX | 0101 | 0000 | 6 | A | 5 | 5 | 0 |
| 1000 | XXXX | XXXX | 0000 | 0100 | 8 | A | 5 | 0 | 4 |
| 0101 | 0101 | 0011 | 0010 | 0001 | 5 | 5 | 3 | 2 | 1 |
| 0010 | 1110 | XXXX | 1110 | 0010 | 2 | E | 3 | E | 2 |
| 0111 | 0110 | XXXX | 1010 | 0010 | 7 | 6 | 3 | A | 2 |
| 1010 | 0101 | 0011 | 0001 | 0000 | A | 5 | 3 | 1 | 0 |
| 0001 | 0001 | 0010 | 0100 | 0000 | F | D | 2 | E | 0 |
| 0100 | 0110 | 0101 | 0000 | 0101 | 4 | 6 | 5 | 0 | 5 |
| 1001 | XXXX | XXXX | 1111 | 0010 | 9 | F | 5 | F | 2 |
| 1011 | 1100 | 1010 | 1110 | 0010 | B | C | A | E | 2 |

**2. Experimental verification of operation of your circuits:**

Our verification results were the same as the sequence table. The outcomes of States which were indicated by 4 LED lights while giving the inputs of A,B and S, were matched. Therefore, the lab was considered successful.

**Discussion and Conclusions:**

The circuit was not able to operate some of the logic operations at the beginning. Connection error was considered the problem which caused the failure. It had been rechecked through every part of the circuit from the lowest entity to the top ones and no connection error was found. After communicating with teaching assistants, we suddenly realized it was a big mistake by setting the input of each connection of parts to be ground. It would make the actual input be opposite of our expectations and give the incorrect output. It was fixed and the lab was successfully operated.