3TB4 – Lab 1 Prelab

Hank Bae

400062215

Jan 24, 2019

# Tutorial Report

In this tutorial, we familiarised ourselves with Intel Quartus programming the De1-SoC so that it was effectively an XOR gate where it would turn an on board LED on and off depending on the state of two on board switches.

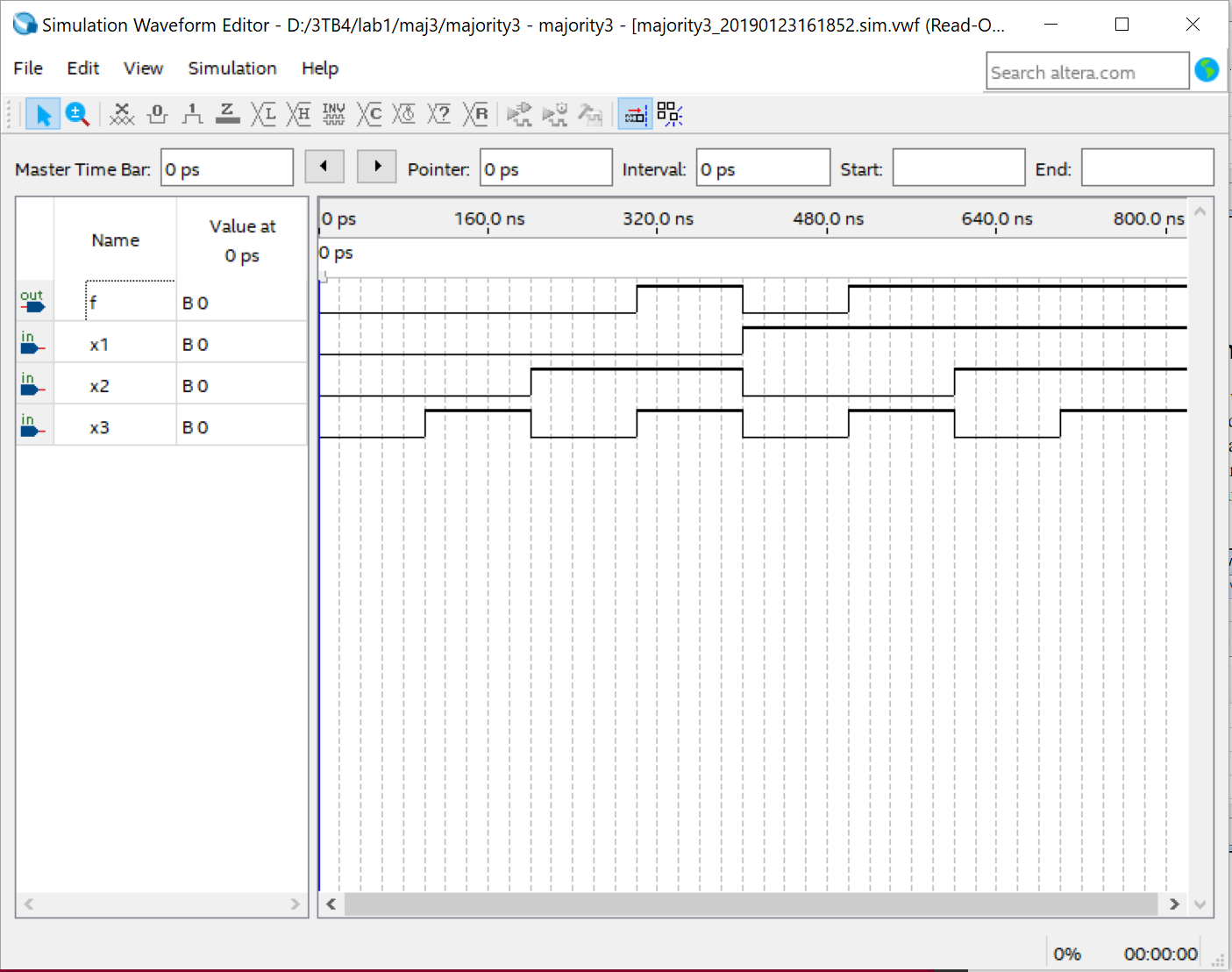


Figure 1: Waveform for Majority3

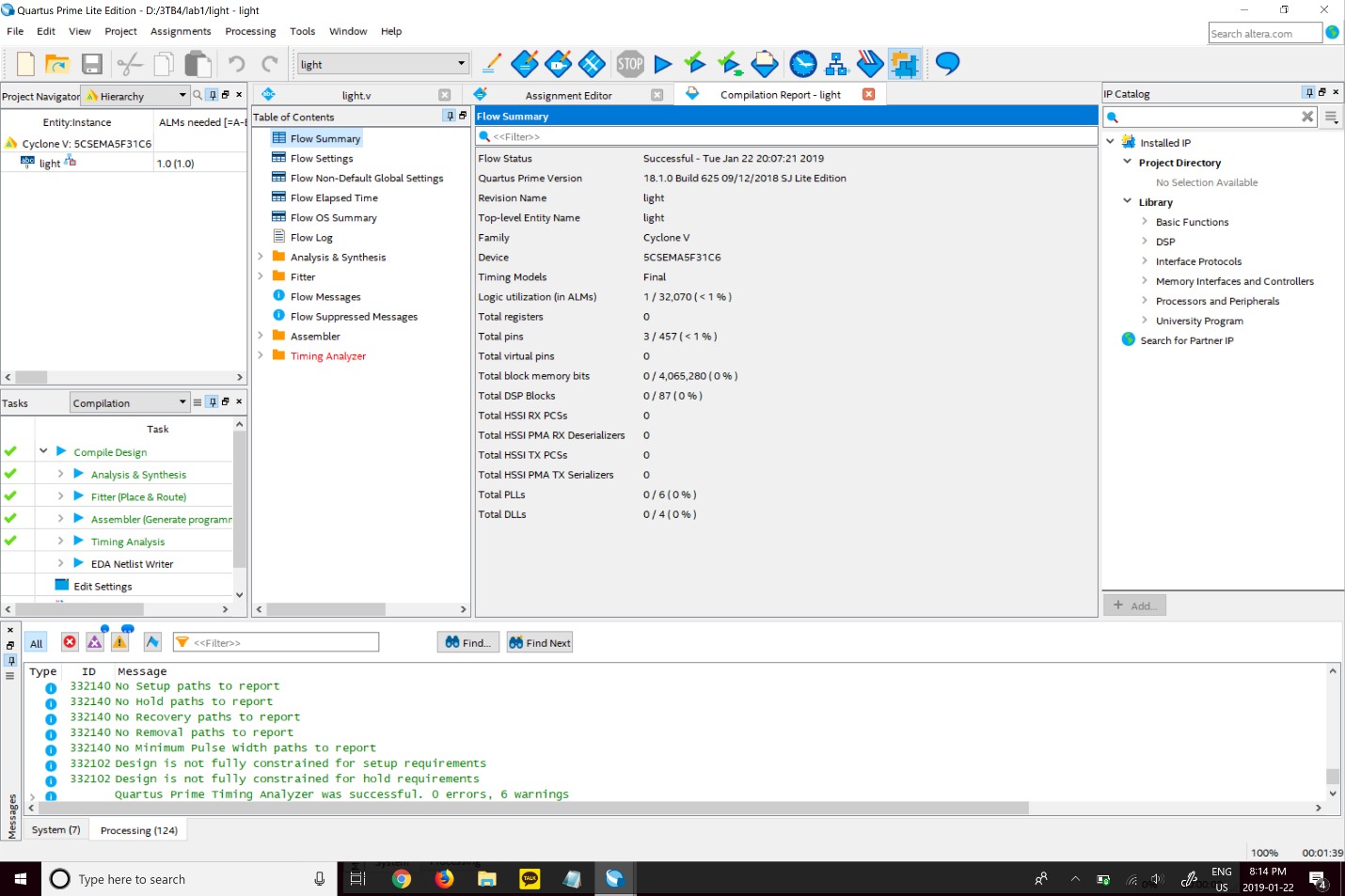


Figure 2: Screenshot of compilation report

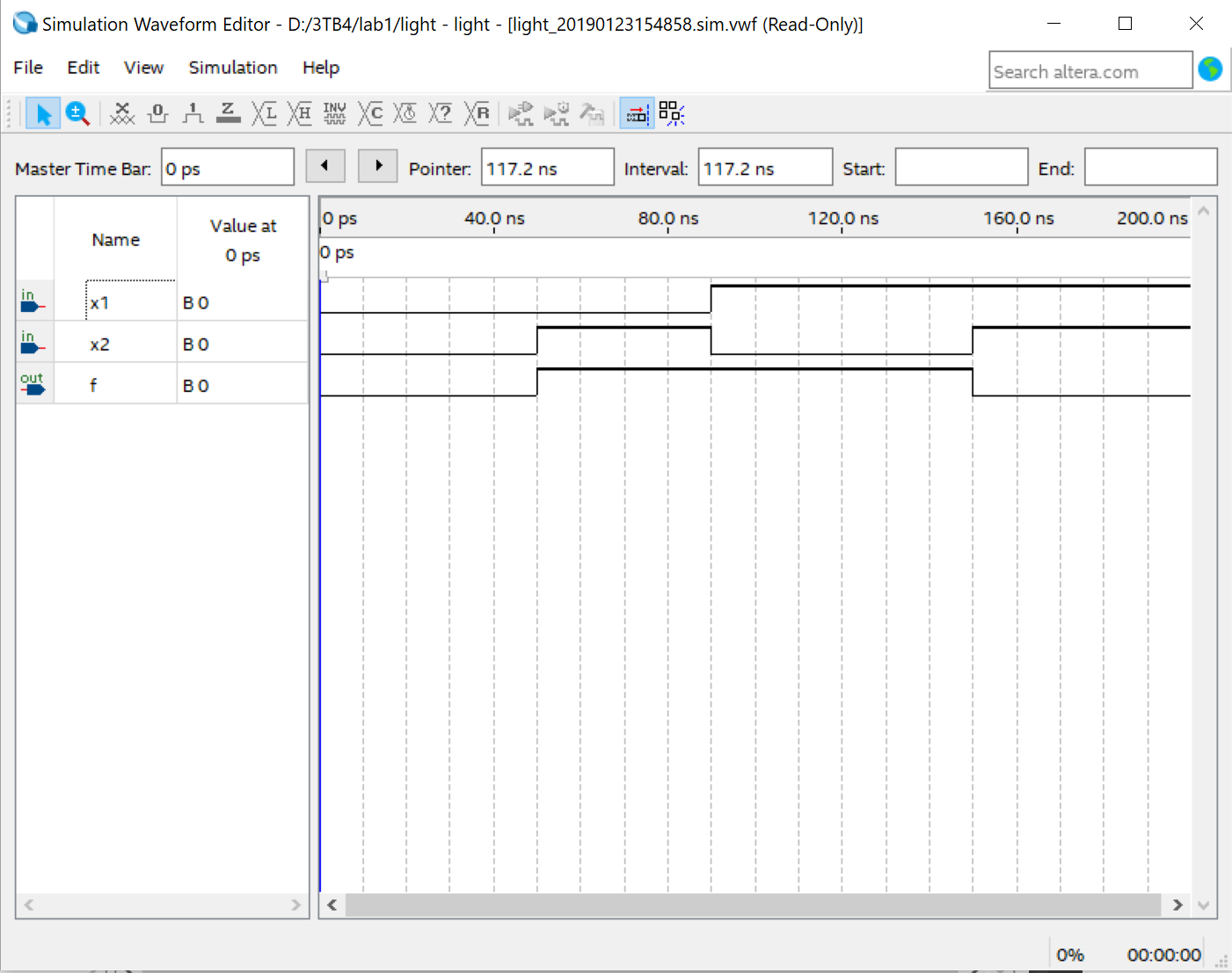


Figure 3: Waveform for Light File

# Short Answer Questions

1. **reg** can store a value whereas **wire** will only create a node in a circuit.
2. Yes, the **wire** data type can be used on the left side of an expression.
3. The parameters of a module must be specified as either **input** or **output** and their data type must be specified as well unless they will be used are **wire** data types.
4. Continuous assignment are definitions that do not change whereas blocking and nonblocking assignments are executed when only ran. The difference between blocking and nonblocking is that blocking is sequential whereas nonblocking is not meaning if a statement depends on the previous statement, the result will differ depending on a blocking or nonblocking assignment is made.
5. In combinatorial logic, **assign** and **always** may be used but in sequential logic, only **always** may be used.
6. To prevent inferred latches, the code must be complete and disjoint meaning it just cover all cases.
7. **<<** is a binary logical left shift operator whereas **<<<** is a binary arithmetic left shift operator.
8. To declare an array of 6 elements of a 7-bit wire: **wire [6:0] x[5:0];**

# Verilog Modules

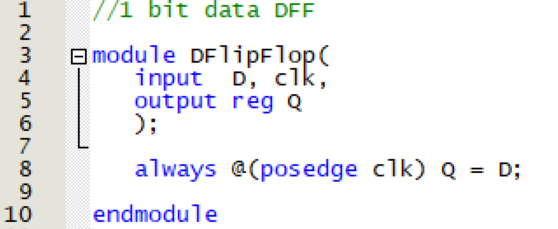


Figure 4: One-Bit data width D flip-flop

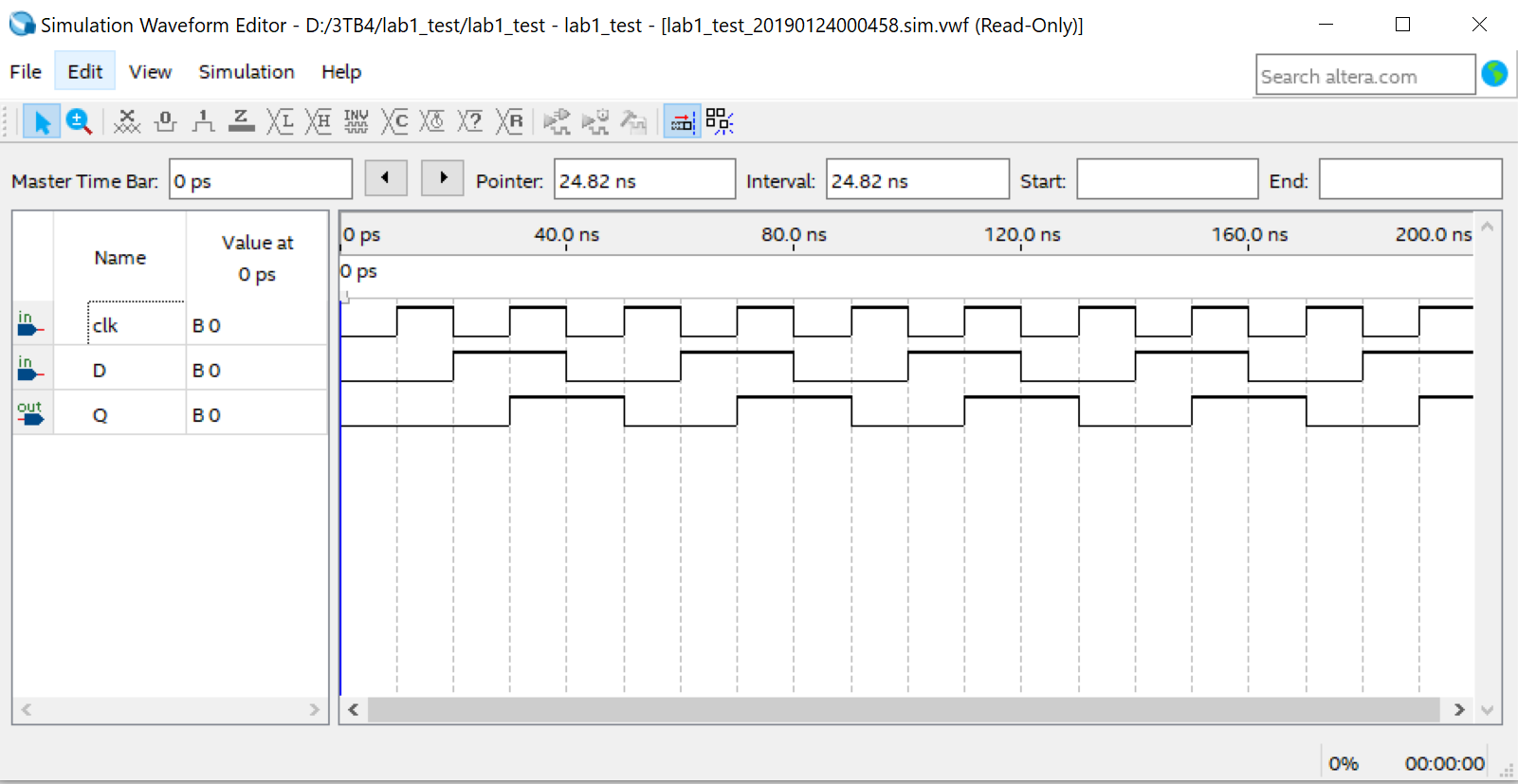


Figure 5: Waveform for One-Bit data width D flip-flop

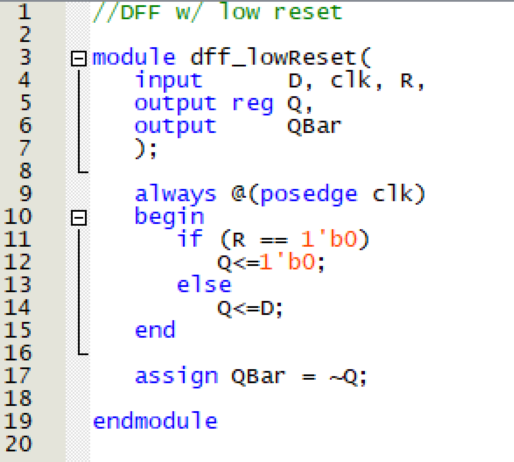


Figure 6: One-Bit data width D flip-flop with active low synchronous reset

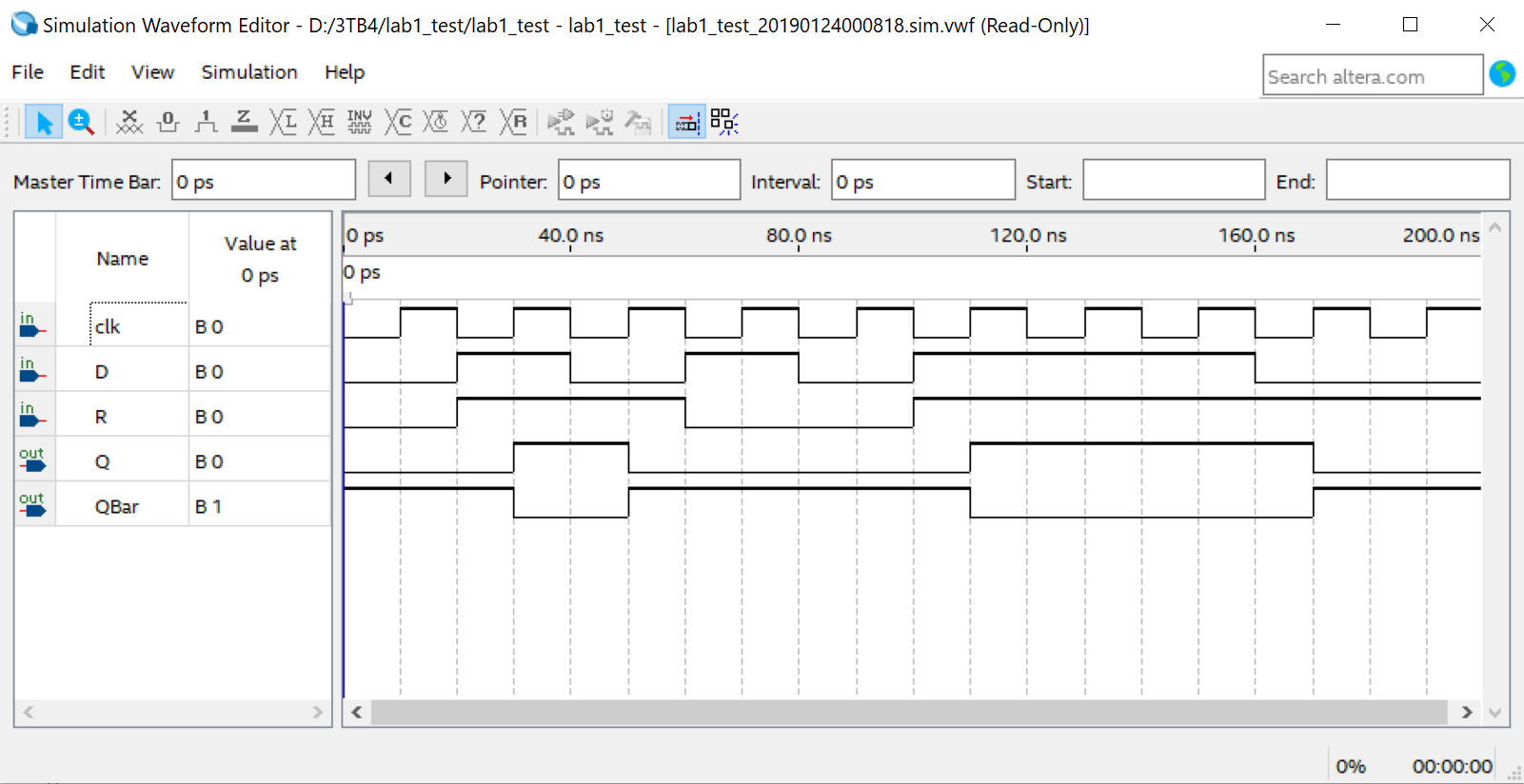


Figure 7: Waveform for One-Bit data width D flip-flop with active low synchronous reset

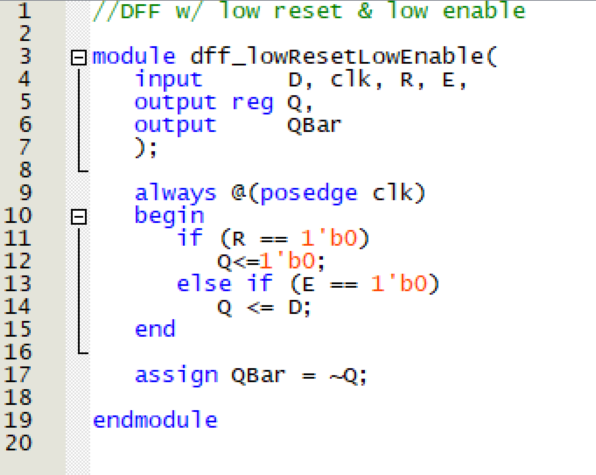


Figure 8: One-Bit data width D flip-flop with active low synchronous reset and active low enable

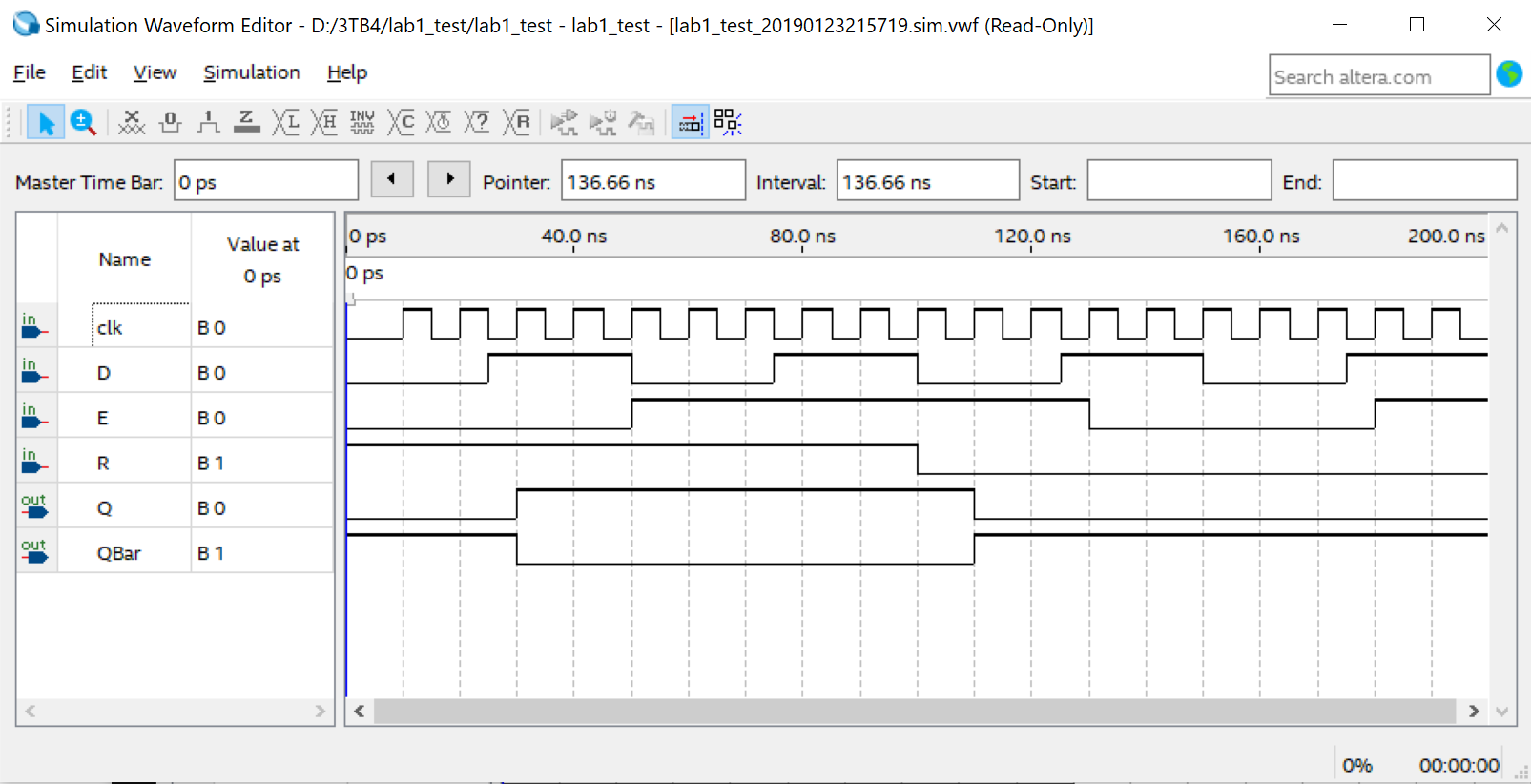


Figure 9: Waveform of One-Bit data width D flip-flop with active low synchronous reset and active low enable

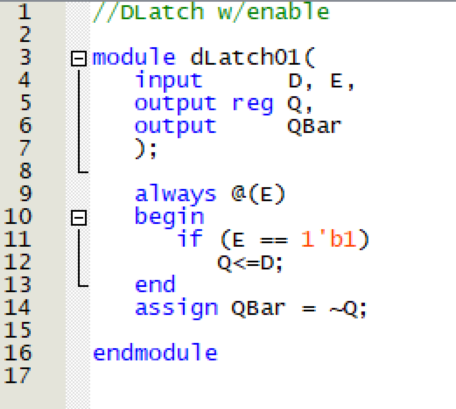


Figure 10: D-Latch with synchronous enable control

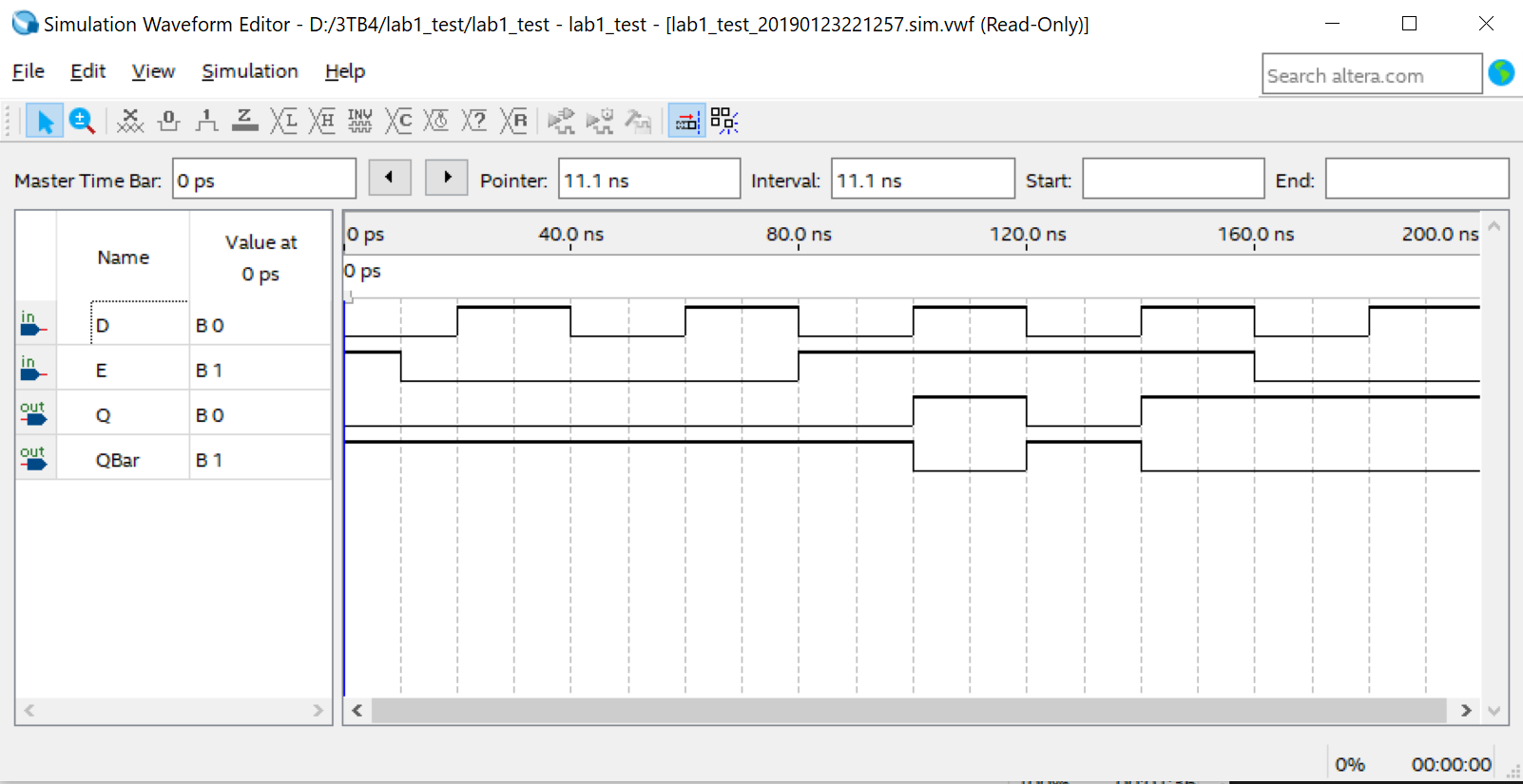


Figure 11: Waveform of D-Latch with synchronous enable control

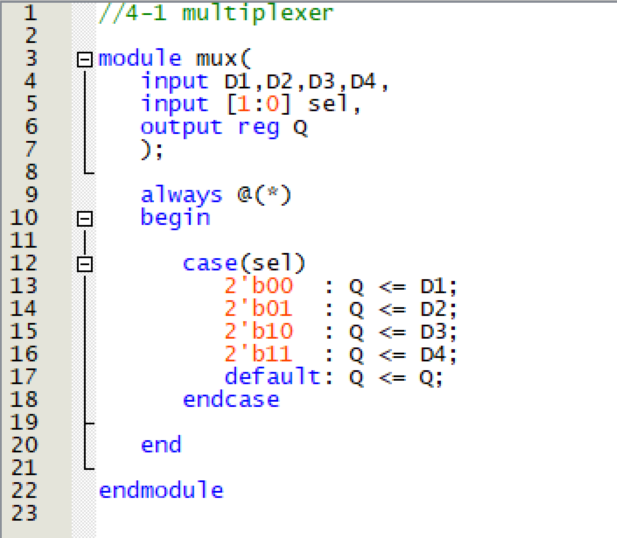


Figure 12: 4-to-1 multiplexer

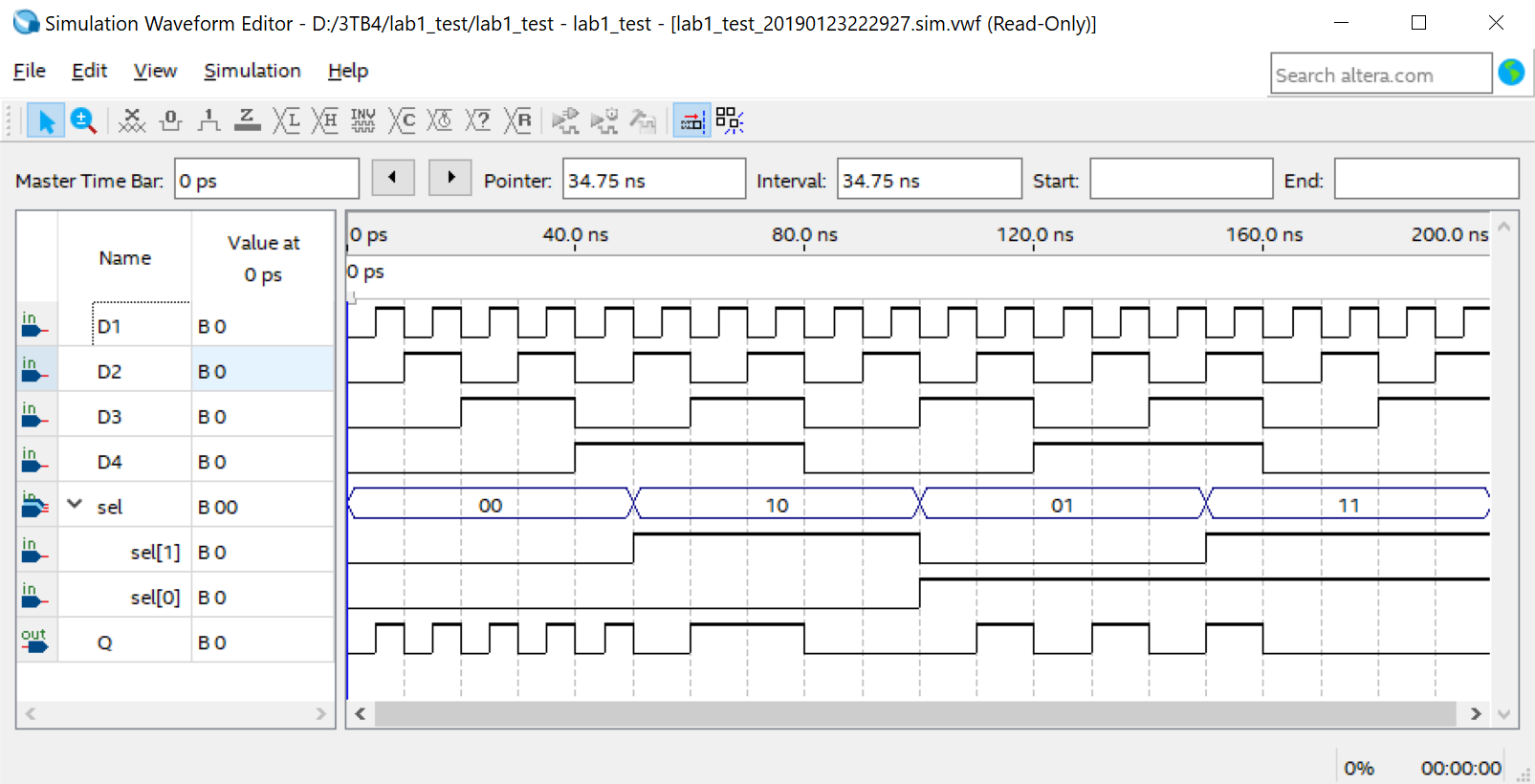


Figure 13: Waveform of 4-to-1 multiplexer

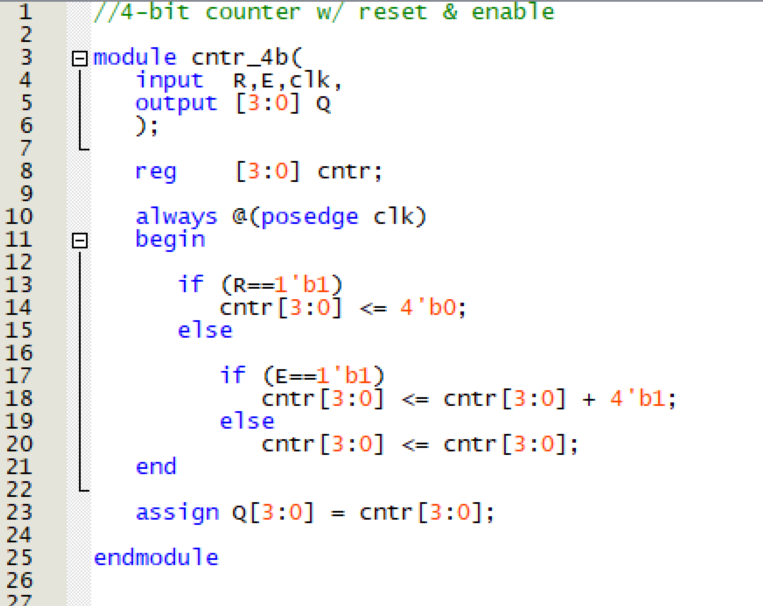


Figure 14: 4-bit counter with reset and enable controls

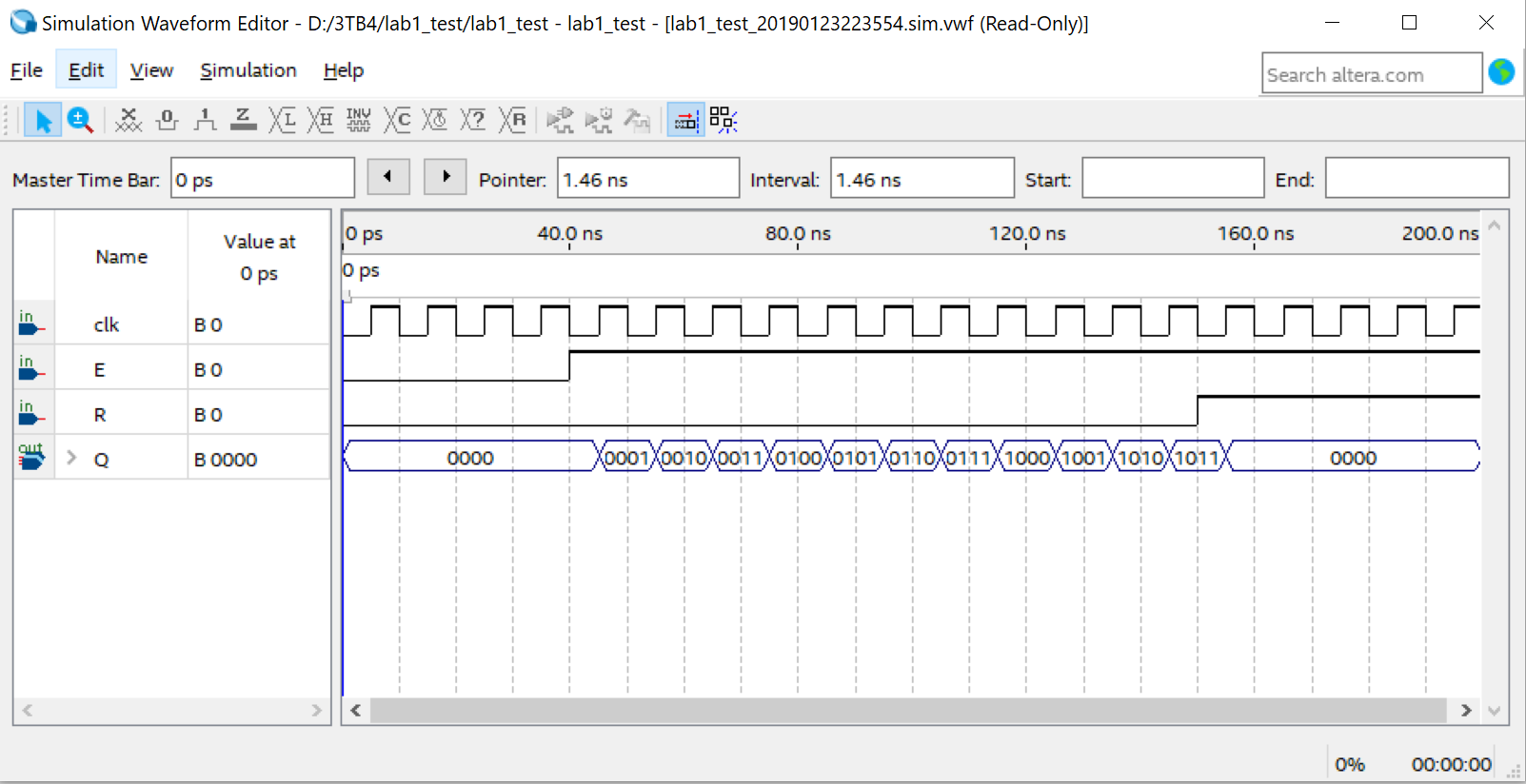


Figure 15: Waveform of 4-bit counter with reset and enable controls

# Truth Table for Seven-Segment Decoder

Table 1: Truth table for 7 segment display

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Char | Input | | | | Segment | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |
| **1** | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | **1** | **1** |
| **2** | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **0** |
| **3** | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **0** |
| **4** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **0** |
| **5** | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **0** | **1** | **0** | **0** |
| **6** | **0** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| **7** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **1** | **1** | **1** | **1** |
| **8** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **9** | **1** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **H** | **1** | **0** | **1** | **0** | **1** | **1** | **0** | **1** | **0** | **0** | **0** |
| **A** | **1** | **0** | **1** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **N** | **1** | **1** | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **1** | **0** |
| **K** | **1** | **1** | **0** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **0** |
| **B** | **1** | **1** | **1** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| **NULL** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |

# Logic Expressions

# Verilog Circuit

# 

# 