ADS ASSIGNMENT 1

Lucien Tran

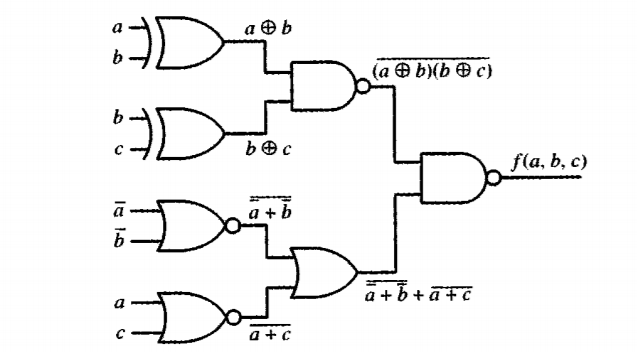
2019

**Purpose**

Using VHDL and Quartus, implement a combinational and sequential circuit by making use of assumptions and discussing the test results simulated during the experiment.

**Part 1: Combinational Circuit Modelling**

Using VHDL, we were asked to implement the combinational circuit below.



The first step to design this circuit is to find the equation for the output f and simplify the expression. After that, the truth table of the circuit was made.

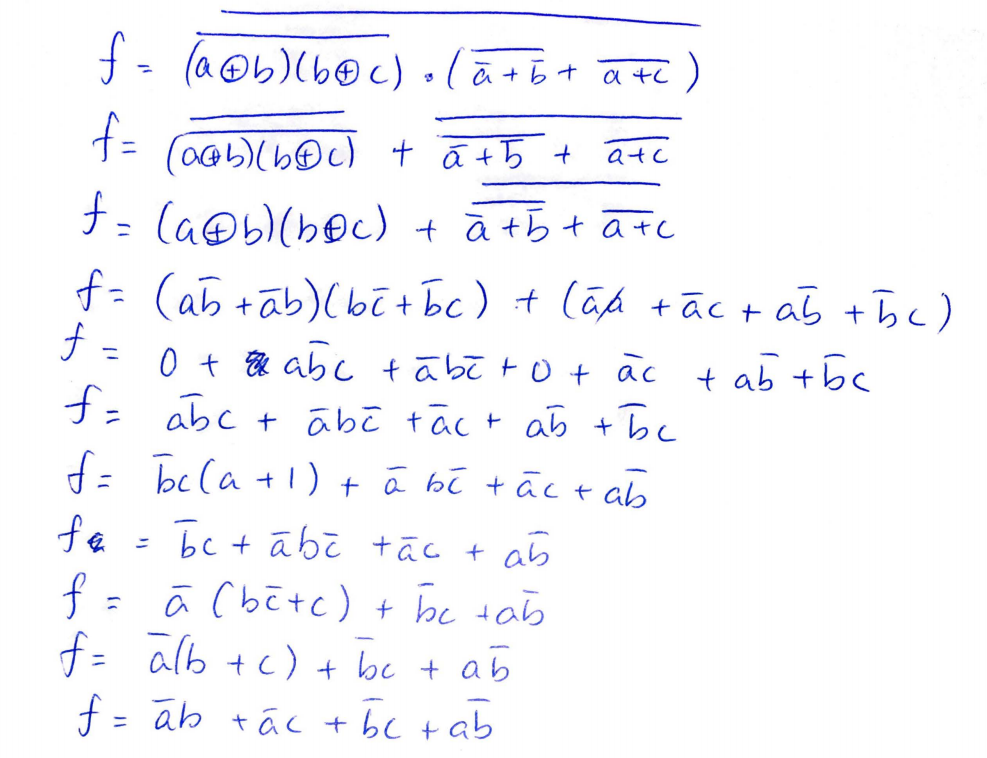


Fig. Boolean Equation simplification

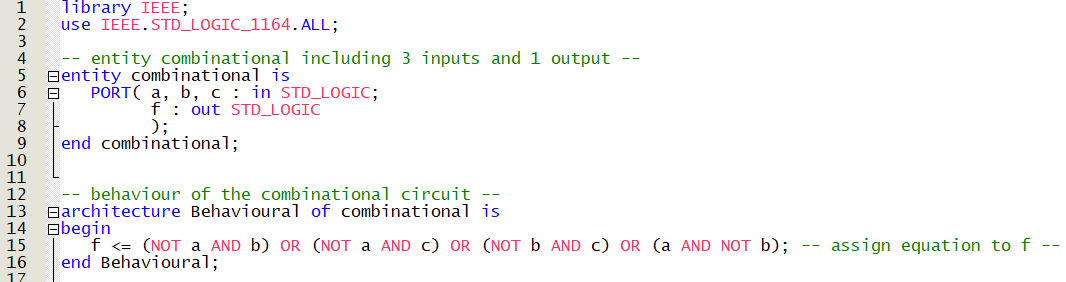
**Assumption**

Testing when the input are a=0, b= 1 and c =1 will give us an output of 1 as shown on the truth table below.

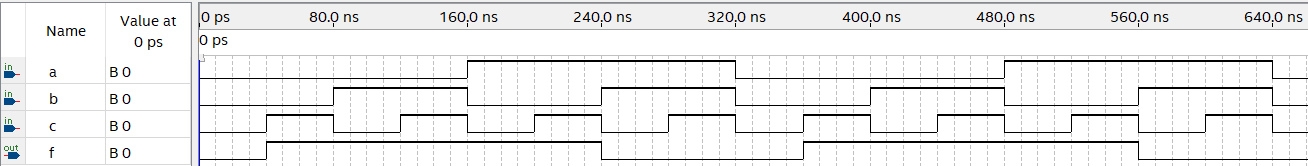
|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **f(A,B,C)** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Table. Truth Table found from the simplified boolean equation

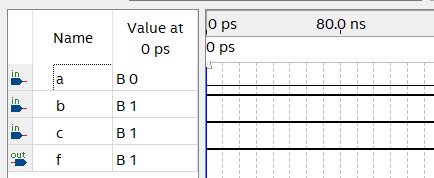
After simplifying the boolean expression of f, create a VHDL file on a new project and create the entity. The entity will have 3 inputs (a, b and c) and 1 output (f). Write down the expression found earlier in behavioural of the circuit and compile it.



The results of the simulation are found below:



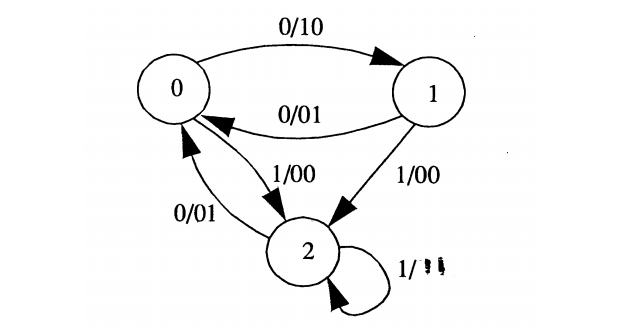
Testing with the input a=0, b=1, c=1, we get the following results:



**Discussion**

With the simulation, we compare it with the truth table found earlier using the boolean equation found above. We can notice that the simulation is matching the truth table and the boolean equation. Thus, it satisfies our assumptions.

**Part 2: Sequential Circuit Modelling**

****

The second part of the report explains the implementation of a sequential circuit and testing this input sequence: 100101. Another requirement for the sequential circuit is to make a synchronous reset and state transitions on the falling edge of the clock. The reset feature outputs “00” and returns to state 0. Making the reset synchronous means that the reset is synchronized with the clock and in this case the falling edge.

**Assumption**

In order to implement the circuit, the truth table and a table showing the next state and output of the sequence is needed and found using the state diagram above. To avoid confusion, state 0,1 and 2 will be respectively replaced by state A, B, C.

|  |  |  |  |
| --- | --- | --- | --- |
| Current State | Input | Next State | 2-bit output |
| 00 (A) | 0 | 01 (B) | 10 |
| 00 (A) | 1 | 10 (C) | 00 |
| 01 (B) | 0 | 00 (A) | 01 |
| 01 (B) | 1 | 10 (C) | 00 |
| 10 (C) | 0 | 00 (A) | 01 |
| 10 (C) | 1 | 10 (C) | 11 |

Table. Truth table of the sequential circuit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| State | A | C | A | B | C | A |
| Input Sequence | 1 | 0 | 0 | 1 | 0 | 1 |
| Next State | C | A | B | C | A | C |
| Output | 00 | 01 | 10 | 00 | 01 | 00 |

Table. States and Output for Input Sequence

Testing the sequence above, we assume that it will satisfy the above table.

To get started with coding in VHDL, we need first to create a new entity for the sequential circuit. It has 3 inputs, that includes the clock, the reset and the input that determines the next state and the 2-bit output.

Creating an architecture for a sequential circuit mildly differs from the combinational circuit code. A state signal has to be created listing the different states possible and the PROCESS keyword has to be used. Depending on the input and current state, the next state and output will change accordingly. When the reset input is high, it will wait for the falling edge of the clock to reset the circuit, which makes the reset synchronous.

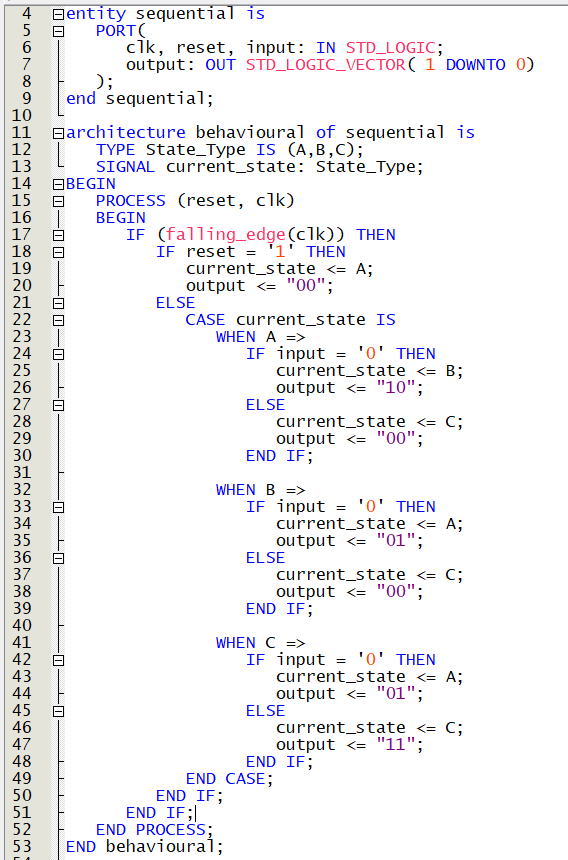
****

Fig. VHDL code for the sequential circuit

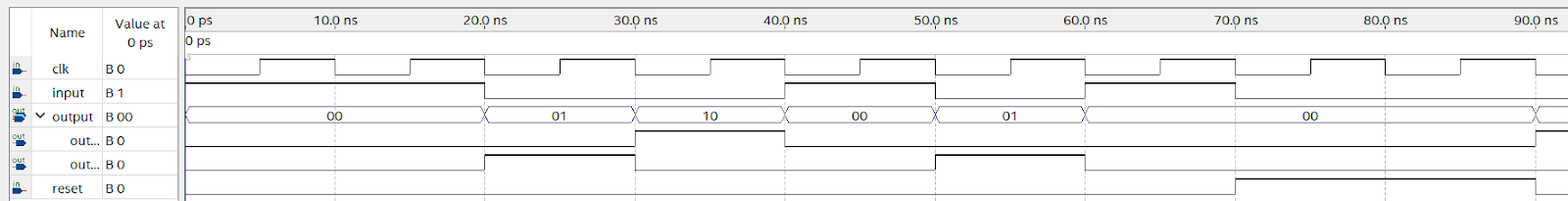
****

Fig. Simulation of the input sequence for the sequential circuit

**Discussion**

As shown above, an extra clk period is needed in the beginning of the simulation for the sequential circuit to satisfy the table above. The cause of this event is the use of the operand <=. After the falling edge of the first clock period, the circuit is now operational.

In conclusion and as seen above and the comparison between the simulation of the sequence and the table, we can notice that both results are identical and therefore satisfies our assumption.