**Lab 2: Boolean Logic**

**ITI 1100 C – Digital Systems 1**

**Winter 2016**

**School of Electrical Engineering and Computer Science**

**University of Ottawa**

**Course Coordinator: Dr. Ahmed**

**Teaching Assistants:**

**Heli Amarasinghe**

Group #: 26

Student Name and Number: Abigail Woods, 7619696

Student Name and Number: Felix Singerman, 7970742

Experiment Date: 2016-02-23

Submission Date: 2016-03-01

**Lab 1: Logic Gates**

**Objectives**

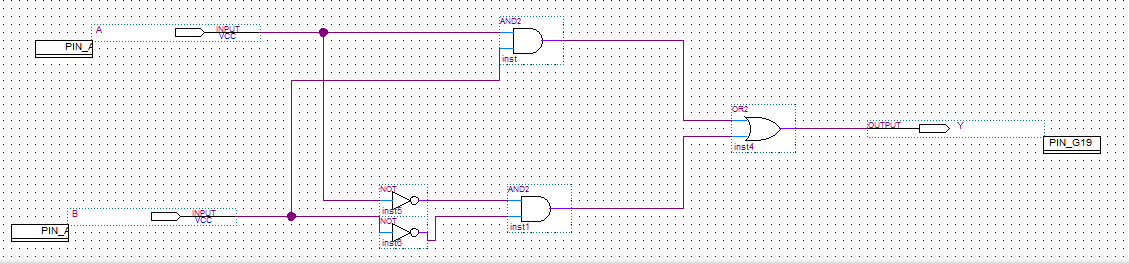
* Simplify logic functions starting from their truth tables or Boolean expressions
* Synthesize, implement and test minimized combinational circuits
* Devise and design combinational logic circuits from specifications
* Implement combinational circuits using any type of available logic gates
* Implement combinational circuits using NAND gates only

**Equipment and Components**

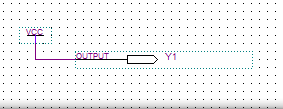
* Quartus II 13.0 Service-Pack 1 Software (64-bit)
* Altera DE2-115 circuit board
* Altera DE2-115 chip (EP4CE115F29C7N)

**Circuit Diagrams**

**Part 1 – Combinational Logic Circuits Minimization by Boolean Algebra**

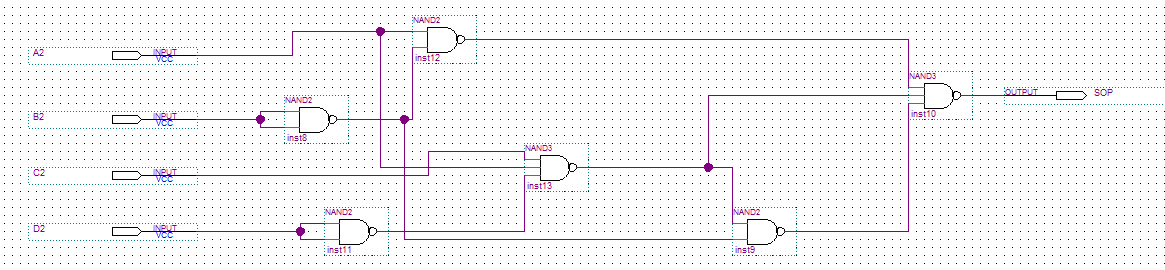
****

**Figure 1.1.1:** Screen-shot of the combinational logic circuit minimized by Boolean algebra diagram.

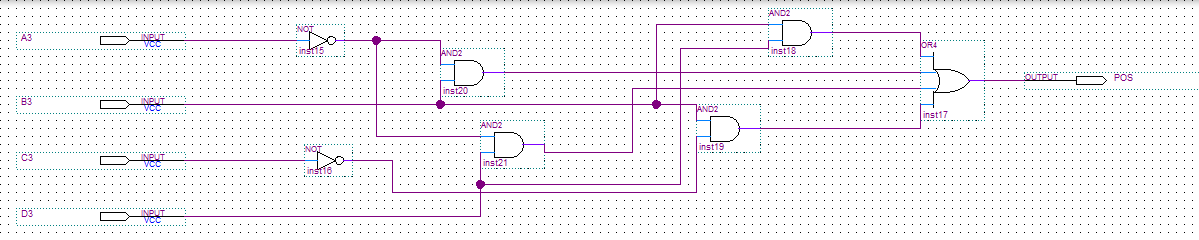
****

**Figure 1.1.2:** Screen-shot of the simplified combinational logic circuit minimized by Boolean algebra diagram.

**Part II - Combinational Logic Circuits Minimization by the Karnaugh Map Method**

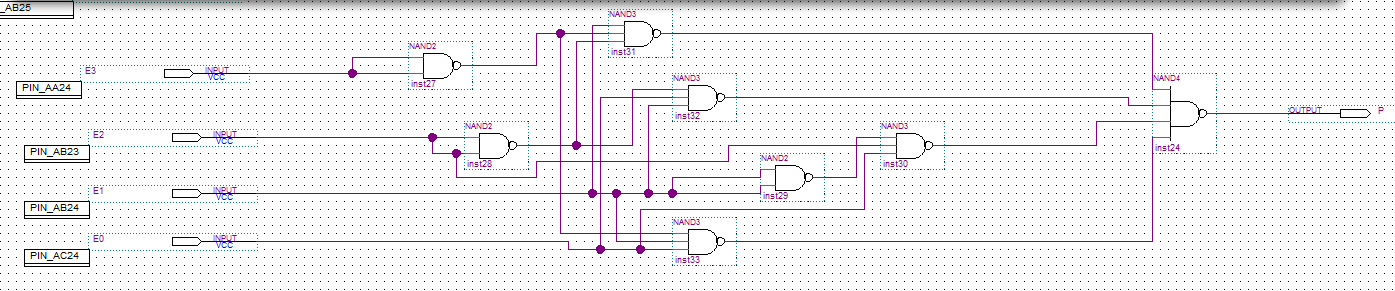
****

**Figure 2.1.1:** Screen-shot of the SOP combinational logic circuit minimized by Karnaugh Map Method diagram.

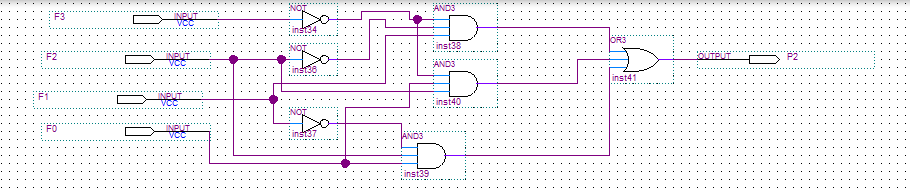


**Figure 2.1.2:** Screen-shot of the POS combinational logic circuit minimized by Karnaugh Map Method diagram.

**Part III – Design of Combinational Logic Circuits**



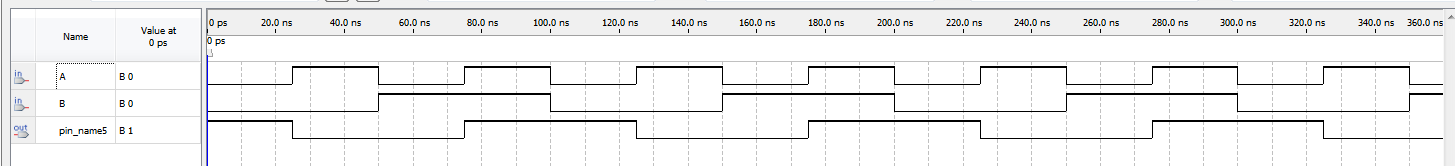
**Figure 3.1.1:** Screen-shot of the combinational logic circuit using only NAND diagram.



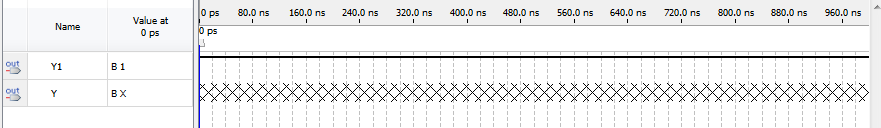
**Figure 3.1.2:** Screen-shot of the simplified combinational logic circuit.

**Experimental Data and Data Processing**

**Part 1 – Combinational Logic Circuits Minimization by Boolean Algebra**



**Figure 1.2.1:** Screen-shot of the simulation output of combinational logic circuits minimized by Boolean algebra.

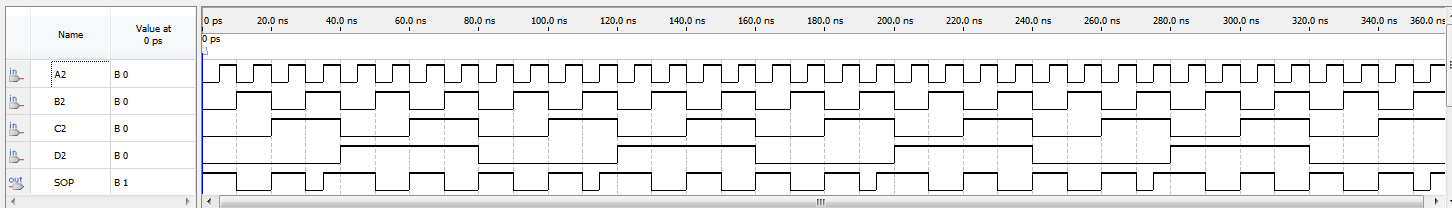


**Figure 1.2.2:** Screen-shot of the simulation output of simplified combinational logic circuits minimized by Boolean algebra.

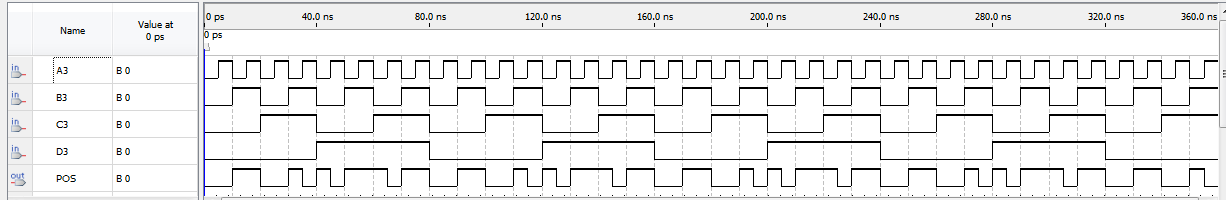
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input Given** | | | | **Observed Output** |
| **A** | **B** | **C** | **D** | **Y** |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Table 1.3:** Truth table for the combinational logic circuits minimization by Boolean algebra.

**Part II - Combinational Logic Circuits Minimization by the Karnaugh Map Method**

****

**Figure 2.2.1:** Screen-shot of the simulation output of the SOP combinational logic circuit minimized by Karnaugh Map Method.

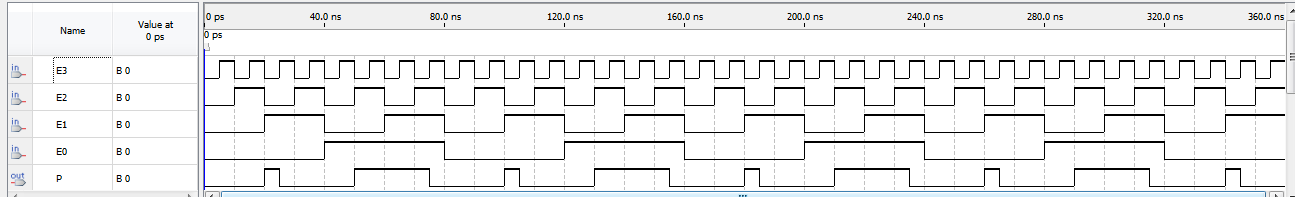


**Figure 2.2.2:** Screen-shot of the simulation output of the POS combinational logic circuit minimized by Karnaugh Map Method.

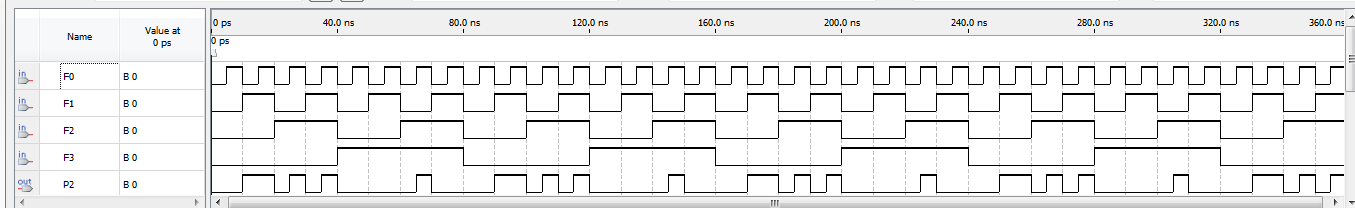
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input Given** | | | | **Observed Output** | |
| **A3** | **B3** | **C3** | **D3** | **SOP** | **POS** |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |

**Table 2.3:** Truth table for the combinational logic circuits minimization by the Karnaugh map method.

**Part III – Design of Combinational Logic Circuits**



**Figure 3.2.1:** Screen-shot of the simulation output of combinational logic circuit using only NAND.



**Figure 3.2.2:** Screen-shot of the simulation output of the simplified combinational logic circuit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input Given** | | | | **Observed Output** |
| **D3** | **D2** | **D1** | **D0** | **P** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

**Table 3.3:** Truth table for the combinational logic circuits minimization by the Karnaugh map method.

**Comparison of Theoretical Data and Experimental Data**

**Part 1 – Combinational Logic Circuits Minimization by Boolean Algebra**

Tautology Logic Circuit (Circuit Simplified with Boolean Algebra)

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Given** | | **Observed Results** | **Expected Results** |
| **A** | **B** | **Y** | **Y** |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

**Table 1.4:** Comparison of the theoretical and experimental results of the tautology logic circuit.

As expected with a tautology logic circuit all the possible input combinations resulted in the expression being true. Though the laws of Boolean logic the expression was reduced to just 2 variables (A, B) from its original 4 variables (A, B, C, D). The results observed experimentally were identical to the theoretical results computed for this circuit.

**Part II - Combinational Logic Circuits Minimization by the Karnaugh Map Method**

Combinational Logic Circuit Simplified with Karnaugh Map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input Given** | | | | **Observed Results** | **Expected Results** |
| **A3** | **B3** | **C3** | **D3** | **POS** | **POS** |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

**Table 2.4:** Comparison of the theoretical and experimental results of a combinational logic circuit simplified with a K-map.

The Original Boolean expression was quite complicated and would have contained an insane amount of logic gates should it have been implemented into a logic circuit. Through the use of K-maps the expression was simplified into a much more manageable logic circuit. The experimental results viewed on the circuit board were equal to those calculated theoretically for this logical expression. When this circuit was implemented using NAND gates only the inputs and outputs remained the same as the theoretical.

**Part III – Design of Combinational Logic Circuits**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input Given** | | | | **Observed Results** | **Expected Results** |
| **D3** | **D2** | **D1** | **D0** | **P** | **P** |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

**Table 3.4:** Comparison of the theoretical and experimental results of a combinational logic circuit.

This was yet another quite complex Boolean expression. Through the use of the SOP and the K-map, the expression was taken down to a much more manageable size. The experimental results of this constructed logic circuit matches the previously computed theoretical results. When the simplified SOP underwent manipulation via De Morgan’s law to implement using NAND gates only the experimental and theoretical results were once again the same.

**Discussion and Conclusions**

The objective of this experiment was to simplify logic functions starting from their non-simplified Boolean expressions, create diagrams on Quartus, simulate, and then apply these logic circuits to the Altera circuit board. Devise and design combination logic circuits from specifications using any type of available logic gates or by using NAND gates only. The methodology of this experiment was to apply our knowledge of combinational logic circuits minimization by Boolean Algebra, combinational logic circuits minimization by the Karnaugh map method, and design of combinational logic circuits. The experimental results of our lab matched our expected results, this means our calculations in our pre-lab were correct. This experiment used De Morgan’s theorem in part 1 to simplify the given Boolean expression and to convert the logical expressions to utilize NAND gates only. In part 2, we used the rules of Karnaugh maps (no zeros allowed, no diagonals, only power of 2 number of cells in each group, groups should be as large as possible, everyone must be in at least one group, overlapping allowed, wrap around allowed, fewest number of groups possible) to simplify the SOP. We did not have any deviations in our results nor did we have problems throughout the lab.

**Appendix (Pre-Lab)**

See the following pages for the pre-lab predictions written for this lab.