EE 243 – Project 1

BCD Multiplier

Ahmet Furkan Akıncı
2020401093

2022 – 2023 Fall

November 27, 2022

TABLE OF CONTENTS

1– Introduction

In the project, it is aimed to design a BCD multiplier that multiplies a 2-digit BCD number with a 1-digit BCD number and returns the result as 3-digit BCD number. In addition, a seven-segment decoder that convert a BCD number to a seven-segment form aimed to design. All the circuit elements expected to be properly functioning in the simulation.

2 – Design and Method

2.1 – BCD multiplier (Top Level Design)

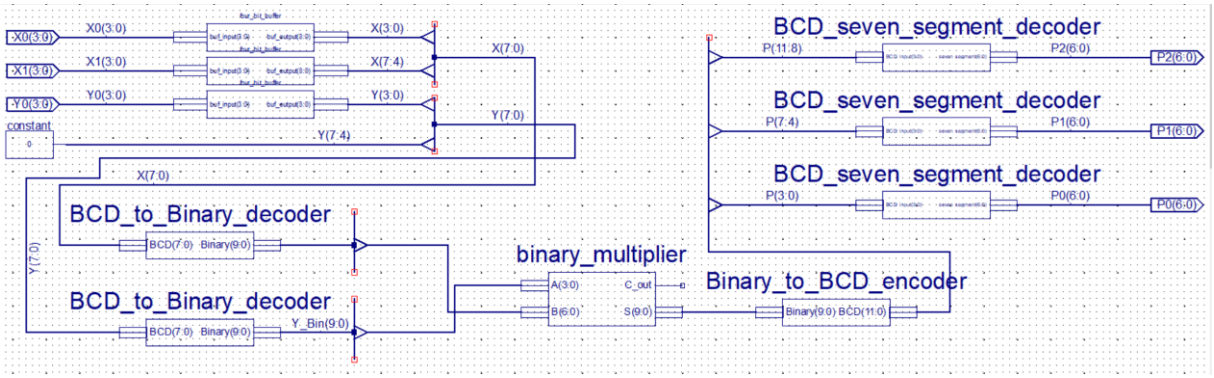


Figure 1 - BCD multiplier schematic

In order to accomplish desired task, a circuit with main components such as BCD to Binary Decoder, Binary Multiplier, Binary to BCD Encoder and BCD seven segment decoder is designed. It is preferred to convert BCD numbers to binary numbers then multiplying the binary numbers. Because, in that way calculations become simpler.

As the input the circuit takes 3 BCD number and produces 3 number in the seven-segment decoded form.

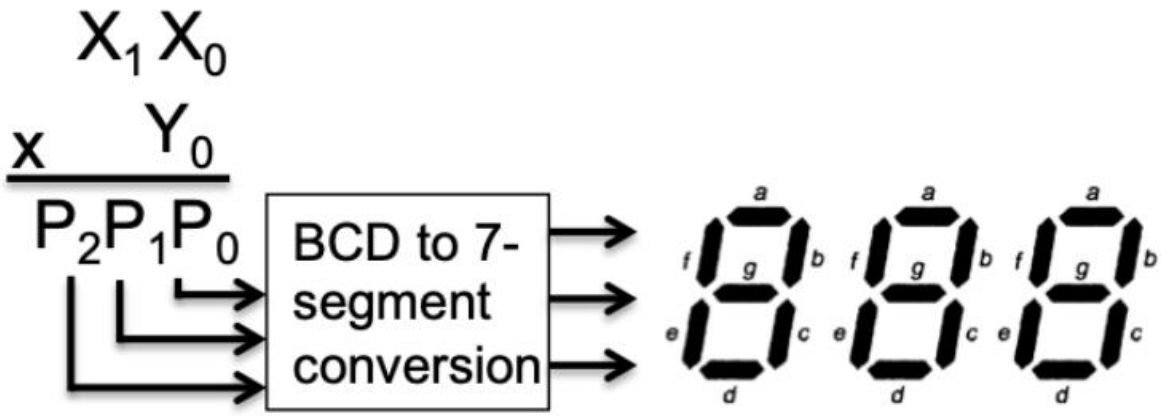


Figure 2 - algorithm summary

2.2 – Binary Multiplier

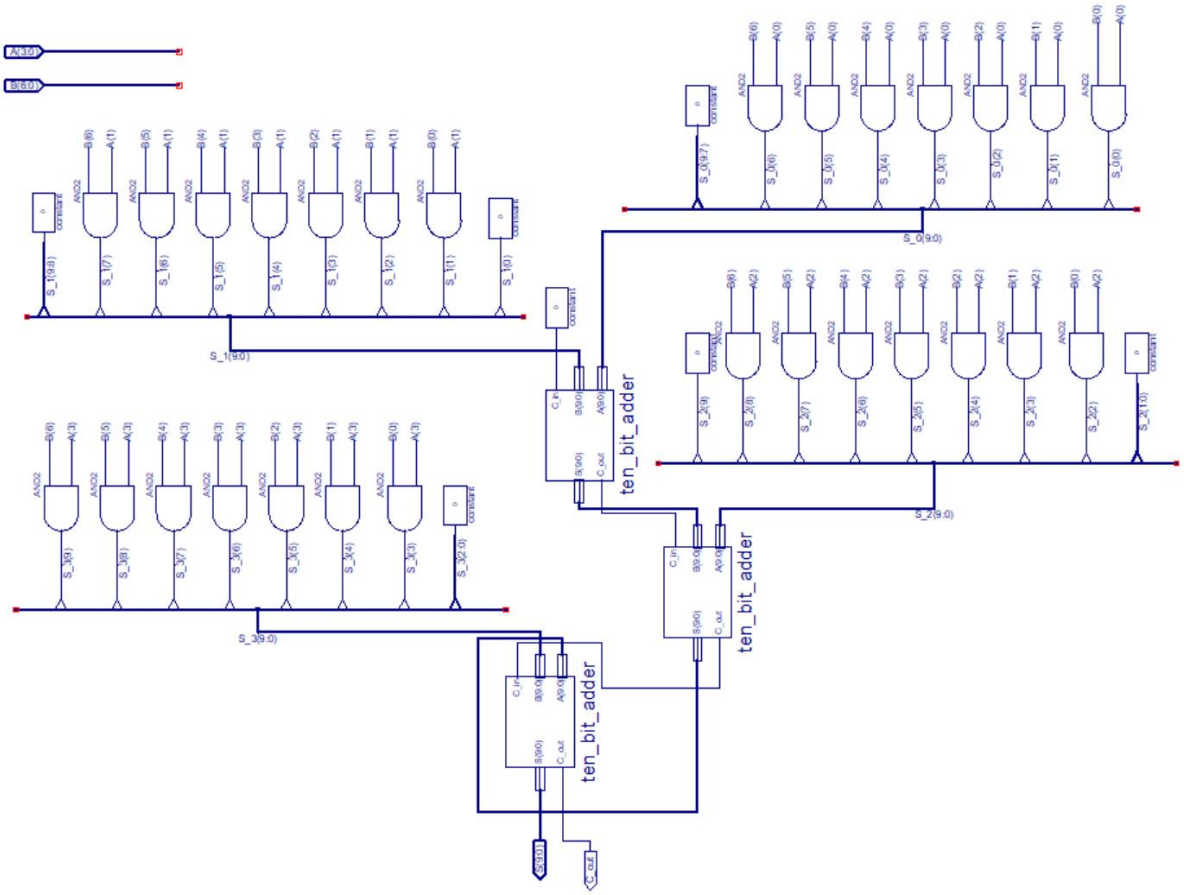


Figure 3 - Binary multiplier schematic

The Binary Multiplier multiplies a 7-digit binary number with a 4-digit binary dumber and produces 10-digit binary number as output. The method is multiplying every digit in 7-digit number with one of the digit of 4-digit number. For every digit starting from the right-hand side, one digit space is inserted to the end of number. At the end, produced numbers are summed.

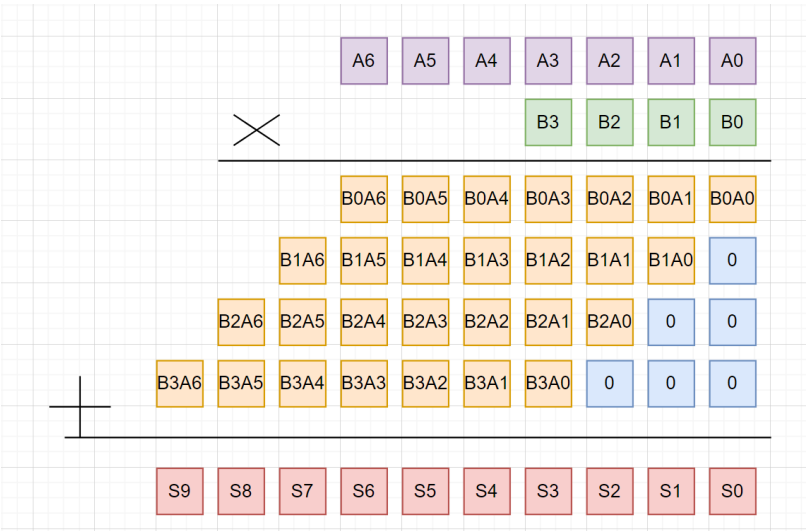


Figure 4 - Binary multiplication algorithm

2.2.1 – Ten Bit Adder, Four Bit Adder, Full Adder, Half Adder

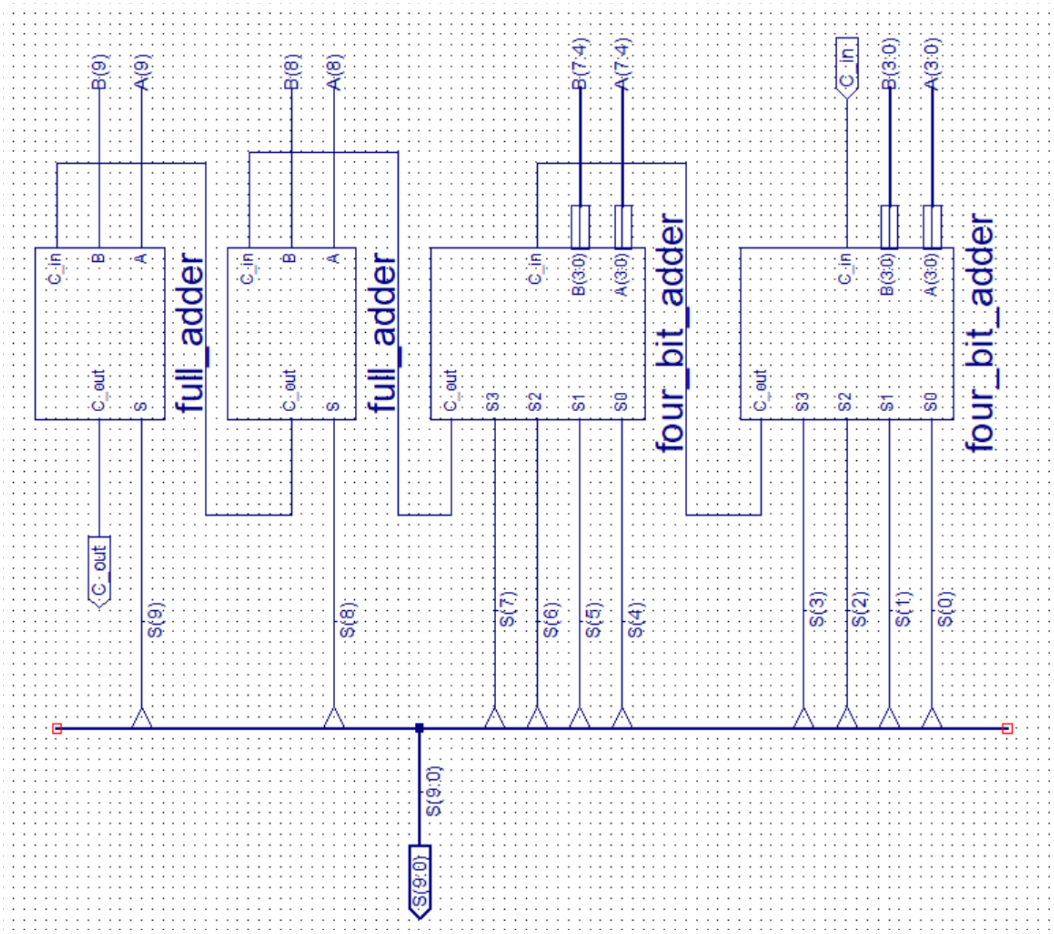


Figure 5 - ten bit adder schematic

Ten Bit Adder is composed of 2 full-adder and 2 four-bit adder. Every four-bit adder is also composed of a four full adder. In addition, every full adder is composed of 2 half adder. Adders except half adders, get 3 binary number (A, B, C_in) to be summed. As the output they return a binary number (S) whose first digit is specified as C_out. Add operation is done by using “and” and “xor” gates.

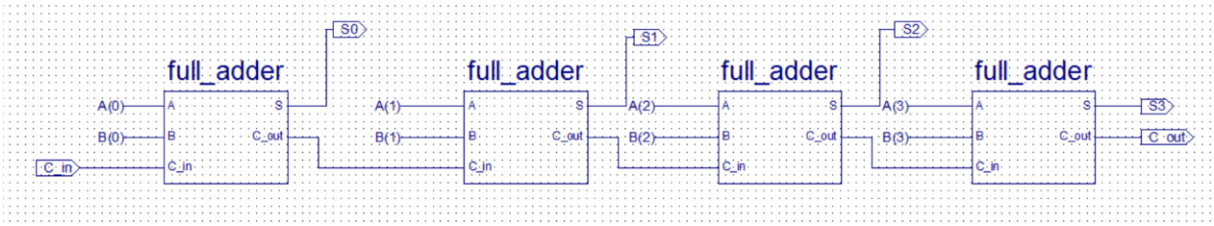


Figure 6 - four bit adder schematic

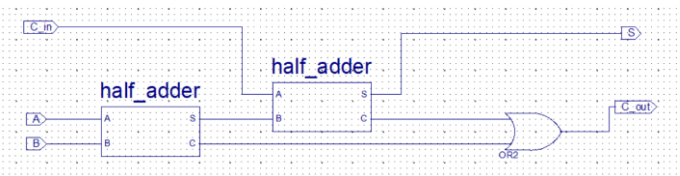


Figure 7 - full adder schematic

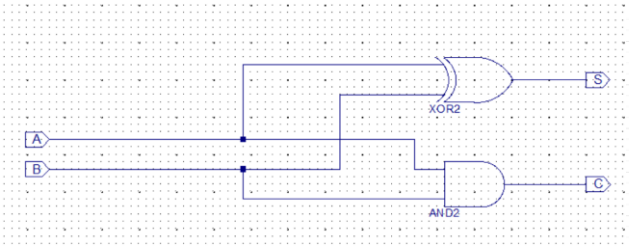


Figure 8 - half adder K-maps

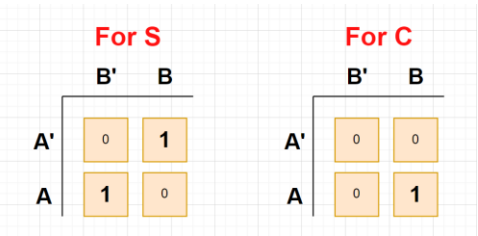


Figure 9 - half adder scheamtic

2.3 – BCD to Binary Decoder

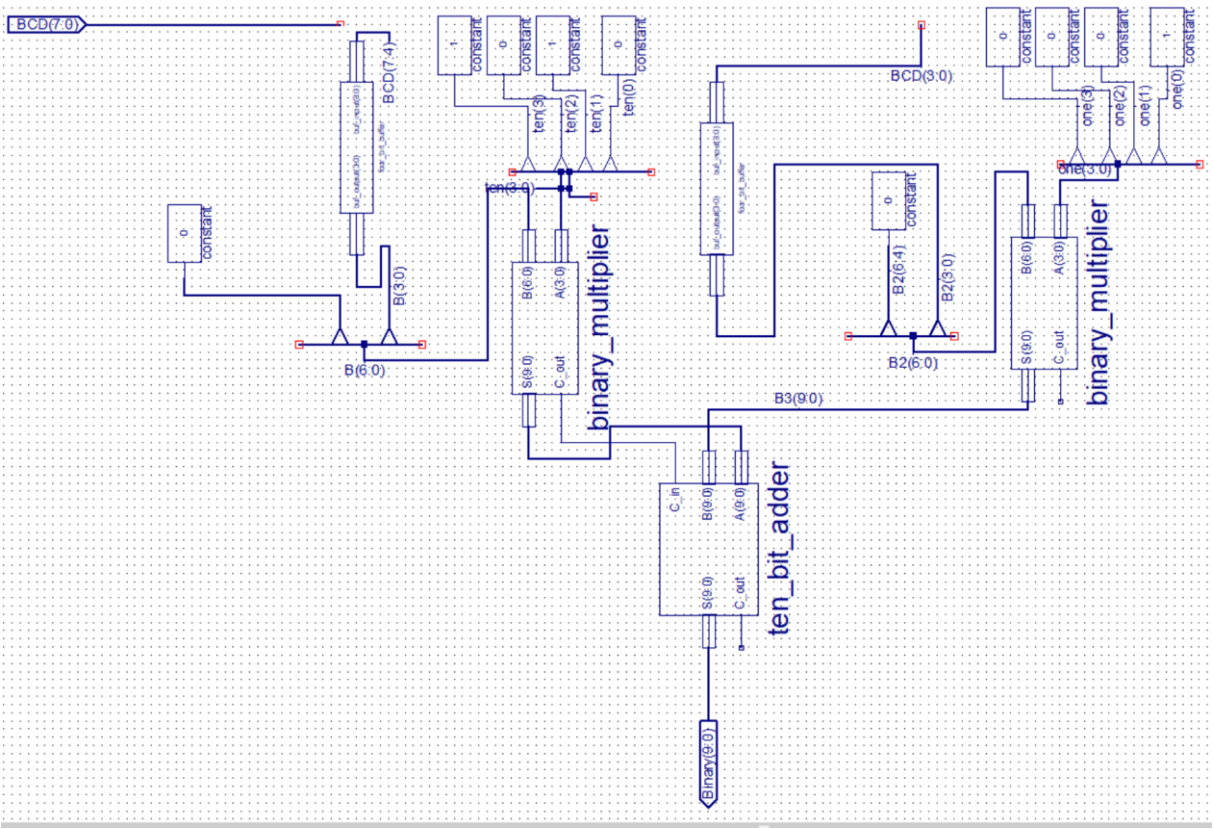


Figure 10 - BCD to Binary decoder schematic

The BCD to Binary Decoder gets a 8 bit BCD number, simply multiplies first four digit with “0001” (1 in decimal) and multiplies next four digit with “1010” (10 in decimal). Finally, add two results to give binary output.

2.4 – Binary to BCD encoder

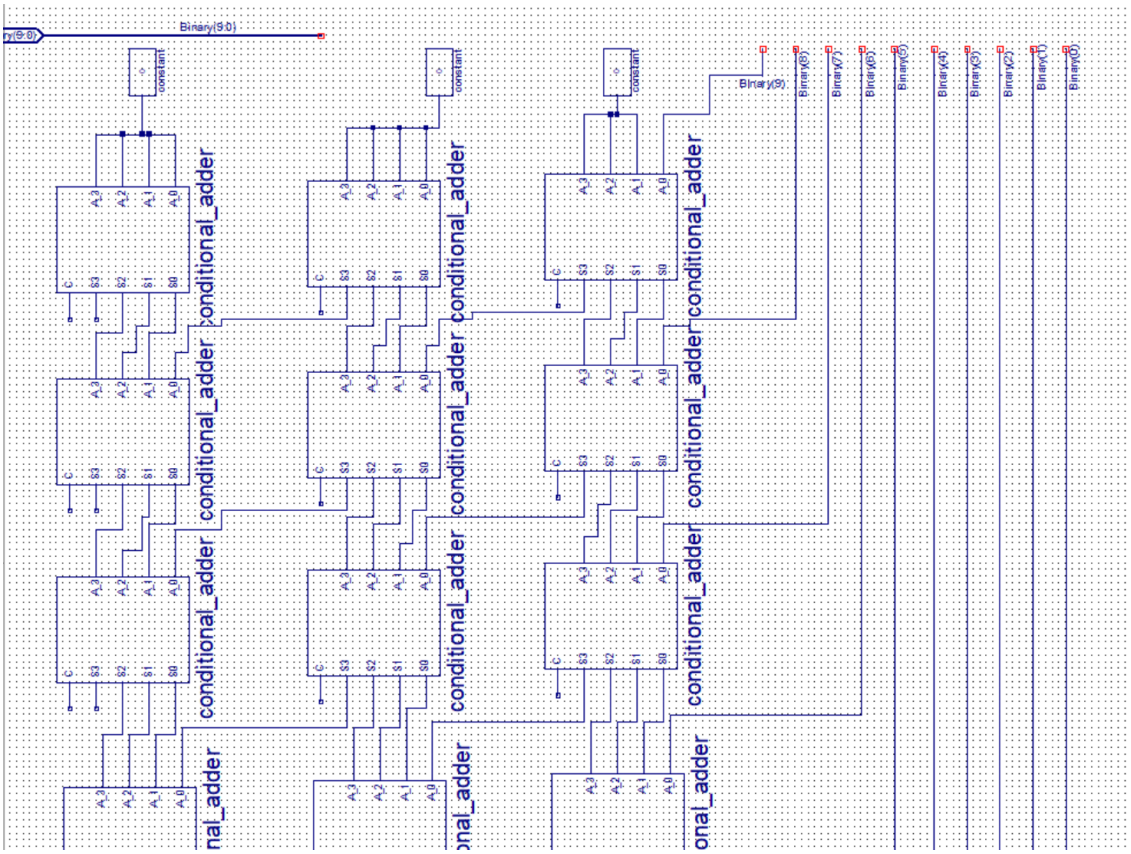


Figure 11 - Binary to BCD encoder (part of the schematic)

For converting Binary numbers to BCD numbers, a circuit that performs double-dabble algorithm is preferred. Double – dabble algorithm imply to shift the binary input one digit from right to left and for every shift BCD number should be checked. If the checked number is greater than 4, 3 is added to that number.

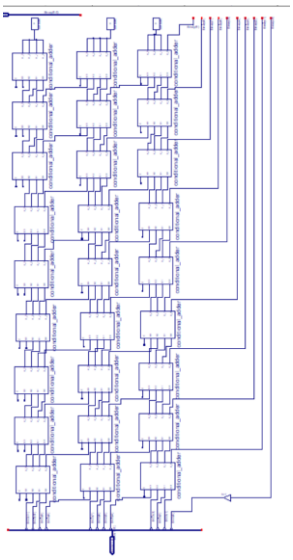


Figure 13 - Binary to BCD encoder schematic

Example for Double – Dabble Alghoritm, converting 1100011 to BCD

0 0 0 0 0 0 0 0	1 1 0 0 0 1 1	initialize
0 0 0 0 0 0 0 1	1 0 0 0 1 1 0	shift 1 bit left
0 0 0 0 0 0 1 1	0 0 0 1 1 0 0	shift 1 bit left
0 0 0 0 0 1 1 0	0 0 1 1 0 0 0	shift 1 bit left
0 0 0 0 1 0 0 1	0 0 1 1 0 0 0	add 3, 6 > 4
0 0 0 1 0 0 1 0	0 1 1 0 0 0 0	shift 1 bit left
0 0 1 0 0 1 0 0	1 1 0 0 0 0 0	shift 1 bit left
0 1 0 0 1 0 0 1	1 0 0 0 0 0 0	shift 1 bit left
0 1 0 0 1 1 0 0	1 0 0 0 0 0 0	add 3, 9 > 4
1 0 0 1 1 0 0 1	0 0 0 0 0 0 0	shift 1 bit left

Figure 12 - Example Double - Dabble Algorithm

2.4.1 – Conditional Adder, Magnitude Comparator

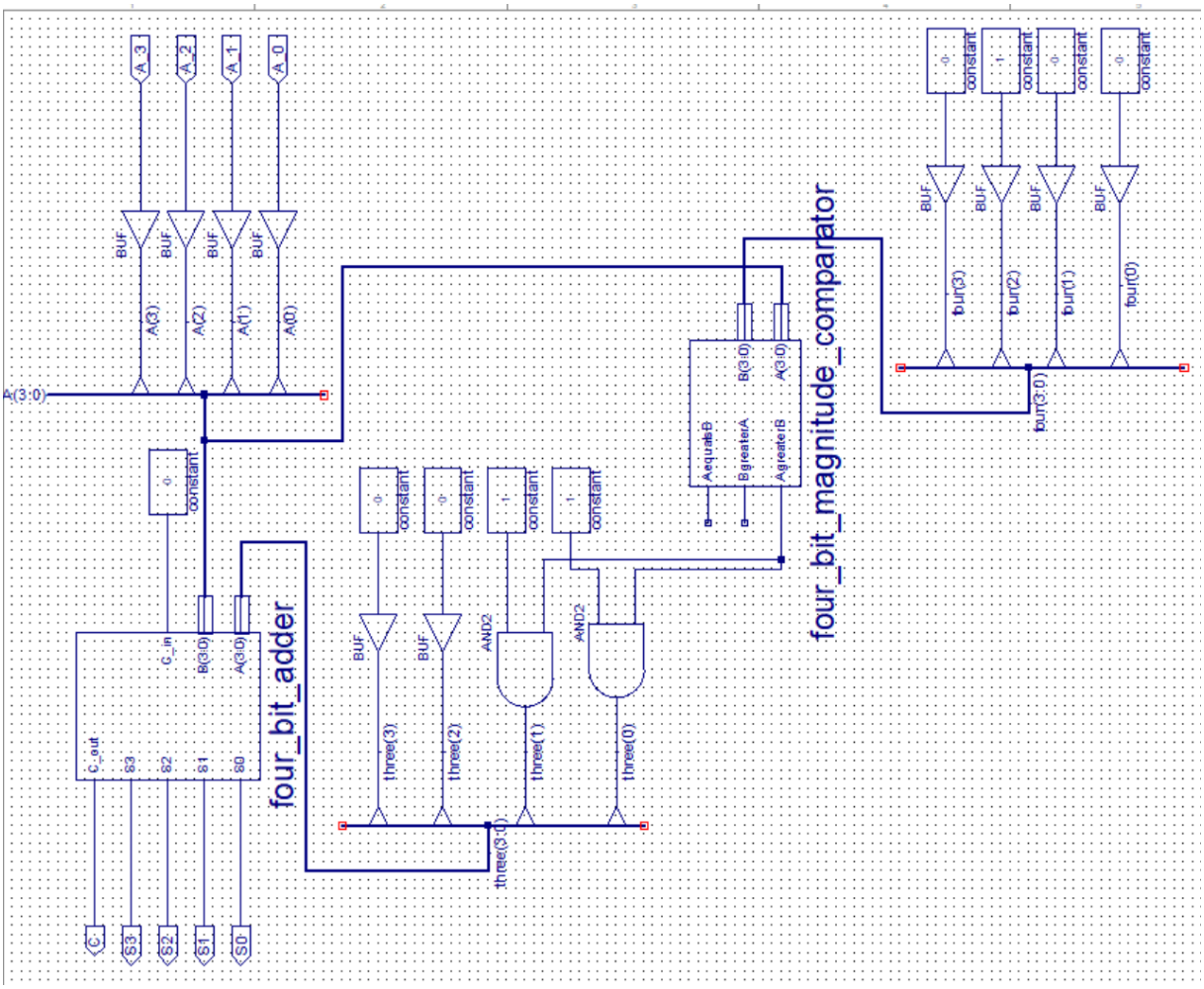


Figure 14 - Conditional adder schematic

The Conditional Adder checks if given input is greater than 4. If the given input is greater than 4, with four bit adder 3 is added to input. If the given input is not greater than 4, with four-bit adder 0 is added to input.

The Magnitude Comparator compares the bits of the inputs starting from the most significant bit. It has 3 outputs as “A>B”, “A<B” and “A=B”.

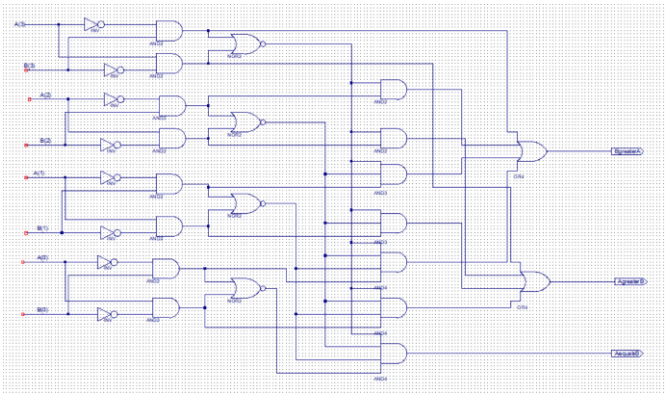


Figure 15 - Magnitude comperator schematic

2.5 – Seven segment decoder

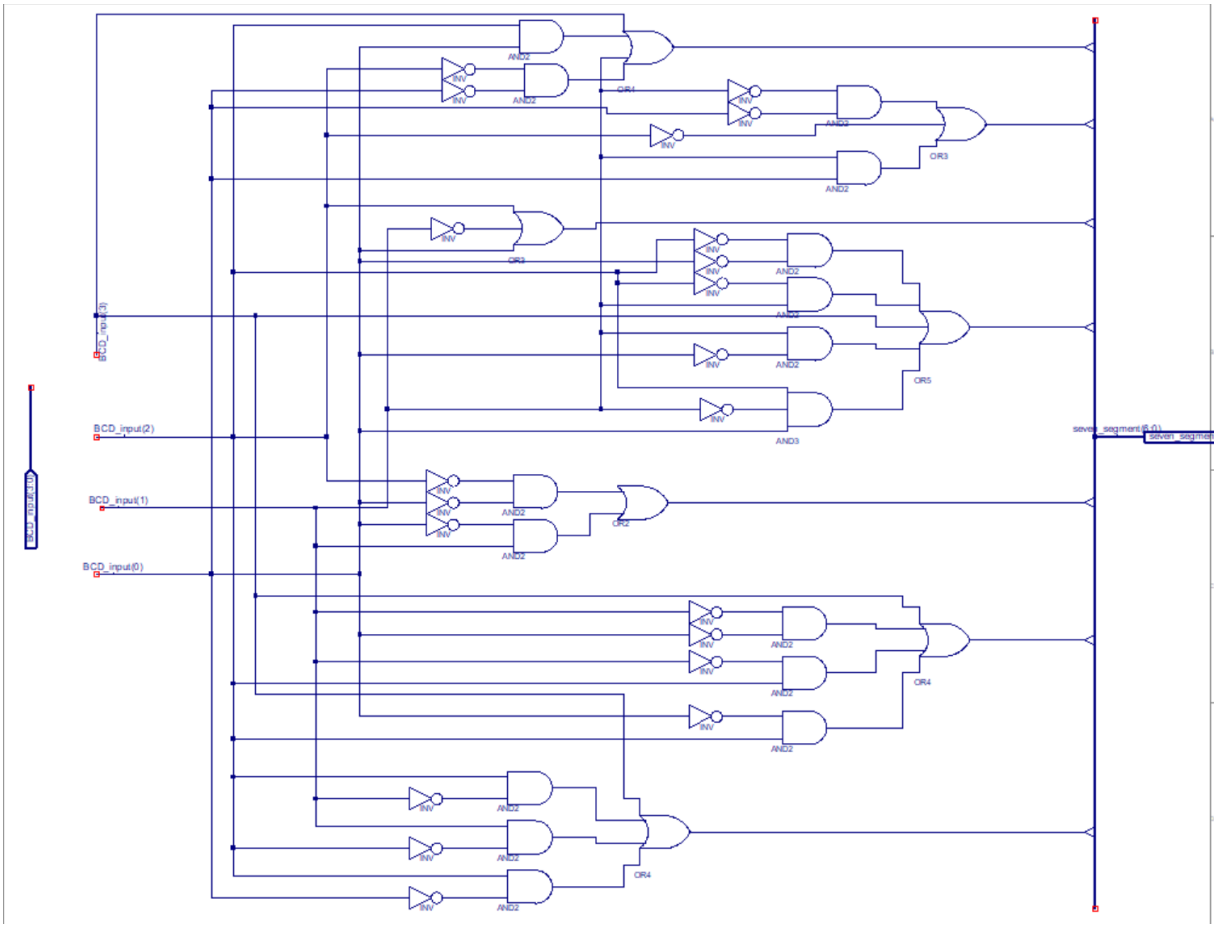


Figure 16 - Seven segment decoder schematic

The seven-segment decoder, convert the BCD number to the seven-segment form to show it on a display. Its outputs are a, b, c, d, e, f and g. Their meanings are showed in the below image. While creating the circuit K-maps are calculated for each input. The example K-map for output a is provided below.

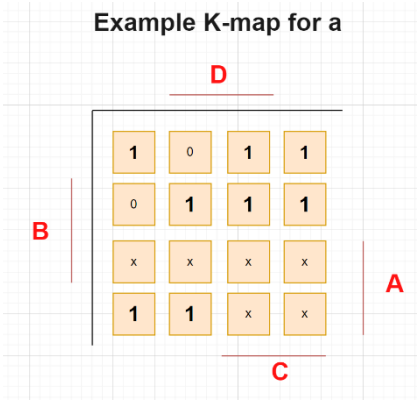


Figure 18 - Example K-map for a

Figure 19 - seven segment outputs

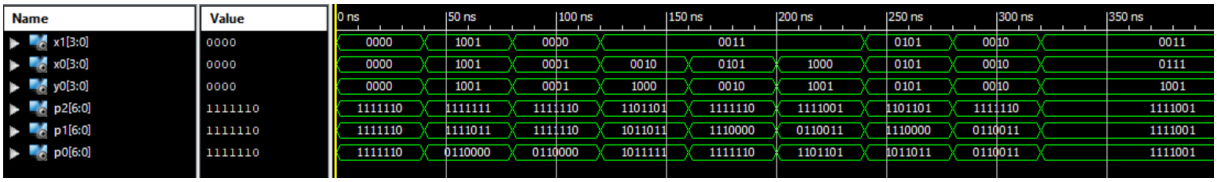
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

Figure 17 - input - output table for 7 segment decoder

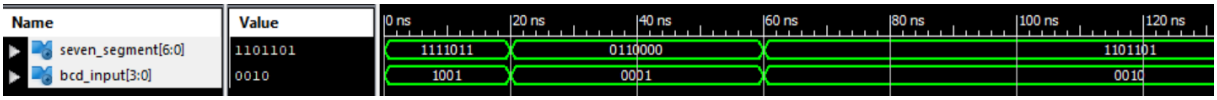
3 – Analysis and Tests

3.1 – Tests

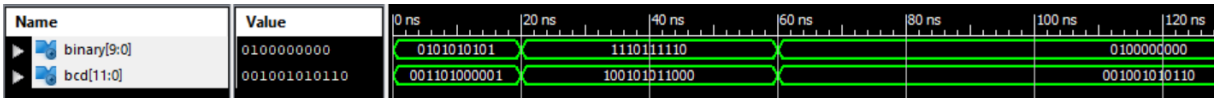
3.1.1 – BCD Multiplier



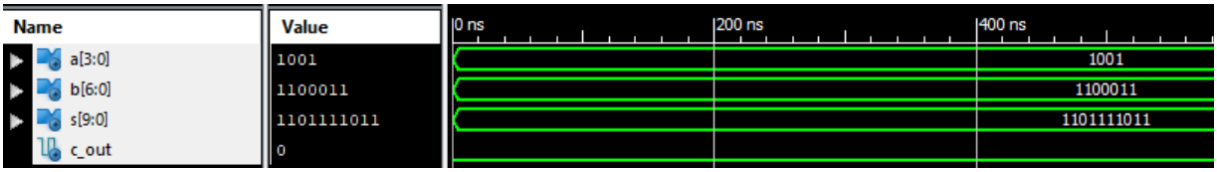
3.1.2 – Seven Segment Decoder



3.1.3 – Binary to BCD encoder



3.1.4 – Binary Multiplier



3.2 – Result

The design can successfully perform the desired operation.

4 – References

[1] https://www.geeksforgeeks.org/bcd-to-7-segment-decoder/

[2] https://en.wikipedia.org/wiki/Double_dabble

[3] https://www.electronicshub.org/binary-multiplication/

[4] http://users.encs.concordia.ca/~asim/COEN_6511/Projects/final6511report.pdf