

Digital Systems ( ENCS 2340)

Course Project

( BCD Adder / Subtractor )

Verilog Code

Student’s Name: Pierre Backleh

Student’s No: 1201296

Date : 02-01-2022

Dr. Mohammed Hussain (section 2)

Table of Contents

[Introduction 3](#_Toc92021196)

[Overview 3](#_Toc92021197)

[Components 4](#_Toc92021198)

[*Full Adder:* 4](#_Toc92021199)

[*Half Adder:* 5](#_Toc92021200)

[*2x1 Multiplexer:* 5](#_Toc92021201)

[6](#_Toc92021202)

[*9’s Complement:* 6](#_Toc92021203)

[BCD Adder / Subtractor 7](#_Toc92021204)

[*BCD Adder Code:* 8](#_Toc92021205)

[*BCD Adder / Subtractor :* 9](#_Toc92021206)

[BCD Simulation 10](#_Toc92021207)

[Block Diagram for BCD Adder - Subtractor Using Half adders and full Adders 11](#_Toc92021208)

[Project explanation 12](#_Toc92021209)

[Wave Form explanation 14](#_Toc92021210)

[*Output of the addition* 15](#_Toc92021211)

[*Output of the subtraction* 16](#_Toc92021212)

[References 17](#_Toc92021213)

Introduction

In this report, we will represent the final code which we have designed using Verilog HDL language using Quartus 2.

This code contains five modules with various methods and functions.

Overview

We designed many components to use them in this project.

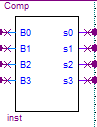
Full adder:

We used:

Four full adders

Three half adders

Four 2x1 multiplexers



One 9’s complement

Two AND gates

One OR gate

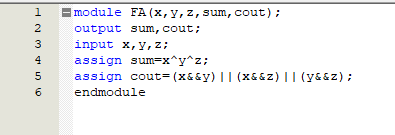


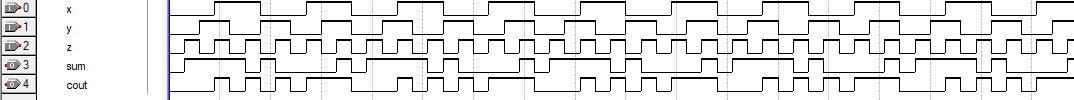
Components

## Full Adder:

Full adder circuit takes three inputs (x, y, z) and returns two outputs (sum, carry).

Full Adder Code:

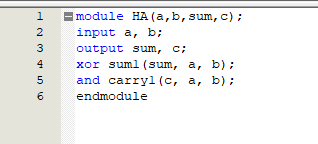
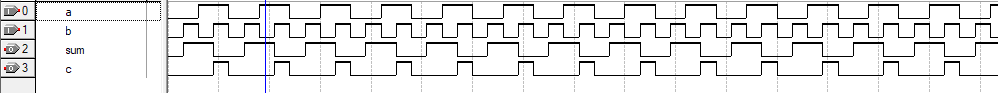




## Half Adder:

Half adder circuit takes two inputs (a, b) and returns two outputs (sum, c).

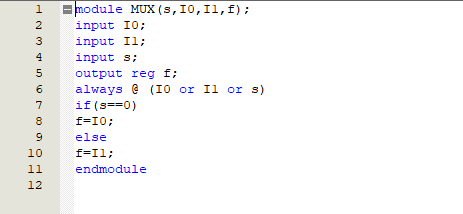
Half Adder Code:



## 2x1 Multiplexer:

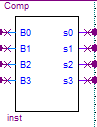
Multiplexer circuit takes two inputs and a selection line to choose one of the inputs and returns it as an output.

2x1 Multiplexer Code:

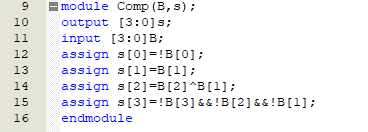


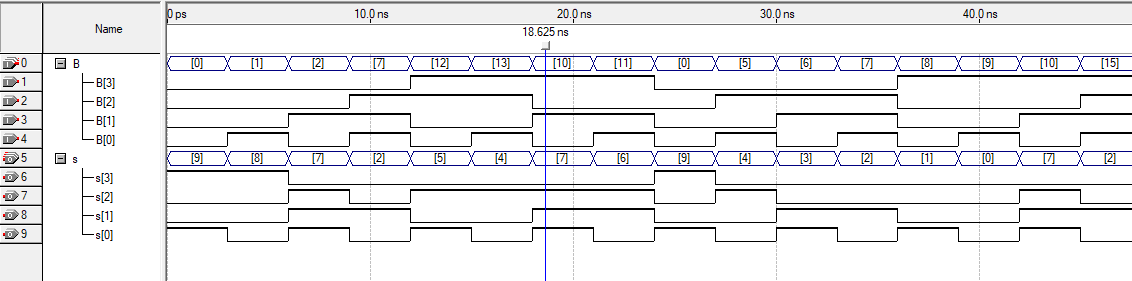
## 

## 9’s Complement:

This circuit is used to find the subtraction of the decimal numbers. In this project, the 9’s complements takes four inputs and returns four outputs.

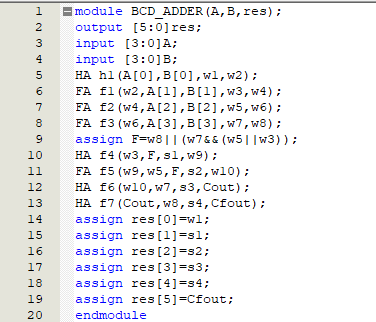
9’s Complement Code:



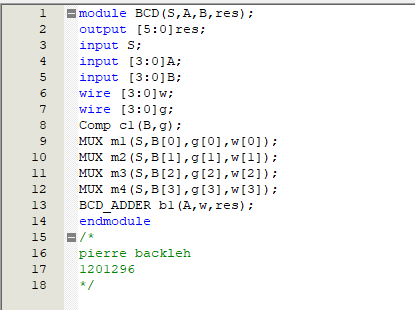


BCD Adder / Subtractor

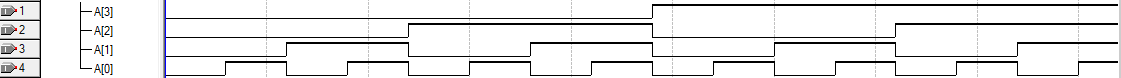
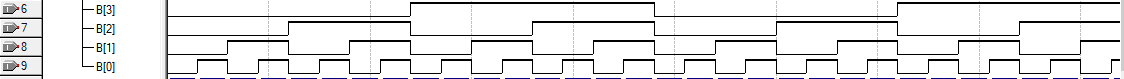
## BCD Adder Code:

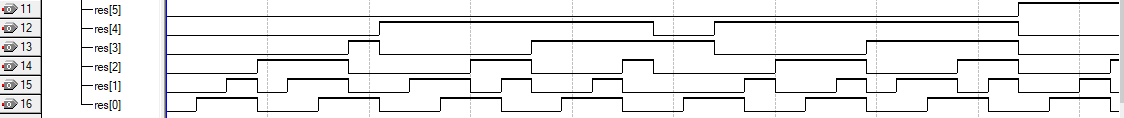


*Main Code:*



Simulation for BCD Code:

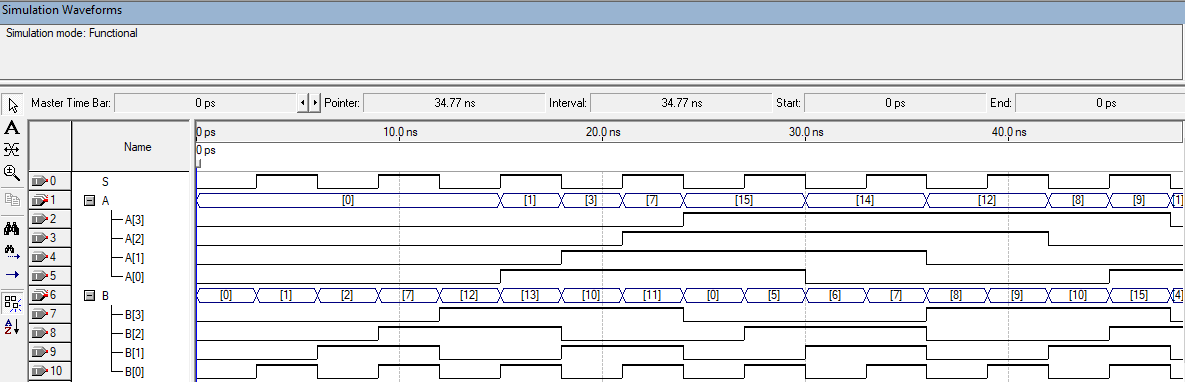
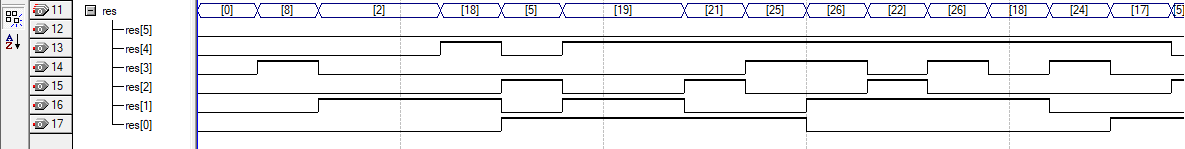


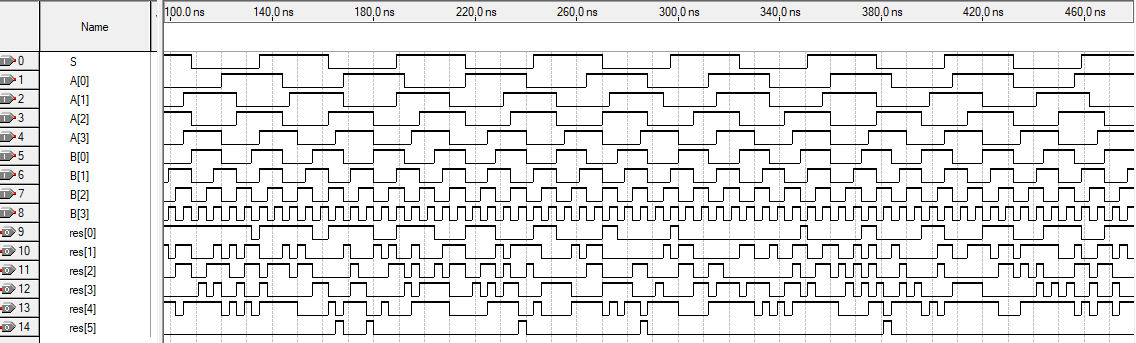


## BCD Adder / Subtractor :

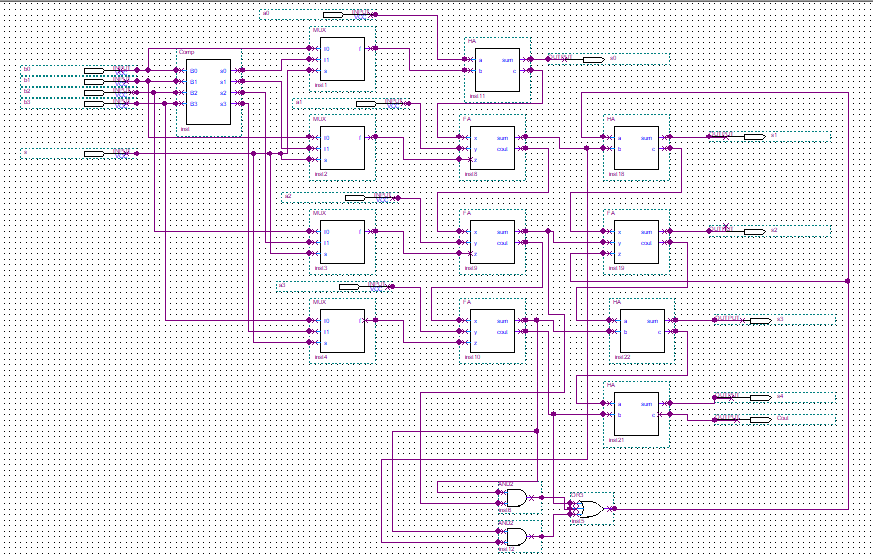
The main code is used to pass the second number to the 9’s complement circuit to subtract it from the first number. If the selection line value is zero, the circuit will add the first number to the second one and the result will be the output. But if the selection line value is one, the circuit will subtract the second number from the first one and the result will be the output.

BCD Simulation





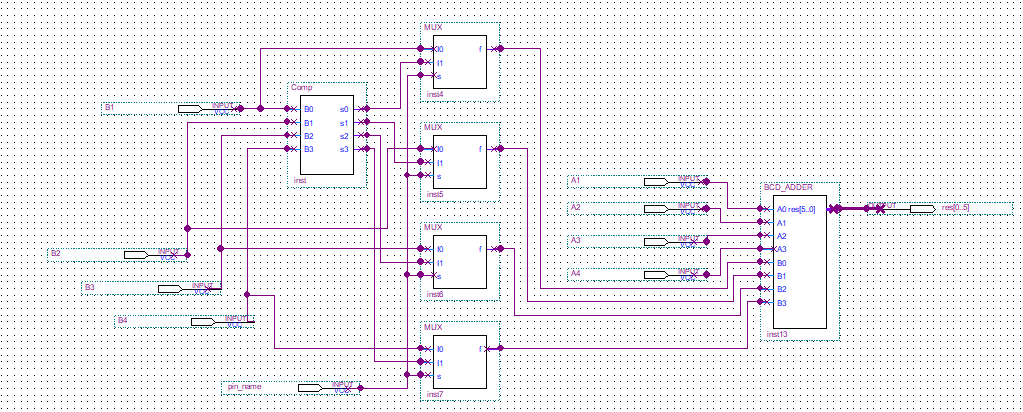
Block Diagram for BCD Adder - Subtractor Using Half adders and full Adders



This part for adding 0110 if the number

Is greater than 9

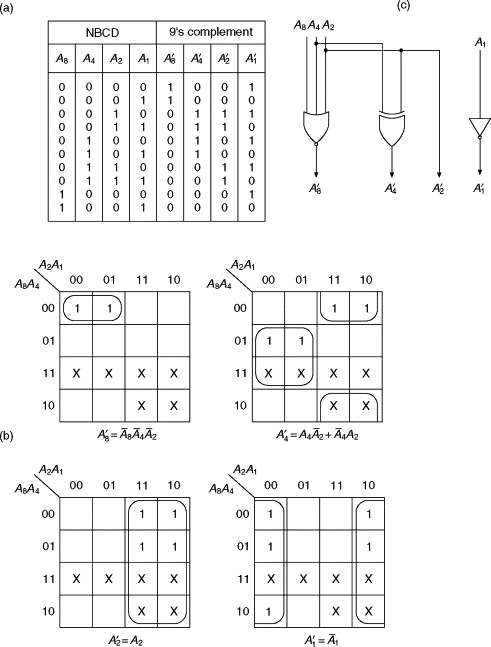
Block Diagram



Project explanation

We designed the 9’s complement according to this truth table.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| input | | | | output | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |



The truth table for the – 0110 – adder if the number is greater than 9:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| input | | | | output |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 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 |

K-map

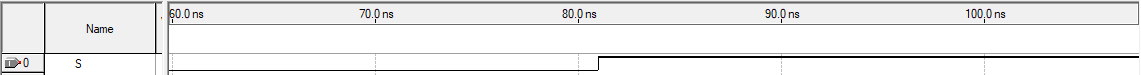
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB \ CD | 00 | 01 | 11 | 10 |
| 00 |  |  |  |  |
| 01 |  |  |  |  |
| 11 | 1 | 1 | 1 | 1 |
| 10 |  |  | 1 | 1 |

Output= A.B + A.C + Carry out

Wave Form explanation

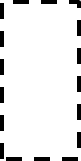
## Output of the addition

Selection line:



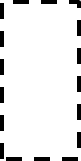
 Input for the first number:

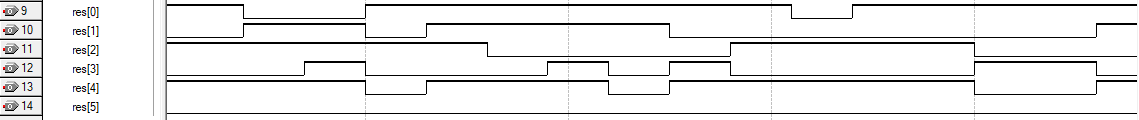
Selected section : 0011

Input for the second number:



Selected section : 0000

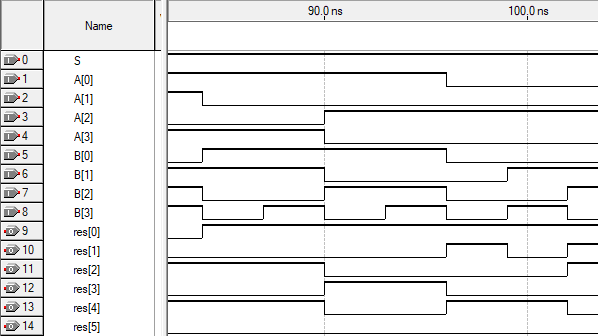
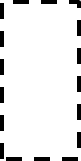
The BCD adder result (Output):



The output for the selected section : 00011

Final carry out the result of the addition

## Output of the subtraction



Selected section inputs:

S=1 ( subtraction )

A[0]=1 A[1]=0 A[2]=1 A[3]=0 A=0101

B[0]=1 B[1]=0 B[2]=1 B[3]=0 B=0101

Selected section outputs:

Res[0]=1

Res[1]=0

Res[2]=0 Res = 001001

Res[3]=1

Res[4]=0

Res[5]=0

References

[1] <https://www.sciencedirect.com/topics/computer-science/logic-circuitry>