-- Alexandra Zajda (260807517)
-- Nicolas Barreyro (260730549)
-- Gorkem Yalcinoz (260710053)
-- GROUP 12

ECSE 222- DIGITAL LOGIC

LABORATORY REPORT #1 :

g12_1_bit_Adder_Subtractor Circuit Summary and Functionality

-- Alexandra Zajda (260807517)
-- Nicolas Barreyro (260730549)
-- Gorkem Yalcinoz (260710053)
-- GROUP 12

TABLE OF CONTENTS

-- Alexandra Zajda (260807517)
-- Nicolas Barreyro (260730549)
-- Gorkem Yalcinoz (260710053)
-- GROUP 12

INTRODUCTION

The goal of the first lab was to design a circuit, or a combination of circuits, capable of doing simple 1-bit addition or 1-bit subtraction. This report will describe in detail how such a circuit was implemented, showing all inputs, outputs and a logic-gate analysis of the circuit. It will also describe how the design of this circuit tested and proved to be functional, using Quartus II's waveform simulation tool for VHDL code.

PART 1 : g12_1_bit_Adder_Subtractor_Circuit

The 1-bit_Adder_Substractor circuit that we implemented is composed of 2 sub-circuits: a **1-bit adder** and a **1-bit subtractor**. The two sub-circuits have 2 common 1-bit inputs: *a* and *b.* In the adder sub-circuit, a and b are treated as (2) digits that are added. In the subtractor, a is treated as the minuend and b as the subtrahend.

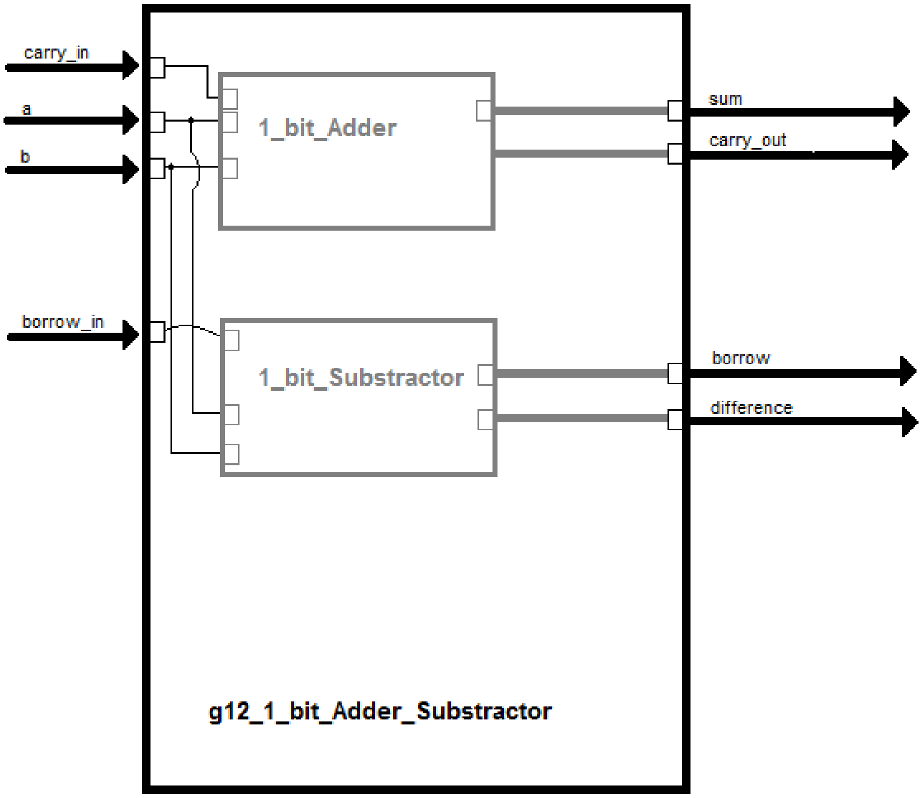


FIGURE 1 : Pinout Diagram of g12_1_bit_Adder_Subtractor

The (2) other inputs, carry-in and borrow-in, are exclusively used in either the 1-bit adder sub-circuit (for carry-in) or in the 1-bit subtractor sub-circuit (for borrow_in). Figure 1 provides a list of all the inputs and outputs of the 1-bit_Adder_Subtractor circuit, and shows in which sub-circuit these signals are used.

A list of all inputs/outputs in the overall circuit is specified in table 1, for added clarity:

SIGNAL NAME	PRESENT IN ADDER SUBCIRCUIT	PRESENT IN SUBTRACTOR SUBCIRCUIT
IMPUT NAME		
a	x	x
b	x	x
carry_in	x	
borrow_in		x
OUTPUT NAME		
sum	x	
carry_out	x	
difference		x
borrow		x

TABLE 1 : Inputs/Outputs of g12_1_bit_Adder_Subtractor

For a logic gate-level implementation of the circuit, see *Annexe 1* at the end of the report. This implementation is represented in our VHDL files attached to this report.

SIMULATION PROCEDURE EXPLANATION

To test if the circuit was adequate, we ran a waveform simulation of the circuit using Altera Quartus II software. Each input was set to be toggled from 0/1 at varying time intervals, so that all possible input combinations would appear in the simulation. For example, in the 1-bit adder circuit simulation, the inputs were toggled at 2ns, 4ns and 8ns respectively.

To make it clearer, we isolated the imputs/outputs relevant to each sub-circuit in a separate waveform simulation and compared the results of it to the truth table that was provided in our lab instructions. We saw that the waveforms corresponded to the truth tables for all entries of the truth tables, thus making our circuit adequate to represent the addition/subtraction functions shown on the truth tables.

-- Alexandra Zajda (260807517)
-- Nicolas Barreyro (260730549)
-- Gorkem Yalcinoz (260710053)
-- GROUP 12

PART 2 : 1-bit Adder

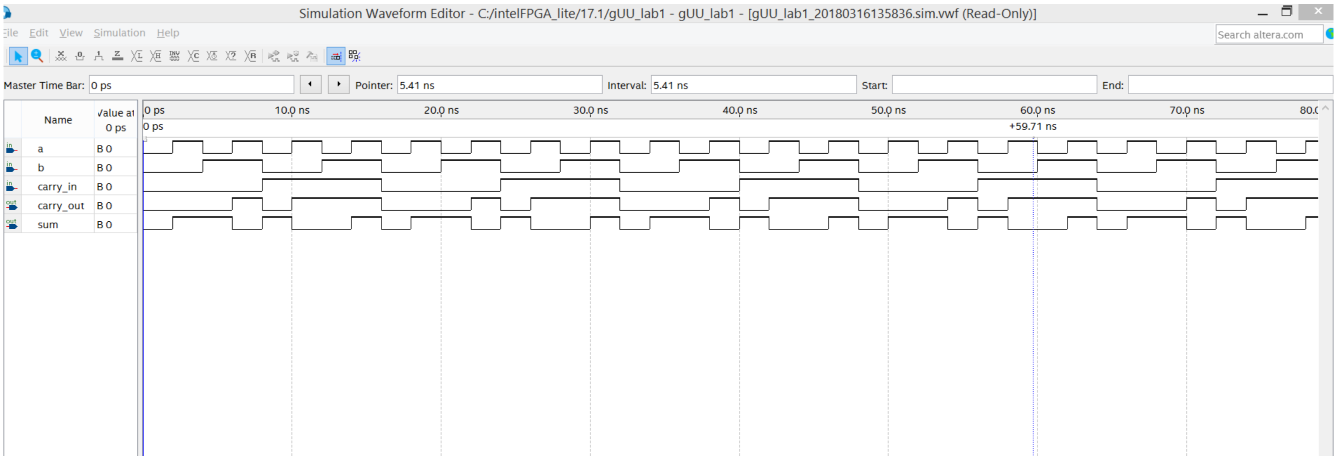
FUNCTIONALITY

The function of the 1-bit adder is to add *a* and *b,* while also considering a *carry-in* which might exist (and thus have a value of 1) or might not exist (having a value of 0). The result of this addition is the *sum*, and a *carry-out.* The minterm (sum of products) and XOR circuit representing the functionality of the adder is the following:

$$carryout = \overline{(a} \cdot b \cdot carryin) + (a \cdot b \cdot \overline{carryin}) + (a \cdot b \cdot carryin) + (a \cdot \overline{b} \cdot carryin)$$

$$sum = a \oplus b \oplus carryin$$

SIMULATION OF SUB-CIRCUIT



Simulation Waveform Editor - C:/intelFPGA_lite/17.1/gUU_lab1 - gUU_lab1 - [gUU_lab1_20180316135836.sim.vwf (Read-Only)]

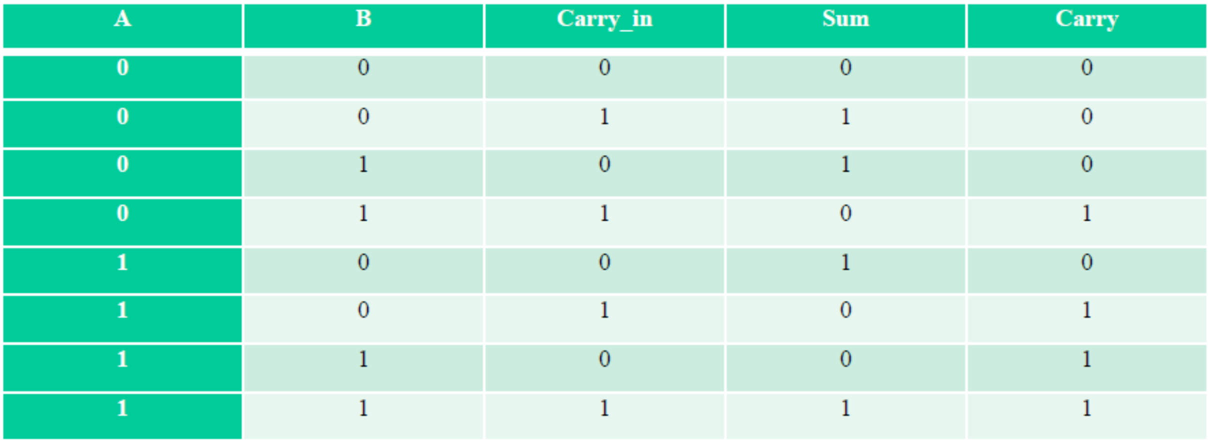
Master Time Bar: 0 ps
Pointer: 5.41 ns
Interval: 5.41 ns
Start:
End:

Name	Value at 0 ps
a	B 0
b	B 0
carry_in	B 0
carry_out	B 0
sum	B 0

FIGURE 2 : 1-bit Adder Sub-Circuit Simulation

ANALYSIS AND RESULTS

To analyze the validity of our circuit, we compared the waveform simulation to the 1-bit adder truth table provided in the laboratory instructions. All 8 possible *a/b/carry-in* combinations provided us with the correct outputs in the simulation and thus proved that our circuit was adequate.



A	B	Carry_in	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

TABLE 2 : 1-bit Adder Truth Table (Clark)

DISCUSSION

Our minterm expression was implemented using an AND-OR-XOR circuit which provided us with an adequate result. However, all AND-OR circuits can be implemented as NAND-NAND circuits which are more cost-effective in practice since they are implemented with simpler electronic circuits (Brown & Vranesic). This is a limitation in our circuit.

PART 3 : 1-bit Subtractor

FUNCTIONALITY

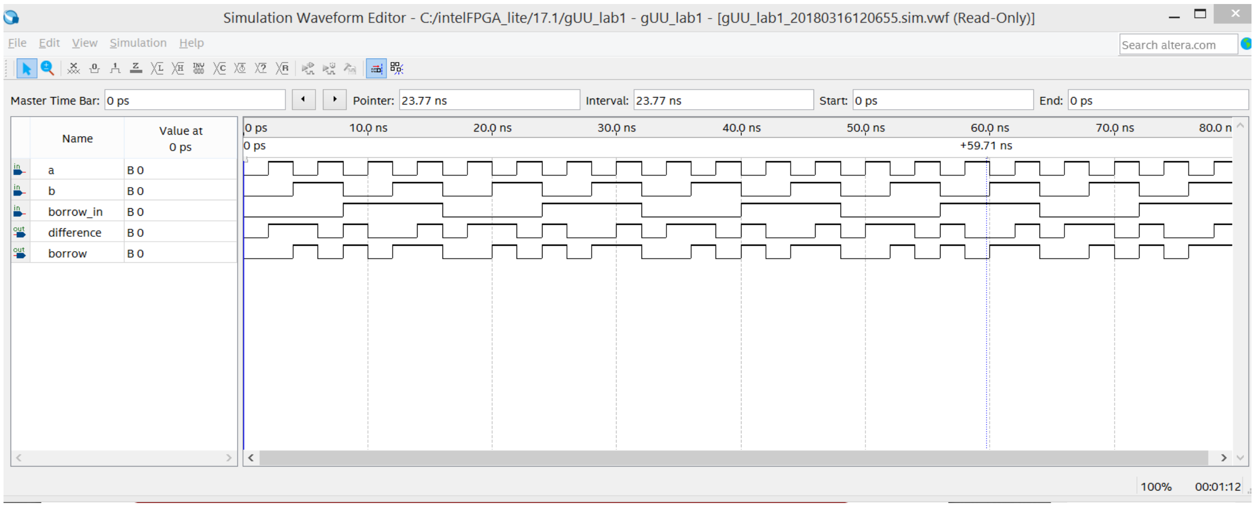
The function of the 1-bit subtractor is to subtract *b* from *a*, while also considering a *borrow-in*. The sum of b and borrow-in is subtracted from a. The subtraction results in a *difference* and a *borrow.* The minterm (sum of products) and XOR circuit representing the functionality of the subtractor is the following:

$$borrow = \overline{(a} \cdot\ borrowin) + (\overline{a} \cdot\ b) + (b \cdot\ borrowin)$$

$$diff = a\ \oplus b\ \oplus borrowin$$

-- Alexandra Zajda (260807517)
-- Nicolas Barreyro (260730549)
-- Gorkem Yalcinoz (260710053)
-- GROUP 12

SIMULATION OF SUB-CIRCUIT



Simulation Waveform Editor - C:/intelFPGA_lite/17.1/gUU_lab1 - gUU_lab1 - [gUU_lab1_20180316120655.sim.vwf (Read-Only)]

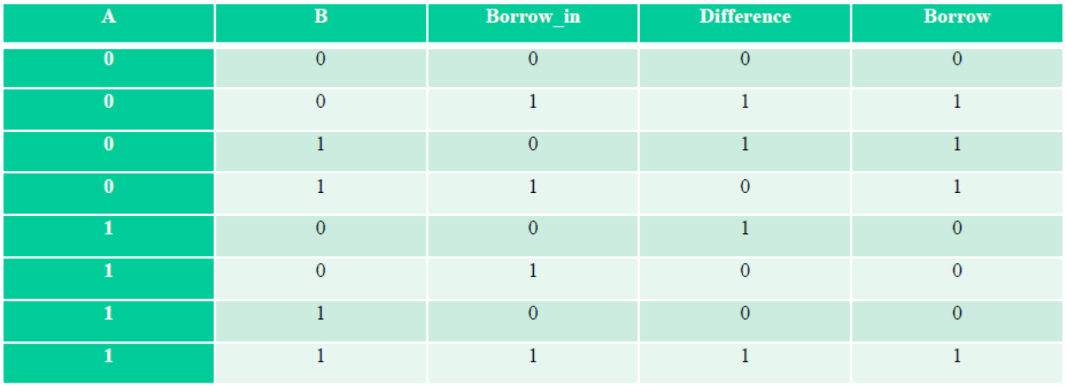
Master Time Bar: 0 ps Pointer: 23.77 ns Interval: 23.77 ns Start: 0 ps End: 0 ps

Name	Value at 0 ps
a	B 0
b	B 0
borrow_in	B 0
difference	B 0
borrow	B 0

FIGURE 3: 1-Bit Subtractor Sub-Circuit Simulation

ANALYSIS AND RESULTS

To analyze the validity of our circuit, we compared the waveform simulation to the 1-bit subtractor truth table provided in the laboratory instructions. All 8 possible *a/b/borrow-in* combinations provided us with the correct outputs in the simulation and thus proved that our circuit was adequate.



A	B	Borrow_in	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

TABLE 3 : 1-bit Subtractor Truth Table (Clark)

-- Alexandra Zajda (260807517)
-- Nicolas Barreyro (260730549)
-- Gorkem Yalcinoz (260710053)
-- GROUP 12

DISCUSSION

Similarly to the adder sub-circuit, our minterm expression was implemented using an AND-OR circuit which provided us with an adequate result. However, all AND-OR circuits can be implemented as NAND-NAND circuits which are more cost-effective in in practice since they are implemented with simpler electronic circuits (Brown & Vranesic). This is a limitation in our circuit. Note that the XOR difference gate could also be implemented as a combination of NAND gates.

-- Alexandra Zajda (260807517)
-- Nicolas Barreyro (260730549)
-- Gorkem Yalcinoz (260710053)
-- GROUP 12

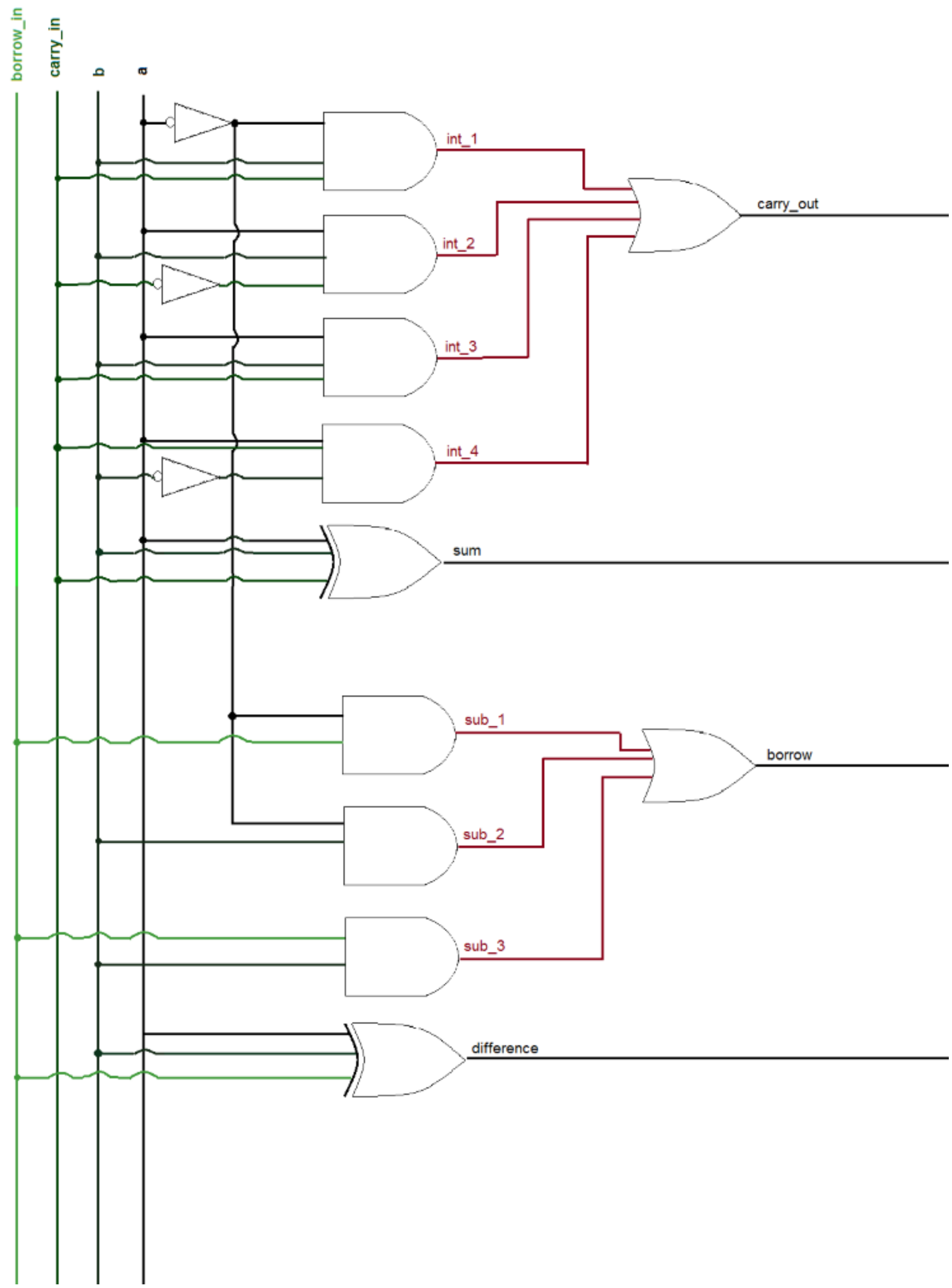
References

Clark, J. *“Lab#1 – Using the Altera Quartus Software”, ECSE323,* 2018*.*

Brown S. & Vranesic, Z. (2009). Fundamentals of Digital Logic with VHDL Design. *McGraw Hill,* Chapter 3*.*

-- Alexandra Zajda (260807517)
-- Nicolas Barreyro (260730549)
-- Gorkem Yalcinoz (260710053)
-- GROUP 12

ANNEXE A (GATE-LEVEL g12_1_bit_Adder_Subtractor CIRCUIT)


borrow_in
carry_in
b
a
int_1
int_2
int_3
int_4
carry_out
sum
sub_1
sub_2
sub_3
borrow
difference