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

ECSE-222 Digital Logic
LAB REPORT #3
Lab-3 Sequential Circuit Design: full_adder_substractor

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

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

INTRODUCTION

The goal of this lab was to design an 8-bit adder/subtractor calculator that could be mapped to pins, switches and LED's on the FPGA board to be tested. The circuit was designed using registers and allows the user to input 2 8-bit numbers to be added or subtracted to his will. For this purpose, the circuit contains a *clock*, *select_register*, *enable* and *reset* lines.

DESIGN (I/O)

The circuit has as **inputs** :

- *clock* input bit (actions are only processed during rising edge)
- *reset* input bit (clears internal 32-bit register)
- *enable* input bit (allows the user to store data in a register)
- *register_select* (*wr_reg*) input bit (1 to store data in R1, 0 to store data in R0)
- *add_mode* input bit (1 to add data in R0 and R1, 0 to subtract data in R1 from R0)
- 8-bit *user_input* vector which forwards physical switch inputs to internal registers R1/R0

The circuit has as **outputs**:

- 8-bit *data_out* vector (containing sum/diff of R1 and R0)
- *bit_out* (containing carry-out/borrow-out from sum/diff of R1 and R0)

The circuit makes use of a 32-bit internal signal that represents 4 8-bit registers. The registers' indexation starts from rightmost bit. These registers are used as follows:

- R0 : stores an 8-bit number, inputted by the user (*user_input*).
- R1 : stores an 8-bit number, inputted by the user (*user_input*).
- R2 : stores the result of an addition/subtraction.
- R3 : LSB of R3 stores *bit_out.*

A mapping of the inputs/outputs (except clock line) to switches/buttons/LEDs the Altera board can be seen in Figure 1.

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

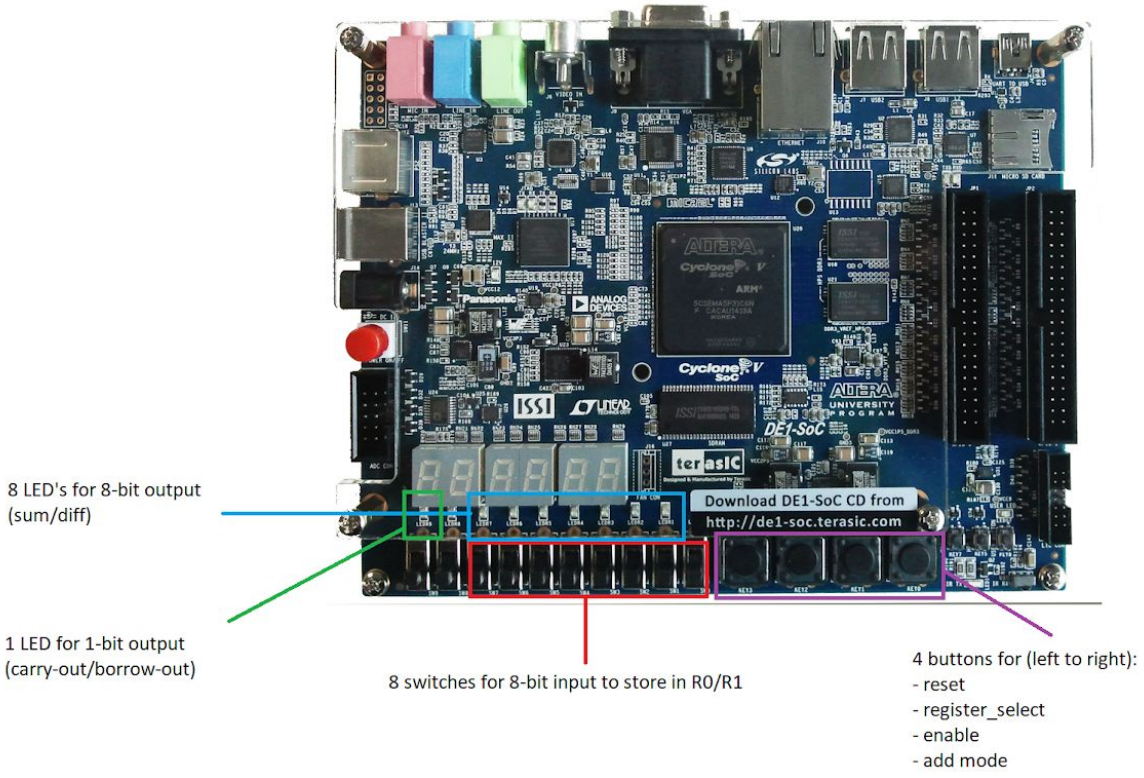

FIGURE 1 : Location of User-Input Pins/Buttons/Switches on the Altera Board

The circuit makes use of 2 components that were introduced in previous labs : the 8-bit subtractor, and the 8-bit adder. The components are mapped to internal signals, such as the registers, and perform addition/subtraction on the inputs stored in the registers R0/R1.

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

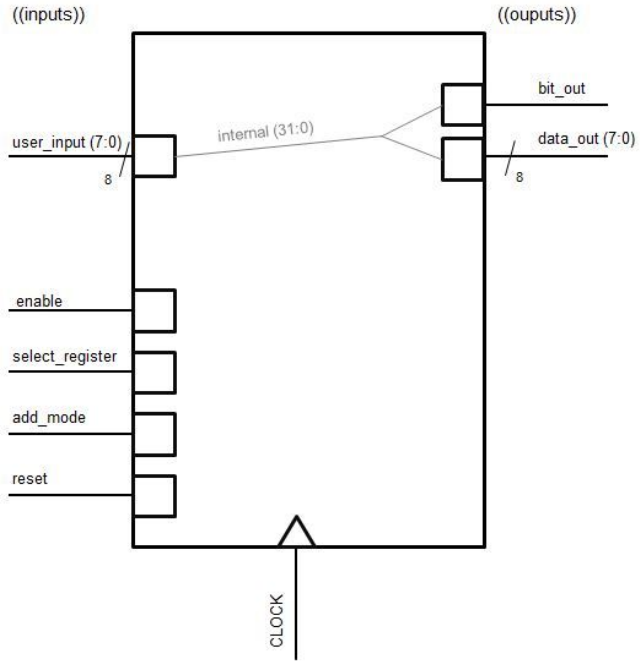

FIGURE 2 : Pin-out Diagram for lab3_g12

FUNCTIONALITY

The specific circuit uses user input to perform its various functions. This input is physically provided via an Altera Dec-Soc educational board. With the exception of the *clock* input, which was mapped to a predetermined 50kHz frequency clocking signal pin, all 4 other inputs (*reset*, *enable*, *write*, *register_select*) were user-inputs mapped to the 4 active-low buttons of the board. Thus, if the switches were pressed, the value of their signal was “0”.

Regardless of which operation the user wants to realise, to write data in either register, the enable button must be pressed simultaneously to the corresponding *select_register* button. To write to register R0, the *register_select* button must be pressed (it will have a value of “0”) and to write to register R1, the *register_select* button must not be pressed (it will have a value of “1”).

After data has been stored in both registers, an **addition** operation will be performed on the contents of R1 and R0. If the *add_mode* button has been pressed on the rising clock edge following the writing of data into both registers, a **subtraction** of R0 minus R1 will be performed.

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

The 8 rightmost switches are the 8 input bits (*user_input*), where the leftmost switch is the MSB. If the switch is up, then the input for that bit is 1; if its down, the input is 0. The 8 rightmost LEDs above the switches represent the contents of register *data_out:* the result of the addition or subtraction. The LED directly to the left of the MSB of the 8 input switches represents the *bit_out*, which is a *carry_out* (addition) or a *borrow_out* (subtraction).

A reset button was provided to clear the contents of all 4 registers.

The circuit is capable of showing negative numbers using two's complement notation.

SIMULATION PROCEDURE EXPLANATION

To test if the circuit was adequate, we first ran a timing waveform simulation of the circuit using Altera Quartus II software. Considering the amount of input variations and the fact that output was depending on timing (due to the clock signal) the waveform was used to evaluate specific carefully timed operations to see if the outputs *data_out* and *bit_out* were correct.

Atop of that, these same operations were reproduced on the Altera board once it was synched to the circuit.

Finally, a SignalTap II simulation was run, to evaluate real-time time delays that happened when our circuit was run on that specific piece of hardware. This was done to contrast the waveform simulation, which was a simulation that was not hardware-specific.

ANALYSIS

ADD_MODE = 1 (ADDITION)

To analyze the efficiency of the circuit, given the massive amount of possible inputs, we decided to settle on a specific set of inputs. For the sake of simplicity, we will analyze a specific example where our circuit adds the numbers 3 and 4 stored in R1 and R0 respectively. The output is seen in the *data_out* vector at the next clock rising edge.

As can be seen in Figure 3, inside the blue square, there are 3 clock rising edges, meaning 3 actions were performed. The reset input is set to 1 constantly (so there is no *reset* occuring), the *add_mode* input is set to 1 constantly (*data_out* and *bit_out* will represent the sum and carry-out respectively).

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

During the first clock rising edge, enable is transitioning from 0 to 1, the register_select input (*wr_reg*) is set to 1 and the input is set to 3. Thus, 3 is stored in R1. During the second clock rising edge, enable is transitioning from 0 to 1, the register_select input (*wr_reg*) is set to 0 and the input is set to 4. Thus, 4 is stored in R0. At the third rising edge, the *data_out* vector becomes 7 and *bit_out* is 0.

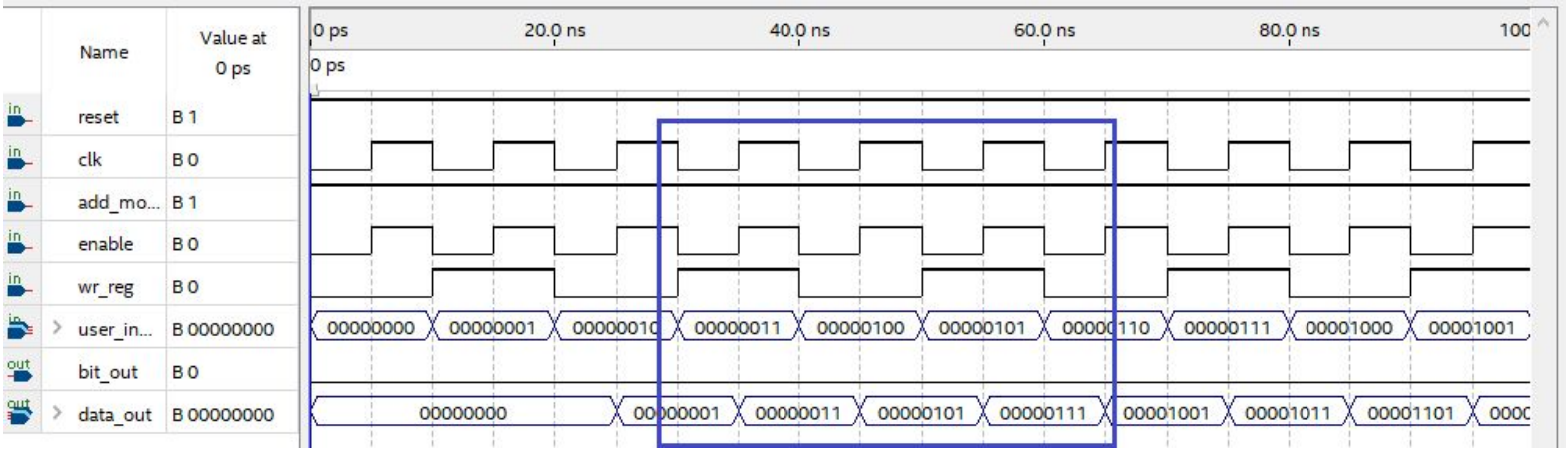

FIGURE 3 : Waveform showing Timing Simulation of circuit (R0 = 4 + R1 = 3 inside blue rectangle).

ADD_MODE = 0 (SUBSTRACTION)

For the sake of simplicity, we will analyze a specific example where our circuit substracts the numbers 4 and 3 stored in R0 and R1 respectively. The output is seen in the *data_out* vector at the next clock rising edge.

As can be seen in Figure 4, inside the blue square, there are 3 clock rising edges, meaning 3 actions were performed. The reset input is set to 1 constantly (so there is no *reset* occuring), the *add_mode* input is set to 0 constantly (*data_out* and *bit_out* will represent the difference and borrow-out respectively).

During the first clock rising edge, enable is transitioning from 0 to 1, the register_select input (*wr_reg*) is set to 1 and the input is set to 3. Thus, 3 is stored in R1. During the second clock rising edge, enable is transitioning from 0 to 1, the register_select input (*wr_reg*) is set to 0 and the input is set to 4. Thus, 4 is stored in R0. At the third rising edge, the *data_out* vector becomes 1 and *bit_out* is 0.

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

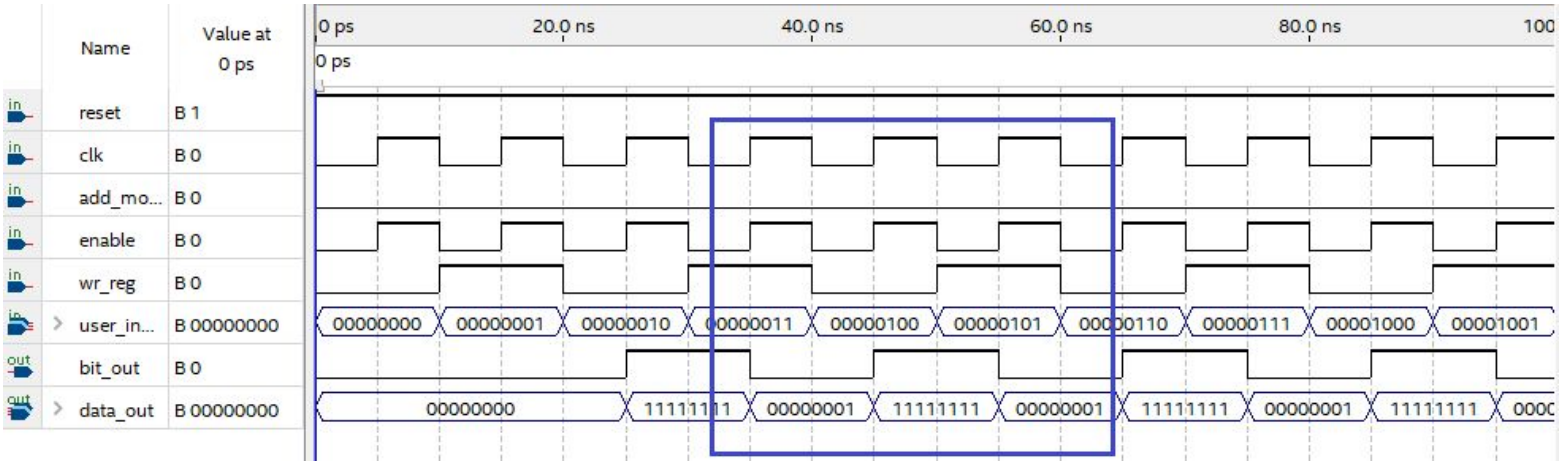

FIGURE 4 : Waveform showing Timing Simulation of circuit (R0 = 4 - R1 = 3 inside blue rectangle).

DISCUSSION

Following the analysis of the waveform simulations of the circuit, we concluded that the design and implementation of the circuit was accurate. The addition mode accurately produced the result 4 + 3 = 7 and the substraction mode produced the result 4 - 3 = 1. The design of our circuit (as seen in ANNEXE A) proved to be accurate.

To improve the performance of the circuit, the combinational inner workings of the adder and substractor could be optimized. In terms of in real-life performance, a higher performing board would result in a better performance of the circuit as well.

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

References

Clark, J. “Lab#3 - Sequential Circuit Design“, ECSE 323, 2018.

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

ANNEXE A (lab3_g12 -- SIMPLE CALCULATOR RTL Circuit Diagram)