Documentation

Hammer

Date: 24/06/2022

1 Team members

- Moataz Elbayaa “ moataz.elbayaa@stud.hshl.de “
- Ahmed Helmy “ ahmed.helmy@stud.hshl.de ’’
- Manoj Luitel “ manoj.luitel@stud.hshl.de “
- Saikot Das Joy “ saikot-das.joy@stud.hshl.de “

2 Introduction

Our Project is the extension of 4 bit Arithmetic Logic Unit(ALU) . Normally an ALU can do all the logic operations (Binary and, or,and,nor, xor,xnor,inverter etc) . But in our project, We have also included Binary Addition, Substraction, Multiplication and Division in 4 Bit ALU. We have used VHDL to design such a 4 bit ALU. We have used VHDL because it permits the behavior of the system to be verified and modeled in advance of the synthesis tools translation of the design into actual gates and wires (hardware).

FPGAs or Field Programmable Gate Arrays are semiconductor chips which are exactly as the name suggests – gate arrays that be configured in the field by the designer based on the needs. It consists of many configurable logic blocks which can be programmed based on what functions the designer requires it to perform.

3 Concept description

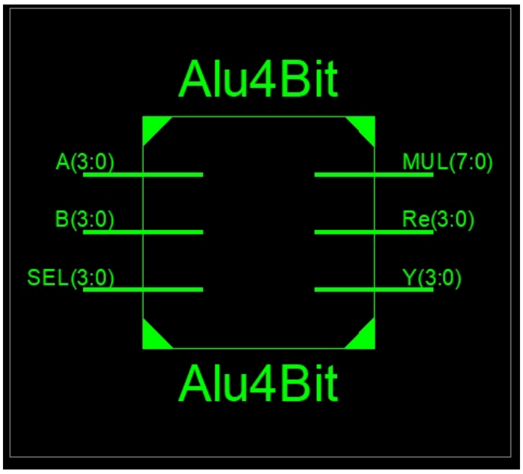


Fig: Block Diagram

Our designed Complex ALU, takes two 4 bit binary number as input and it also takes 4 bit SEL number as input. Based on the SEL number, it performs different operations. For example: it performs Binary “AND” operation for the SEL input “0000”, “OR” operation for SEL input “0001”, “NAND” operation for SEL input”0010”, “NOR” operation for SEL input”0011”, “XOR” operation for SEL input”0100”, “XNOR” operation for SEL input”0101”, “NOT” operation for SEL input”0110”. It also does Binary “ADDITION” operation for SEL input ”1001”, “Subtraction” operation for SEL input ”1010”, “MULTIPLICATION” operation for SEL input”0111” and “DIVISION” operation for SEL input ”1000”. All the output is found in 4 bit output “Y” except Multiplication. Multiplication output is found in output “MUL”. The reminder of division is found in Output “Re” .

4 Project/ Team Management

At first, we made a list of tasks that needs to be done. Then individual team members shared their interests in which tasks they want to do. Then individual team members took the responsibility to perform specific tasks. How did we divide the task is found below:

Saikot Das Joy: VHDL code and VHDL test bench and RTL Schematic creation using Xilinx
Manoj Luitel : Schemetic design in Eagle
Moataz Elbayaa : Schematic design in Eagle and after that PCB design
Ahmed Helmy : PCB design

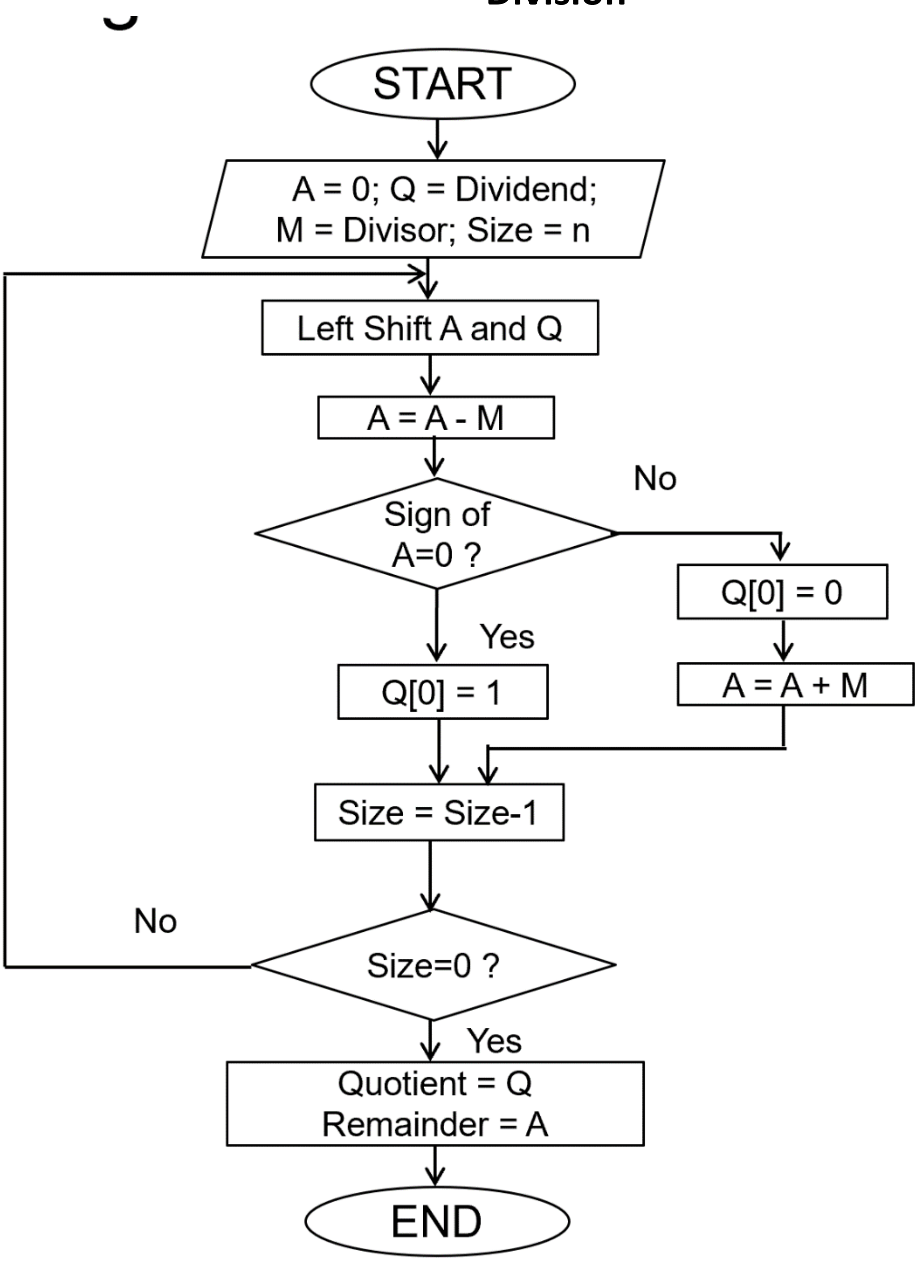
5 Technologies

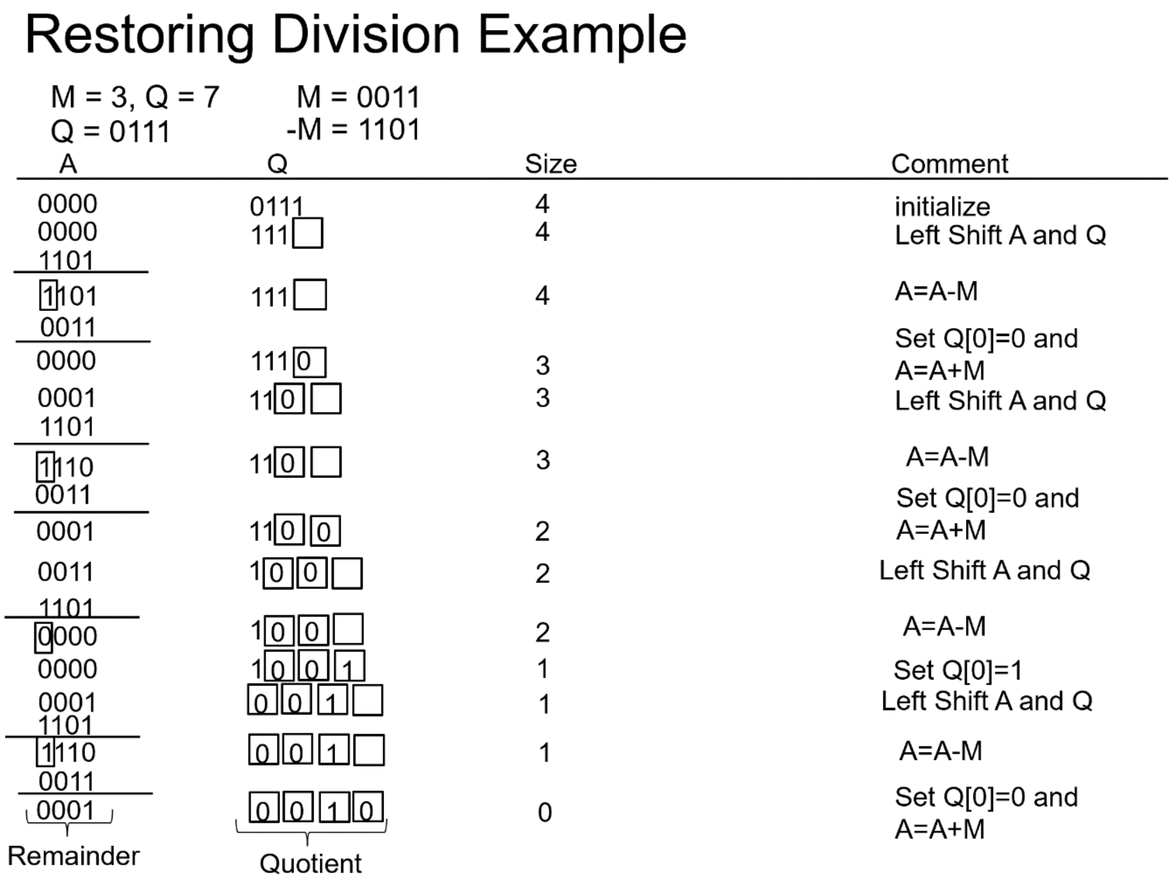
We will use VHDL code to design our project. For this purpose, we will use A software name “Modelsim” . We will also use another software “XILINX ISE 14.7” to get the rtl schemetic. We will also use a software named “Eagle” to draw the schematic and PCB circuit board.

6 VHDL Implementation

Here, in our project, The most complex part is 4-bit binary multiplication and division. So, This algorithm are described below:

Division





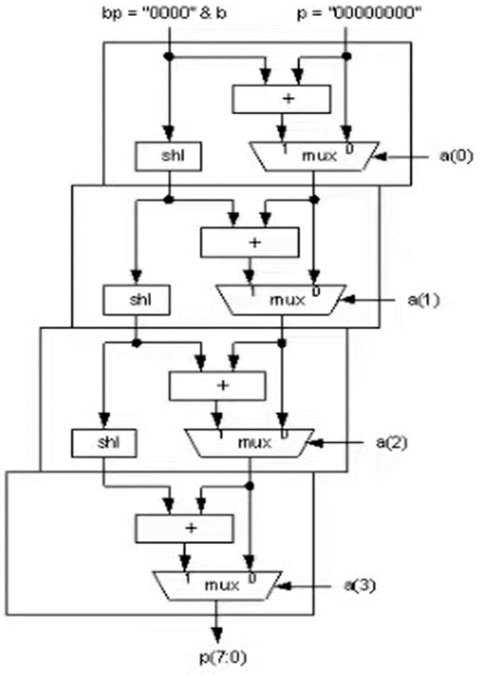
Restoring Division Example

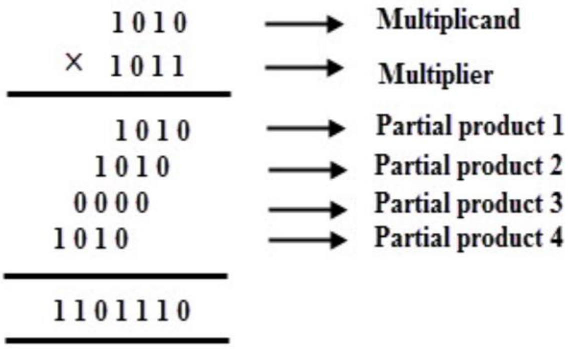
M = 3, Q = 7 M = 0011

Q = 0111 -M = 1101

A	Q	Size	Comment
0000	0111	4	initialize
0000 1101	111□	4	Left Shift A and Q
$\boxed{1}$101 0011	111□	4	A=A-M
0000	111$\boxed{0}$	3	Set Q[0]=0 and A=A+M
0001 1101	11$\boxed{0}$□	3	Left Shift A and Q
$\boxed{1}$110 0011	11$\boxed{0}$□	3	A=A-M
0001	11$\boxed{0}\boxed{0}$	2	Set Q[0]=0 and A=A+M
0011 1101	1$\boxed{0}\boxed{0}$□	2	Left Shift A and Q
$\boxed{0}$000	1$\boxed{0}\boxed{0}$□	2	A=A-M
0000	1$\boxed{0}\boxed{0}\boxed{1}$	1	Set Q[0]=1
0001 1101	$\boxed{0}\boxed{0}\boxed{1}$□	1	Left Shift A and Q
$\boxed{1}$110 0011	$\boxed{0}\boxed{0}\boxed{1}$□	1	A=A-M
$\underbrace{0001}_{\text{Remainder}}$	$\underbrace{\boxed{0}\boxed{0}\boxed{1}\boxed{0}}_{\text{Quotient}}$	0	Set Q[0]=0 and A=A+M

Multiplicaion:

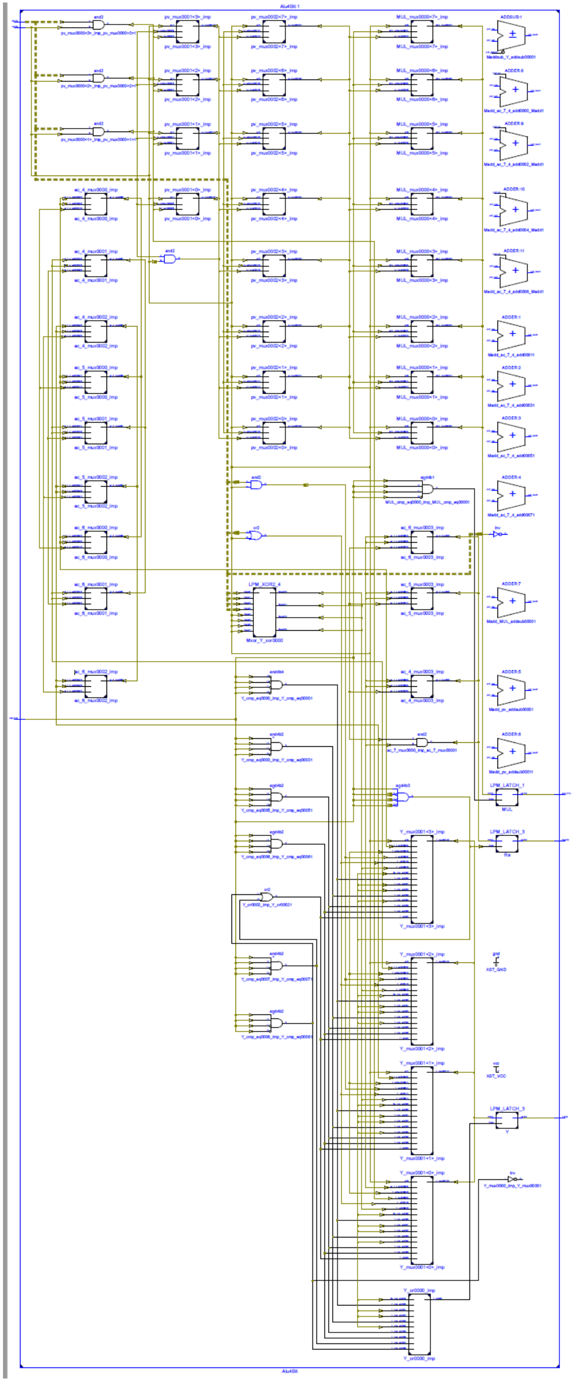




$$\begin{array}{r l}
1010 & \longrightarrow \text{Multiplicand} \\
\times\ 1011 & \longrightarrow \text{Multiplier} \\
\hline
1010 & \longrightarrow \text{Partial product 1} \\
1010 & \longrightarrow \text{Partial product 2} \\
0000 & \longrightarrow \text{Partial product 3} \\
1010 & \longrightarrow \text{Partial product 4} \\
\hline
1101110 & \\
\hline
\end{array}$$

7 RTL Schemetic

RTL schemetic generated using Xilinx is given below:



It is quite impossible to view this as an image file. You can find this diagram as a pdf file with High resolution Github.

8 Source Code

The VHDL code and test code is given below: (in GIthub Also)

Hammer

VHDL: (Source Code)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
use IEEE.STD_LOGIC_arith.ALL;


entity ALULULU is                                  ---- entity declaration
   Port ( A                 : in  STD_LOGIC_VECTOR (3 downto 0);   ---- First Number Input
        B                   : in  STD_LOGIC_VECTOR (3 downto 0);     ------- 2nd number input
        Y                   : out  STD_LOGIC_VECTOR (3 downto 0);   --- Output for all operation
except multiplication
            MUL             : out  STD_LOGIC_VECTOR (7 downto 0);   ---- output for multiplication
        Re                  : out  STD_LOGIC_VECTOR (3 downto 0);   --- Reminder after division
operation
        SEL                 : in  STD_LOGIC_VECTOR (3 downto 0));   ---- selection input to do
specific operation
end ALULULU;

architecture Behavioral of ALULULU is

begin
ALULULU_Proc : Process(A,B,SEL)
          variable ac : STD_LOGIC_VECTOR(7 downto 0);    --- variable declation for division
operation
          variable Mbar : STD_LOGIC_VECTOR(3 downto 0);   ---- variable declation for Division
operation
          variable pv,bp : STD_LOGIC_VECTOR(7 downto 0);  ---- variable declation for
Multiplication operation
Begin
Case SEL is
When "0000" =>
          Y <= A and B;
When "0001" =>
          Y <= A or B;
When "0010" =>
          Y <= A nand B;
When "0011" =>
          Y <= A nor B;
When "0100" =>
          Y <= A xor B;
When "0101" =>
          Y <= A xnor B;
When "0110" =>
          Y <= not A;
```

```
When "0111" =>                    ------ Multiplication

                pv :="00000000";
                bp := "0000" & B;
        for i in 0 to 3 loop
        if a(i) = '1' then
                pv := pv + bp;
        end if;
                bp :=  bp(6 downto 0) & '0';
        end loop;
                MUL <= pv;
When "1000" =>                        ------ Division
        Mbar := not B;
        ac := "0000" & A;
        for i in 1 to 4 loop
        ac(7 downto 0) := ac(6 downto 0) & '0';
        ac(7 downto 4) := ac(7 downto 4) + Mbar + "0001" ;
        if ac(7) = '1' then
        ac(0) := '0' ;
        ac( 7 downto 4) := ac(7 downto 4) + B;
        else
                ac(0) := '1';
        end if;
        end loop;
        Y <= ac(3 downto 0);
        Re <= ac(7 downto 4);

When "1001" =>                   ----- Addition

        Y <= (A + B);

When "1010" =>                   ----- Substraction
           Y <= (A - B);
when others =>
        NULL;
end case;
end process;
end Behavioral;
```

VHDL Test Bench :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
use IEEE.STD_LOGIC_arith.ALL;

entity ALULULU_TB is
```

```
end entity;

architecture CTB of ALULULU_TB is

component ALULULU
   port (A : in std_logic_vector(3 downto 0);
        B : in std_logic_vector(3 downto 0);
           SEL : in std_logic_vector(3 downto 0);
        Y : out std_logic_vector(3 downto 0);
           MUL : out std_logic_vector(7 downto 0);
           Re : Out std_logic_vector(3 downto 0));
end component;

   signal A_TB : std_logic_vector(3 downto 0);
   signal B_TB: std_logic_vector(3 downto 0);
   signal SEL_TB : std_logic_vector(3 downto 0);
   signal Y_TB : std_logic_vector (3 downto 0);
   signal MUL_TB : std_logic_vector (7 downto 0);
   signal Re_TB : std_logic_vector (3 downto 0);

 begin

DUT1: ALULULU port map (A => A_TB , B => B_TB , SEL => SEL_TB, Y => Y_TB,
                                        MUL => MUL_TB, Re => Re_TB);


A_TB <= "0000", "0010" after 100ns, "0100" after 200ns,  "0110" after 300ns,   "0011" after
400ns,  "0011" after 500ns, "0011" after 600ns,  "0011" after 700ns,  "1010" after 800ns,
"1010" after 900ns,"1010" after 1000ns, "1010" after 1100ns, "1010" after 1200ns;

B_TB<= "0000", "0001" after 100ns, "0010" after 200ns,  "0011" after 300ns,   "0100" after
400ns,  "0101" after 500ns, "0110" after 600ns,  "0111" after 700ns,  "1000" after 800ns,
"1001" after 900ns,"1010" after 1000ns, "1011" after 1100ns, "0110" after 1200ns;

SEL_TB<= "0000", "0001" after 100ns, "1000" after 200ns,  "0011" after 300ns,   "0100" after
400ns,  "0001" after 500ns, "0011" after 600ns,  "0010" after 700ns,  "0111" after 800ns,
"0010" after 900ns, "0011" after 1000ns, "0000" after 1100ns, "0001" after 1200ns;



end CTB;
```