



**Universitatea Tehnică “Gheorghe Asachi” din Iași**

**FACULTATEA DE AUTOMATICĂ ȘI  
CALCULATOARE**



# **ELECTRONICĂ DIGITALĂ**

## **Proiect**

**Tema: MODUL ALU-v5**

**Studenți:**

Rotariu Tudor

Arama Luigi

Tăbușcă Codrina

Grupa : 1210B

Coordonator:

Asist. Drd. Ionica Pletea

## Tema proiectului: **ALU – v5**

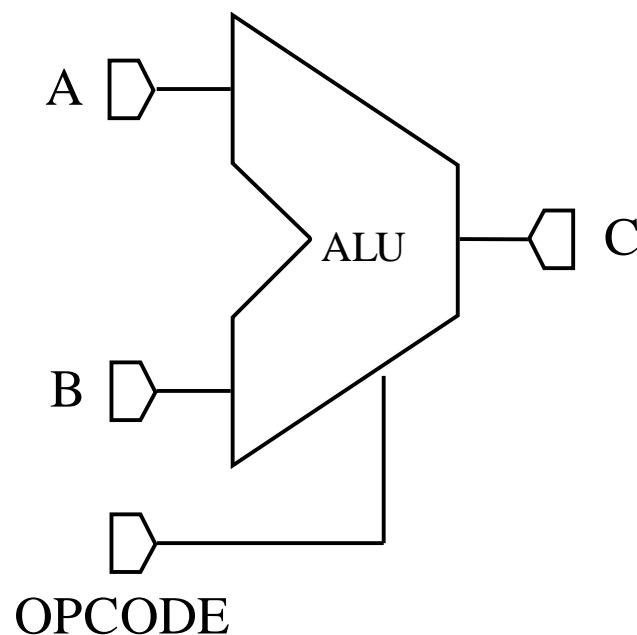
### 1. Specificațiile proiectului:

Să se implementeze în FPGA prin descriere în limbaj VHDL, utilizând programul VIVADO, modulul prezentat în figura 1 care este descris prin următoarele specificații:

- a) operanzii A și B au dimensiunea de 2 biți
- b) operațiile vor fi stabilite prin portul de intrare OPCODE
- c) lista de operații aritmetice: \*, /, +, -

Rezultatele vor fi asignate la portul C și vor fi vizualizate pe Displayul 7 segmente de pe placa de dezvoltare.

Descrierea va fi făcută în mod comportamental.



Fișierul bitstream rezultat în urma procesului de implementare va fi verificat utilizând placa de dezvoltare BASYS3

### 2. Modulul ALU\_v5

Proiectul nostru conține ca input 2 intrări de tip `std_logic_vector` pe 2 biți A și B. Apoi se va introduce intrarea `ALU_Sel` de tip `std_logic_vector` pe 2 biți ce va determina operația ce dorim a fi executată.

\*Pentru **ADUNARE** se va introduce "00"

\*Pentru **SCĂDERE** se va introduce "01"

\*Pentru **ÎNMULȚIRE** se va introduce "10"

\*Pentru **ÎMPĂRȚIRE** se va introduce "11"

În continuare se vor pune variabilele A și B în 2 variabile ajutoare astfel încât să se facă conversia numerelor pe 4 biți pentru a se putea realiza corect calculele dorite. Rezultatul va fi apoi salvat într-o variabilă ajutoare **ALU\_result**. Mai apoi se face afișarea rezultatului pe **placa BASYS 3** prin variabila **LED\_out** ce va fi vizualizat pe **Displayul 7 segmente** de pe placă. Pentru afișarea rezultatului pe Displayul 7 s-a făcut să se definească un case, unde fiecare valoare a **ALU\_result** e asociată unui set specific de semnale pentru aprinderea segmentelor corespunzătoare cifrei care dorim să fie afișată. În cazul nostru numărul maxim ce poate fi afișat este 9, fiind rezultatul operației de înmulțire  $3 \times 3$ , iar **LED\_out** va controla ce segmente vor fi aprinse pentru a reprezenta valoarea obținută în **ALU\_result**.

### 3. Metoda de implementare

În cadrul acestui proiect vor fi utilizate următoarele resurse:

- 1) Circuitul FPGA xc7a35tcbg236-1, din familia ARTIX 7 fiind produs de XILINX, acest circuit fiind utilizat prin intermediul plăcii de dezvoltare Basys3;
- 2) Tool-ul de sinteză VIVADO;
- 3) Limbajul de descriere VHDL (Very High Speed Integrated Circuit Hardware Description Language);

### 4. Descrierea (scurtă) a sistemului de dezvoltare BASYS 3

Placa Basys3 este o platformă de dezvoltare digitală completă, gata de utilizare, bazată pe ultima tehnologie Artix -7 Array Gate Gate Programmable (FPGA) de pe Xilinx.

Cu un FPGA de mare capacitate, un cost scăzut și o colecție de porturi USB, VGA și alte porturi, Basys3 acceptă proiecte care variază de la circuite combinaționale introductive până la circuite secvențiale complexe, cum ar fi procesoarele și controlerele încorporate.

Acesta include suficiente comutatoare, LED-uri și alte dispozitive I/O pentru a finaliza multe proiecte fără a fi nevoie de hardware suplimentar și suficient pentru a permite extinderea proiectului folosind Pmod-uri Digilent sau alte plăci și circuite personalizate.

Artix-7 FPGA este optimizat pentru logica de înaltă performanță, oferind mai multă capacitate , performanță mai mare și mai multe resurse decât modelele anterioare.

Basys 3 include multe instrumente noi si modalități de proiectare ce faciliteaza cele mai recente metode de proiectare, ruland mai repede, astfel fiind mai eficient.

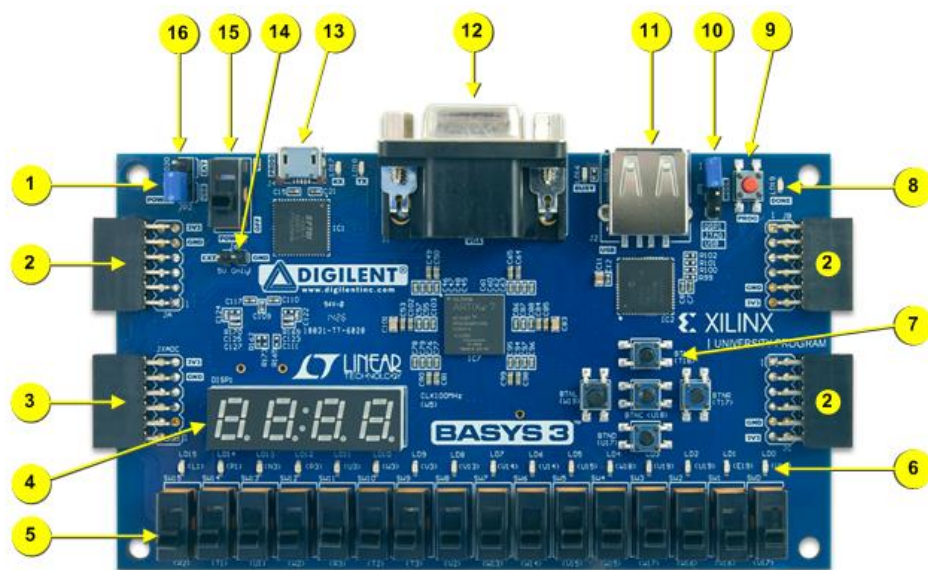
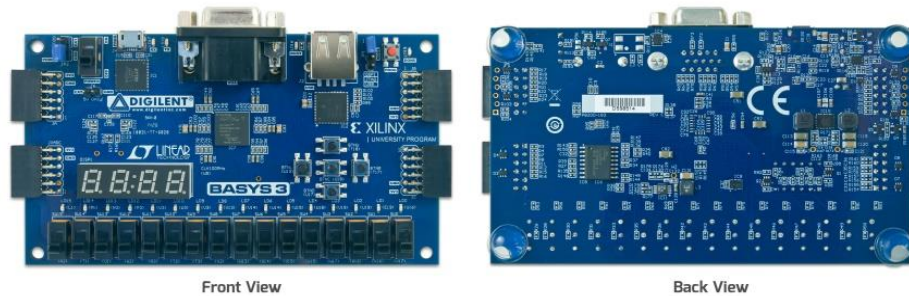


Figure 1. Basys3 board features

Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod connector(s)	10	Programming mode jumper
3	Analog signal Pmod connector (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper



## 5. Editarea fișierului VHDL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use ieee.NUMERIC_STD.all;
```

```
----- ALU 2-bit VHDL -----
```

```
entity ALU is
```

```
Port (
```

```
A, B : in STD_LOGIC_VECTOR(1 downto 0); -- 2 inputs 2-bit
```

```
ALU_Sel : in STD_LOGIC_VECTOR(1 downto 0); -- 1 input 4-bit for selecting function
```

```
LED_out : out STD_LOGIC_VECTOR (6 downto 0);
```

```
an : out std_logic_vector(2 downto 0)
```

```
);
```

```
end ALU;
```

```
architecture Behavioral of ALU is
```

```
signal ALU_Result : std_logic_vector (3 downto 0);
```

```
signal aa : std_logic_vector (3 downto 0);
```

```
signal bb : std_logic_vector (3 downto 0);
```

```
begin
```

```
an(0)<= '1';
```

```
an(1)<= '1';
```

```
an(2)<= '1';
```

```
process(A,B,ALU_Sel)
```

```

begin
aa <= ("00") & (A);
bb <= ("00") & (B);
case(ALU_Sel) is
when "00" => -- Addition
    ALU_Result <= (aa + bb) ;
when "01" => -- Subtraction
    ALU_Result <= (aa - bb) ;
when "10" => -- Multiplication
    ALU_Result <= (std_logic_vector(to_unsigned((to_integer(unsigned(aa)) *
to_integer(unsigned(bb))),4))) ;
when "11" => -- Division
    ALU_Result <= (std_logic_vector(to_unsigned(to_integer(unsigned(aa)) /
to_integer(unsigned(bb)),4))) ;

end case;
end process;
-- ALU out
process(ALU_Result)
begin
    case ALU_Result is
    when "0000" => LED_out <= "1000000"; -- "0"
    when "0001" => LED_out <= "1111001"; -- "1"
    when "0010" => LED_out <= "0100100"; -- "2"
    when "0011" => LED_out <= "0110000"; -- "3"
    when "0100" => LED_out <= "0011001"; -- "4"
    when "0101" => LED_out <= "0010010"; -- "5"
    when "0110" => LED_out <= "0000010"; -- "6"
    when "0111" => LED_out <= "1111000"; -- "7"
    when "1000" => LED_out <= "0000000"; -- "8"
    when "1001" => LED_out <= "0010000"; -- "9"
    when others => LED_out <= "1111111";

end case;
end process;
end Behavioral;

```

---

## 6. Editarea fișierului de constrângeri

```

## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins

```

## - rename the used ports (in each line, after get\_ports) according to the top level signal names in the project

## Clock signal

```
#set_property PACKAGE_PIN W5 [get_ports clk]
    #set_property IOSTANDARD LVCMOS33 [get_ports clk]
    #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
```

## Switches

```
set_property PACKAGE_PIN V17 [get_ports {A[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {A[0]}]
set_property PACKAGE_PIN V16 [get_ports {A[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {A[1]}]
set_property PACKAGE_PIN W16 [get_ports {ALU_Sel[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {ALU_Sel[0]}]
set_property PACKAGE_PIN W17 [get_ports {ALU_Sel[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {ALU_Sel[1]}]
set_property PACKAGE_PIN W15 [get_ports {B[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {B[0]}]
set_property PACKAGE_PIN V15 [get_ports {B[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {B[1]}]
#set_property PACKAGE_PIN W14 [get_ports {B[0]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {B[0]}]
#set_property PACKAGE_PIN W13 [get_ports {B[1]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {B[1]}]
#set_property PACKAGE_PIN V2 [get_ports {B[2]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {B[2]}]
#set_property PACKAGE_PIN T3 [get_ports {B[3]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {B[3]}]
#set_property PACKAGE_PIN T2 [get_ports {C[0]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {C[0]}]
#set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
#set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
#set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
#set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
#set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]
```

## LEDs

```
#set_property PACKAGE_PIN U16 [get_ports {ALU_Out[0]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {ALU_Out[0]}]
```

```

#set_property PACKAGE_PIN E19 [get_ports {ALU_Out[1]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {ALU_Out[1]}]
#set_property PACKAGE_PIN U19 [get_ports {ALU_Out[2]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {ALU_Out[2]}]
#set_property PACKAGE_PIN V19 [get_ports {ALU_Out[3]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {ALU_Out[3]}]
#set_property PACKAGE_PIN W18 [get_ports {led[4]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
#set_property PACKAGE_PIN U15 [get_ports {led[5]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
#set_property PACKAGE_PIN U14 [get_ports {led[6]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
#set_property PACKAGE_PIN V14 [get_ports {led[7]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
#set_property PACKAGE_PIN V13 [get_ports {led[8]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
#set_property PACKAGE_PIN V3 [get_ports {led[9]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
#set_property PACKAGE_PIN W3 [get_ports {led[10]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
#set_property PACKAGE_PIN U3 [get_ports {led[11]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
#set_property PACKAGE_PIN P3 [get_ports {led[12]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
#set_property PACKAGE_PIN N3 [get_ports {led[13]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
#set_property PACKAGE_PIN P1 [get_ports {led[14]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
#set_property PACKAGE_PIN L1 [get_ports {led[15]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]

```

##7 segment display

```

set_property PACKAGE_PIN W7 [get_ports {LED_out[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_out[0]}]
set_property PACKAGE_PIN W6 [get_ports {LED_out[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_out[1]}]
set_property PACKAGE_PIN U8 [get_ports {LED_out[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_out[2]}]
set_property PACKAGE_PIN V8 [get_ports {LED_out[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_out[3]}]

set_property PACKAGE_PIN U5 [get_ports {LED_out[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_out[4]}]
set_property PACKAGE_PIN V5 [get_ports {LED_out[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_out[5]}]

```



```
set_property PACKAGE_PIN U7 [get_ports {LED_out[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED_out[6]}]
```

```
#set_property PACKAGE_PIN V7 [get_ports dp]
    #set_property IOSTANDARD LVCMOS33 [get_ports dp]
#set_property PACKAGE_PIN U2 [get_ports {an[0]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
set_property PACKAGE_PIN U4 [get_ports {an[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
set_property PACKAGE_PIN V4 [get_ports {an[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
set_property PACKAGE_PIN W4 [get_ports {an[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
```

### ##Buttons

```
#set_property PACKAGE_PIN U18 [get_ports reset]
    #set_property IOSTANDARD LVCMOS33 [get_ports reset]
#set_property PACKAGE_PIN T18 [get_ports btnU]
    #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
#set_property PACKAGE_PIN W19 [get_ports btnL]
    #set_property IOSTANDARD LVCMOS33 [get_ports btnL]
#set_property PACKAGE_PIN T17 [get_ports btnR]
    #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
#set_property PACKAGE_PIN U17 [get_ports btnD]
    #set_property IOSTANDARD LVCMOS33 [get_ports btnD]
```

### ##Pmod Header JA

#### ##Sch name = JA1

```
#set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JA[0]}]
```

#### ##Sch name = JA2

```
#set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JA[1]}]
```

#### ##Sch name = JA3

```
#set_property PACKAGE_PIN J2 [get_ports {JA[2]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JA[2]}]
```

#### ##Sch name = JA4

```
#set_property PACKAGE_PIN G2 [get_ports {JA[3]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JA[3]}]
```

#### ##Sch name = JA7

```
#set_property PACKAGE_PIN H1 [get_ports {JA[4]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JA[4]}]
```

#### ##Sch name = JA8

```
#set_property PACKAGE_PIN K2 [get_ports {JA[5]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JA[5]]
##Sch name = JA9
#set_property PACKAGE_PIN H2 [get_ports {JA[6]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JA[6]]
##Sch name = JA10
#set_property PACKAGE_PIN G3 [get_ports {JA[7]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JA[7]]
```

```
##Pmod Header JB
##Sch name = JB1
#set_property PACKAGE_PIN A14 [get_ports {JB[0]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JB[0]]
##Sch name = JB2
#set_property PACKAGE_PIN A16 [get_ports {JB[1]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JB[1]]
##Sch name = JB3
#set_property PACKAGE_PIN B15 [get_ports {JB[2]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JB[2]]
##Sch name = JB4
#set_property PACKAGE_PIN B16 [get_ports {JB[3]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JB[3]]
##Sch name = JB7
#set_property PACKAGE_PIN A15 [get_ports {JB[4]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JB[4]]
##Sch name = JB8
#set_property PACKAGE_PIN A17 [get_ports {JB[5]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JB[5]]
##Sch name = JB9
#set_property PACKAGE_PIN C15 [get_ports {JB[6]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JB[6]]
##Sch name = JB10
#set_property PACKAGE_PIN C16 [get_ports {JB[7]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JB[7]]
```

```
##Pmod Header JC
##Sch name = JC1
#set_property PACKAGE_PIN K17 [get_ports {JC[0]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JC[0]]
##Sch name = JC2
#set_property PACKAGE_PIN M18 [get_ports {JC[1]]
    #set_property IOSTANDARD LVCMOS33 [get_ports {JC[1]]
```

```
##Sch name = JC3
#set_property PACKAGE_PIN N17 [get_ports {JC[2]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JC[2]}]
##Sch name = JC4
#set_property PACKAGE_PIN P18 [get_ports {JC[3]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JC[3]}]
##Sch name = JC7
#set_property PACKAGE_PIN L17 [get_ports {JC[4]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JC[4]}]
##Sch name = JC8
#set_property PACKAGE_PIN M19 [get_ports {JC[5]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JC[5]}]
##Sch name = JC9
#set_property PACKAGE_PIN P17 [get_ports {JC[6]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JC[6]}]
##Sch name = JC10
#set_property PACKAGE_PIN R18 [get_ports {JC[7]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JC[7]}]

##Pmod Header JXADC
##Sch name = XA1_P
#set_property PACKAGE_PIN J3 [get_ports {JXADC[0]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[0]}]
##Sch name = XA2_P
#set_property PACKAGE_PIN L3 [get_ports {JXADC[1]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[1]}]
##Sch name = XA3_P
#set_property PACKAGE_PIN M2 [get_ports {JXADC[2]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[2]}]
##Sch name = XA4_P
#set_property PACKAGE_PIN N2 [get_ports {JXADC[3]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[3]}]
##Sch name = XA1_N
#set_property PACKAGE_PIN K3 [get_ports {JXADC[4]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[4]}]
##Sch name = XA2_N
#set_property PACKAGE_PIN M3 [get_ports {JXADC[5]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[5]}]
##Sch name = XA3_N
#set_property PACKAGE_PIN M1 [get_ports {JXADC[6]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[6]}]
##Sch name = XA4_N
#set_property PACKAGE_PIN N1 [get_ports {JXADC[7]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[7]}]
```

### ##VGA Connector

```
#set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[0]}]
#set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[1]}]
#set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[2]}]
#set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[3]}]
#set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[0]}]
#set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[1]}]
#set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[2]}]
#set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[3]}]
#set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[0]}]
#set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[1]}]
#set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[2]}]
#set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
    #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[3]}]
#set_property PACKAGE_PIN P19 [get_ports Hsync]
    #set_property IOSTANDARD LVCMOS33 [get_ports Hsync]
#set_property PACKAGE_PIN R19 [get_ports Vsync]
    #set_property IOSTANDARD LVCMOS33 [get_ports Vsync]
```

### ##USB-RS232 Interface

```
#set_property PACKAGE_PIN B18 [get_ports RsRx]
    #set_property IOSTANDARD LVCMOS33 [get_ports RsRx]
#set_property PACKAGE_PIN A18 [get_ports RsTx]
    #set_property IOSTANDARD LVCMOS33 [get_ports RsTx]
```

### ##USB HID (PS/2)

```
#set_property PACKAGE_PIN C17 [get_ports PS2Clk]
    #set_property IOSTANDARD LVCMOS33 [get_ports PS2Clk]
    #set_property PULLUP true [get_ports PS2Clk]
#set_property PACKAGE_PIN B17 [get_ports PS2Data]
    #set_property IOSTANDARD LVCMOS33 [get_ports PS2Data]
```

```
#set_property PULLUP true [get_ports PS2Data]
```

```
##Quad SPI Flash
```

```
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using the  
##STARTUPE2 primitive.
```

```
#set_property PACKAGE_PIN D18 [get_ports {QspiDB[0]}]  
    #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[0]}]  
#set_property PACKAGE_PIN D19 [get_ports {QspiDB[1]}]  
    #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[1]}]  
#set_property PACKAGE_PIN G18 [get_ports {QspiDB[2]}]  
    #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[2]}]  
#set_property PACKAGE_PIN F18 [get_ports {QspiDB[3]}]  
    #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[3]}]  
#set_property PACKAGE_PIN K19 [get_ports QspiCSn]  
    #set_property IOSTANDARD LVCMOS33 [get_ports QspiCSn]
```

## 7. Descrierea pașilor de sinteză și testarea circuitului rezultat

1. S-a creat un nou program in Vivado
2. S-a implementat modulul “ALU-v5” printr-o implementare a codului
3. S-a editat fisierul de constrangeri
4. S-au efectuat simularile necesare verificarii functionalitatii codului
5. S-a conectat placa si s-a efectuat rulara codului pe placa
6. Au fost realizate mai multe operatii pe placa pentru a se testa corectitudinea programului implementat

## 8. Concluzii

În concluzie s-a implementat un modul **ALU-v5** prevăzut cu **2 intrări A si B**, ambele fiind de 2 biți. Programul realizeaza operațiile aritmetice +, -, \*, / iar operațiile de intrare vor fi stabilite prin portul de intrare ALU\_sel. Rezultatele sunt asignate la ALU\_result si vor fi vizualizate pe **Displayul 7 segmente** de pe placa BASYS 3.

## Bibliografie:

1. VHDL Reference Manual,  
<http://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf>
2. BASYS 3 Reference Manual, <https://reference.digilentinc.com/reference/programmable-logic/basys-3/reference-manual>