



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



**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

# **ELECTRONICĂ DIGITALĂ**

## **proiect**

**Tema: ALU - v3**

**Studenți:**

Arhip Constantin-Alexandru  
Hrițcu Marina-Dumitrița  
Budu Daniel

**Grupa:**  
1206B

**Coordonator:**  
Asist. Drd. Marius Obreja

**2021**

## 1. Specificațiile proiectului:

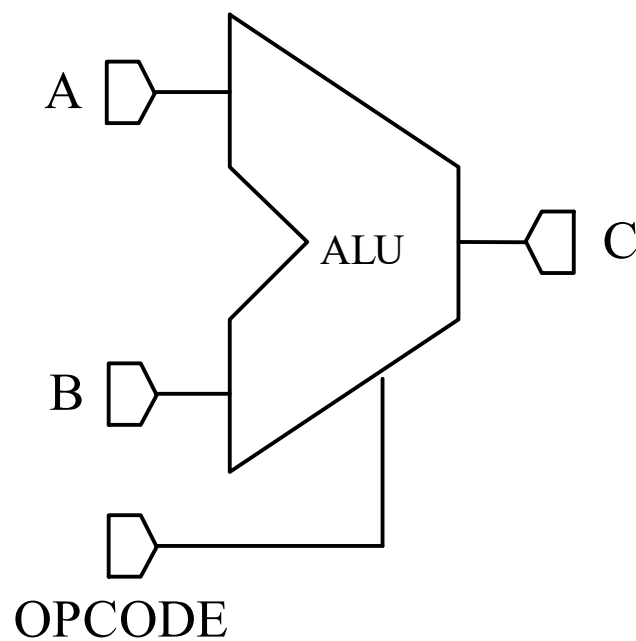
### ALU - v3

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) operandii A și B au dimensiunea de 8 biți
- b) operațiile vor fi stabilite prin portul de intrare OPCODE
- c) lista de operații: operații logice pe biți: OR, AND, NAND, XOR

Rezultatele vor fi asigurate la portul C și vor fi vizualizate prin LED-urile de pe placa de dezvoltare.

Descrierea va fi făcută în mod structural

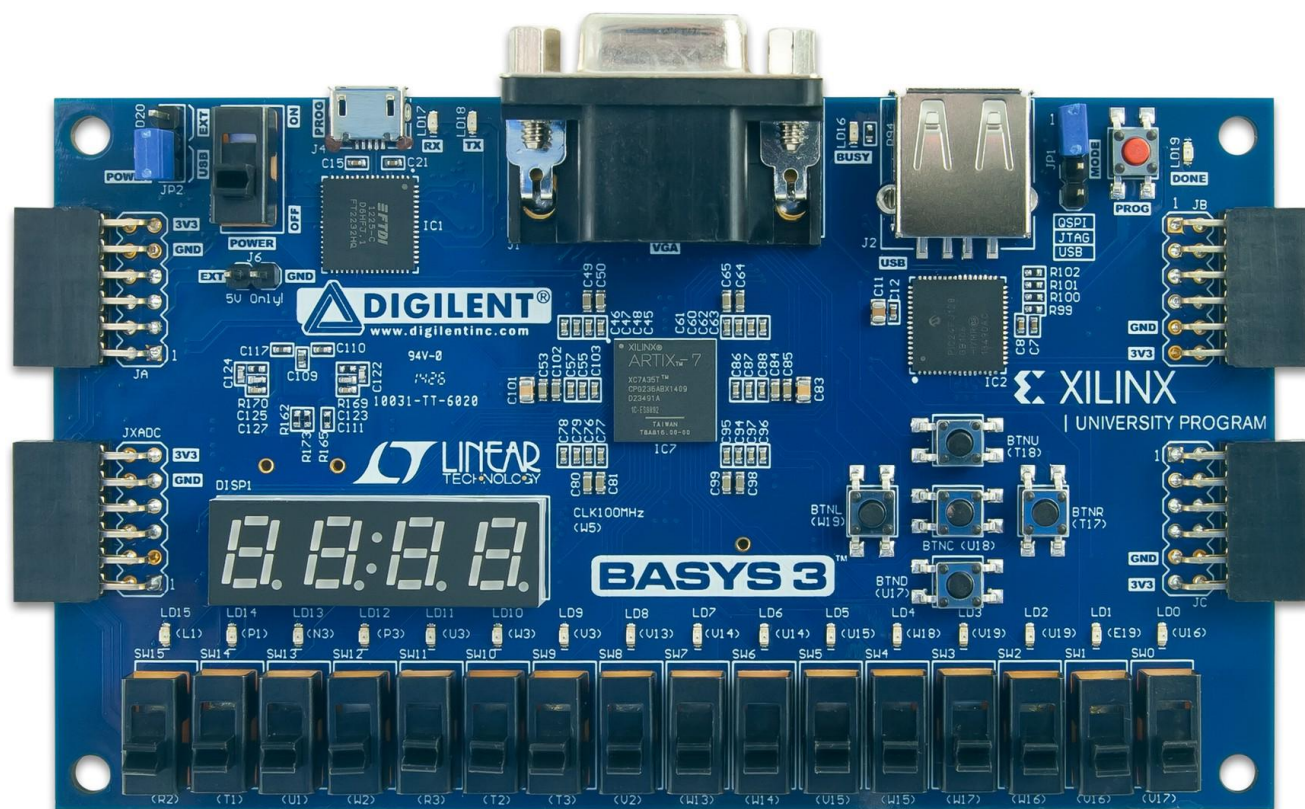


Fișierul bitstream rezultat în urma procesului de implementare va fi verificat utilizând placa de dezvoltare BASYS3.

## 2. Modulul ALU - v3

Modulul ALU - v3 are urmatoarele functionalitati:

1. Contine 2 numere pe 8 biti, care sunt introduce de utilizator cu ajutorul celor 16 switch-uri. (Primul numar poate fi introdus de la dreapta la stanga, incepand cu switch-ul 0 pana la 7 [V17, V16, W16, W17, W15, V15, W14, W13], urmand ca cel de al doilea numar sa fie introdus de la switch-ul 8 pana la 15 [U2, T3, T2, R3, W2, U1, T1, R2]).
2. Odata introduse cele doua numere pe 8 biti, pe langa butonul din centru (U18) al carui rol este de a confirma operatia dorita, avem implementate si celelalte butoane care-l inconjoara, astfel:
  - Butonul de deasupra (T18) va executa operatia “AND” intre cele doua numere pe 8 biti.
  - Butonul din stanga (W19) va executa operatia “OR” intre cele doua numere pe 8 biti.
  - Butonul de dedesubt (U17) va executa operatia “NAND” intre cele doua numere pe 8 biti.
  - Butonul din dreapta (T17) va executa operatia “XOR” intre cele doua numere pe 8 biti.
  - Butonul din mijloc (U18) va executa operatia de confirmare a uneia dintre operatiile alese.
3. Rezultatul oricaror din aceste operatii, este stocat intr-un registru “C”, care este pe 8 biti, urmand ca acesta sa fie afisat, de la dreapta la stanga, cu ajutorul ledurilor de deasupra switch-urile, respectiv (U16, E19, U19, V19, W18, U15, U14, V14).
4. Switch-urile și LED-urile sunt situate pe placă după cum este reprezentat mai jos in figura.



### **3. Metoda de implementare**

Pentru implementarea acestui modul s-au folosit programul de sinteza Vivado si limbajul VHDL. Implementarea proiectului a fost facuta printr-o descriere comportamentala. S-a proiectat entitatea Proiect\_ALUv3. Aceasta are mai multe “procese”:

1. Doua numere pe 8 biti (A,B).
2. Un if statement ce contine operatia “AND”.
3. Un if statement ce contine operatia “OR”.
4. Un if statement ce contine operatia “NAND”.
5. Un if statement ce contine operatia “XOR”.
6. Rezultatul, in functie de operatia dorita va fi stocat intr-o variabila temporara, ulterior aceasta fiind afisata prin intermediul ledurilor.

Toate operatiile mentionate anterior sunt deja implementate si definite in biblioteca “IEEE.STD\_LOGIC\_1164.ALL”. Fisierul bitstream creat de programul Vivado a fost testat cu ajutorul placii BASYS 3 Artix-7xc7a35tcpg236-1.

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

Placa de dezvoltare BASYS 3 este un circuit de dezvoltare complet si ready-to-use bazat pe ultimele Artix-7 Field Programmable Gate Array(FPGA) produse de Xilinx. Cu o mare capacitate de FPGA si cu o colectie de porturi USB, VGA si altele, placa de dezvoltare BASYS 3 permite proiectarea unor design-uri variate, atat circuite introductorii combinationale, cat si circuite secventiale complexe ca procesoarele si controllerele embedded.

## 5. Editarea fișierului VHDL

-----Proiect\_ALUv3.vhd(TOP MODULE)-----

Entitatea Proiect\_ALUv3:

```
1
2 library IEEE;
3 use IEEE.STD_LOGIC_1164.ALL;
4
5 entity Proiect_ALUv3 is
6 port(
7     ok : in STD_LOGIC;
8     A  : in STD_LOGIC_VECTOR (7 downto 0);
9     B  : in STD_LOGIC_VECTOR (7 downto 0);
10    Op_AND : in STD_LOGIC;
11    Op_OR  : in STD_LOGIC;
12    Op_NAND : in STD_LOGIC;
13    Op_XOR : in STD_LOGIC;
14    C      : out STD_LOGIC_VECTOR(7 downto 0)
15 );
16 end Proiect_ALUv3;
17
18 Architecture behavior of Proiect_ALUv3 is
19
20     signal tmp : std_logic_vector(7 downto 0);
21
22 begin
23
24     process(A,B,Op_AND,Op_OR,Op_NAND,Op_XOR,ok)
25
26     begin
```

If statement “AND”:

```
27
28     if rising_edge (ok) then
29         if Op_AND = '1' then
30             tmp <= A and B;
31         end if;
32     end if;
```

If statement “OR”:

```
33
34     if rising_edge (ok) then
35         if Op_OR = '1' then
36             tmp <= A or B;
37         end if;
38     end if;
39
```

If statement “NAND”:

```
39 |  
40 | if rising_edge (ok) then  
41 |     if Op_NAND = '1' then  
42 |         tmp <= not(A and B);  
43 |     end if;  
44 | end if;  
45 |
```

If statement “XOR”:

```
46 | if rising_edge (ok) then  
47 |     if Op_XOR = '1' then  
48 |  
49 |         tmp <= A xor B;  
50 |     end if;  
51 | end if;  
52 |  
53 |  
54 | end process;  
55 |  
56 | C <= tmp;  
57 |  
58 | end behavior;  
59 |
```

Pentru creșterea lizibilității codului, acesta a fost decupat din fișierele vhd vazute prin intermediul programului Vivado 2021.2.



## 6. Editarea fișierului de constrângeri

Switch-uri:

```
11  ## Switches
12  set_property PACKAGE_PIN V17 [get_ports {A[0]}]
13      set_property IOSTANDARD LVCMOS33 [get_ports {A[0]}]
14  set_property PACKAGE_PIN V16 [get_ports {A[1]}]
15      set_property IOSTANDARD LVCMOS33 [get_ports {A[1]}]
16  set_property PACKAGE_PIN W16 [get_ports {A[2]}]
17      set_property IOSTANDARD LVCMOS33 [get_ports {A[2]}]
18  set_property PACKAGE_PIN W17 [get_ports {A[3]}]
19      set_property IOSTANDARD LVCMOS33 [get_ports {A[3]}]
20  set_property PACKAGE_PIN W15 [get_ports {A[4]}]
21      set_property IOSTANDARD LVCMOS33 [get_ports {A[4]}]
22  set_property PACKAGE_PIN V15 [get_ports {A[5]}]
23      set_property IOSTANDARD LVCMOS33 [get_ports {A[5]}]
24  set_property PACKAGE_PIN W14 [get_ports {A[6]}]
25      set_property IOSTANDARD LVCMOS33 [get_ports {A[6]}]
26  set_property PACKAGE_PIN W13 [get_ports {A[7]}]
27      set_property IOSTANDARD LVCMOS33 [get_ports {A[7]}]
28
29  set_property PACKAGE_PIN V2 [get_ports {B[0]}]
30      set_property IOSTANDARD LVCMOS33 [get_ports {B[0]}]
31  set_property PACKAGE_PIN T3 [get_ports {B[1]}]
32      set_property IOSTANDARD LVCMOS33 [get_ports {B[1]}]
33  set_property PACKAGE_PIN T2 [get_ports {B[2]}]
34      set_property IOSTANDARD LVCMOS33 [get_ports {B[2]}]
35  set_property PACKAGE_PIN R3 [get_ports {B[3]}]
36      set_property IOSTANDARD LVCMOS33 [get_ports {B[3]}]
37  set_property PACKAGE_PIN W2 [get_ports {B[4]}]
38      set_property IOSTANDARD LVCMOS33 [get_ports {B[4]}]
39  set_property PACKAGE_PIN U1 [get_ports {B[5]}]
40      set_property IOSTANDARD LVCMOS33 [get_ports {B[5]}]
41  set_property PACKAGE_PIN T1 [get_ports {B[6]}]
42      set_property IOSTANDARD LVCMOS33 [get_ports {B[6]}]
43  set_property PACKAGE_PIN R2 [get_ports {B[7]}]
44      set_property IOSTANDARD LVCMOS33 [get_ports {B[7]}]
```

LED-uri:

```
47  ## LEDs
48
49  set_property PACKAGE_PIN U16 [get_ports {C[0]}]
50      set_property IOSTANDARD LVCMOS33 [get_ports {C[0]}]
51  set_property PACKAGE_PIN E19 [get_ports {C[1]}]
52      set_property IOSTANDARD LVCMOS33 [get_ports {C[1]}]
53  set_property PACKAGE_PIN U19 [get_ports {C[2]}]
54      set_property IOSTANDARD LVCMOS33 [get_ports {C[2]}]
55  set_property PACKAGE_PIN V19 [get_ports {C[3]}]
56      set_property IOSTANDARD LVCMOS33 [get_ports {C[3]}]
57  set_property PACKAGE_PIN W18 [get_ports {C[4]}]
58      set_property IOSTANDARD LVCMOS33 [get_ports {C[4]}]
59  set_property PACKAGE_PIN U15 [get_ports {C[5]}]
60      set_property IOSTANDARD LVCMOS33 [get_ports {C[5]}]
61  set_property PACKAGE_PIN U14 [get_ports {C[6]}]
62      set_property IOSTANDARD LVCMOS33 [get_ports {C[6]}]
63  set_property PACKAGE_PIN V14 [get_ports {C[7]}]
64      set_property IOSTANDARD LVCMOS33 [get_ports {C[7]}]
```

Butoane externe:

```
112  ##Buttons
113
114  set_property PACKAGE_PIN U18 [get_ports ok]
115      set_property IOSTANDARD LVCMOS33 [get_ports ok]
116      set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets ok]
117
118  set_property PACKAGE_PIN T18 [get_ports Op_AND]
119      set_property IOSTANDARD LVCMOS33 [get_ports Op_AND]
120
121  set_property PACKAGE_PIN W19 [get_ports Op_OR]
122      set_property IOSTANDARD LVCMOS33 [get_ports Op_OR]
123
124  set_property PACKAGE_PIN T17 [get_ports Op_XOR]
125      set_property IOSTANDARD LVCMOS33 [get_ports Op_XOR]
126
127  set_property PACKAGE_PIN U17 [get_ports Op_NAND]
128      set_property IOSTANDARD LVCMOS33 [get_ports Op_NAND]
129
```



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

1. S-a creat un proiect nou în programul Vivado
2. S-a implementat modulul "Proiect\_ALUv3" printr-o descriere comportamentală (având drept "proces" 4 if statement-uri cu operațiile pe biți OR,AND,NAND,XOR.)
3. S-a editat fișierul de constrângeri în vederea realizării legăturilor între switch-uri și intrări(A,B), cele 5 butoane care au rolul de a alege operația dorită cât și de a confirma selecția , led-urile de deasupra switch-urilor și ieșirea C.
4. S-a realizat analiza RTL(Register Transfer Level)
5. S-a sintetizat modulul(pentru se vedea design-ul sintetizat)
6. S-a lansat implementarea proiectului care a avut ca efect final generarea fișierului bitstream.
7. S-a programat placa de dezvoltare BASYS 3 cu fișierul bitstream și s-a testat funcționarea corespunzătoare a modulului implementat

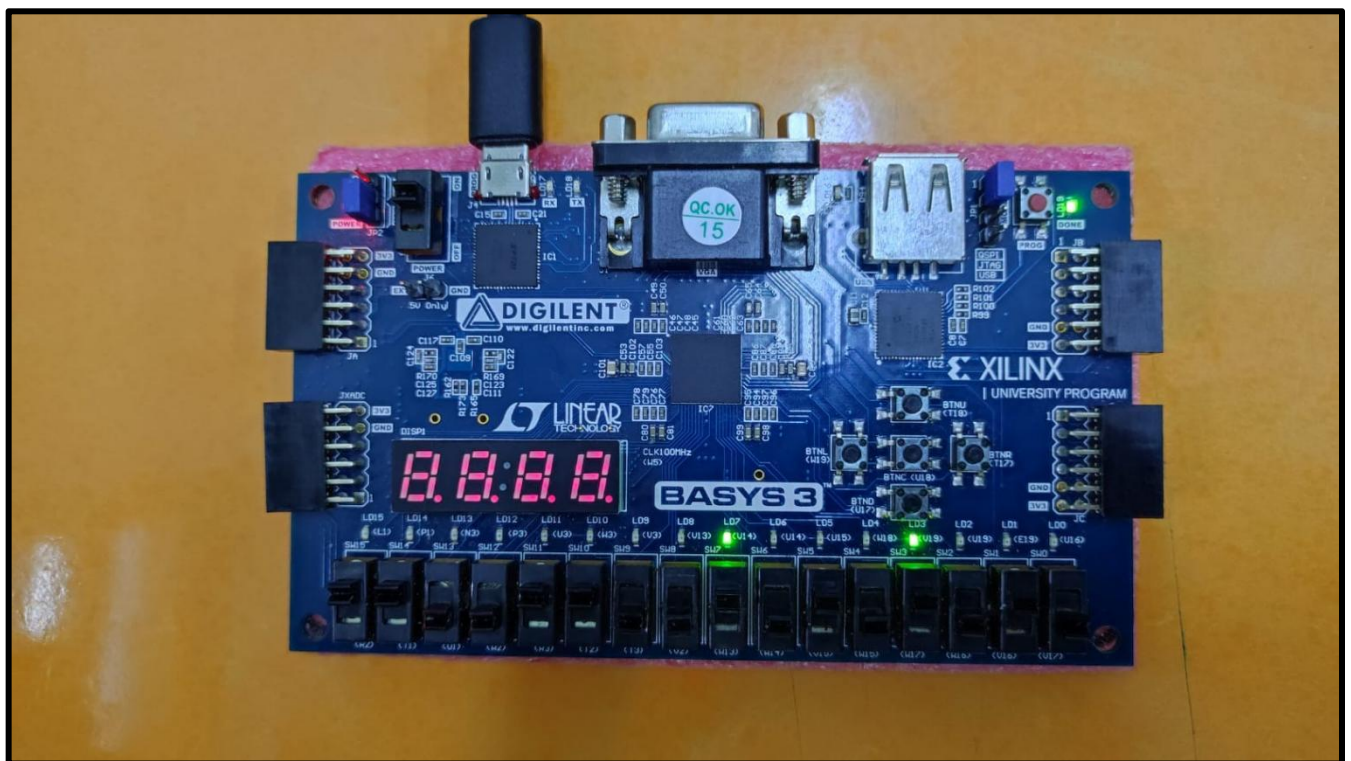
## 8. Fotografii cu funcționarea modulului

Primul număr:           01010101

Al doilea număr:       00110011

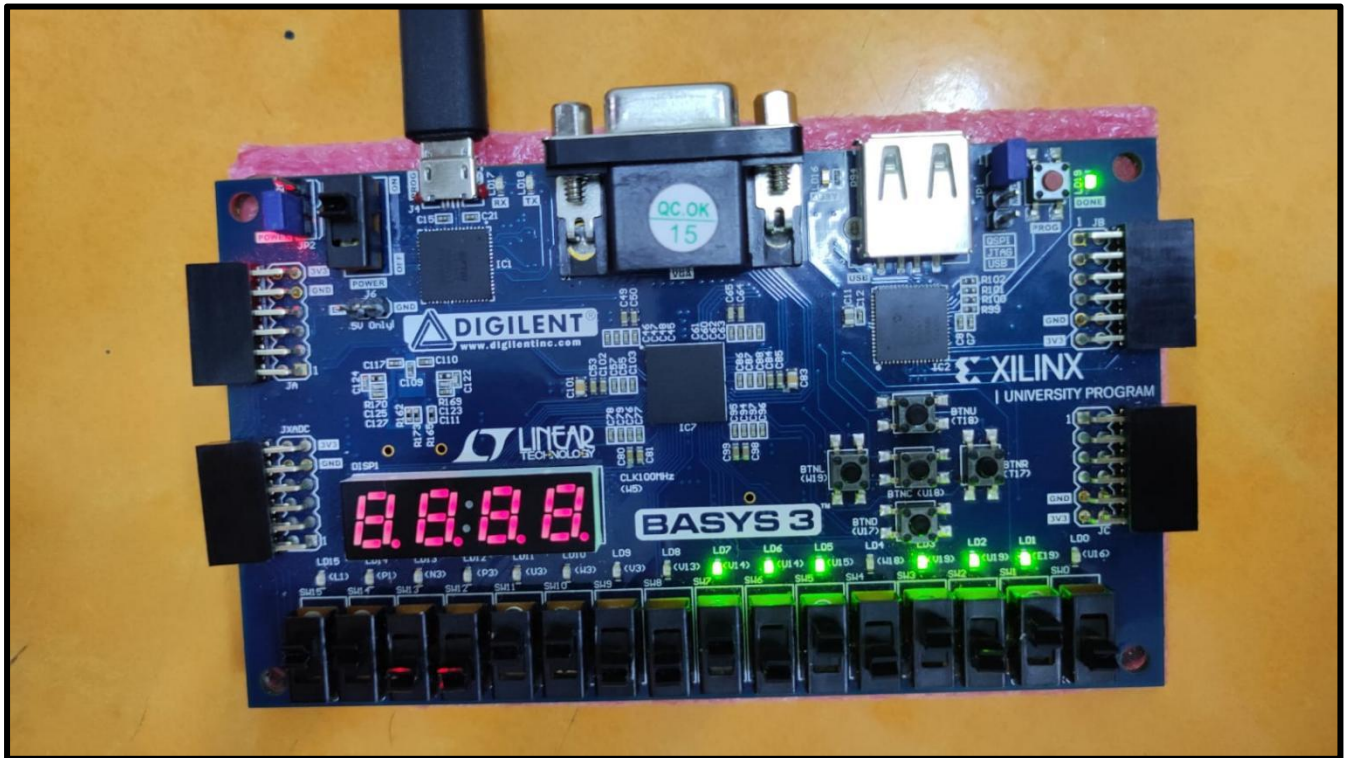
**Operația "AND" :**

Rezultat: 00010001



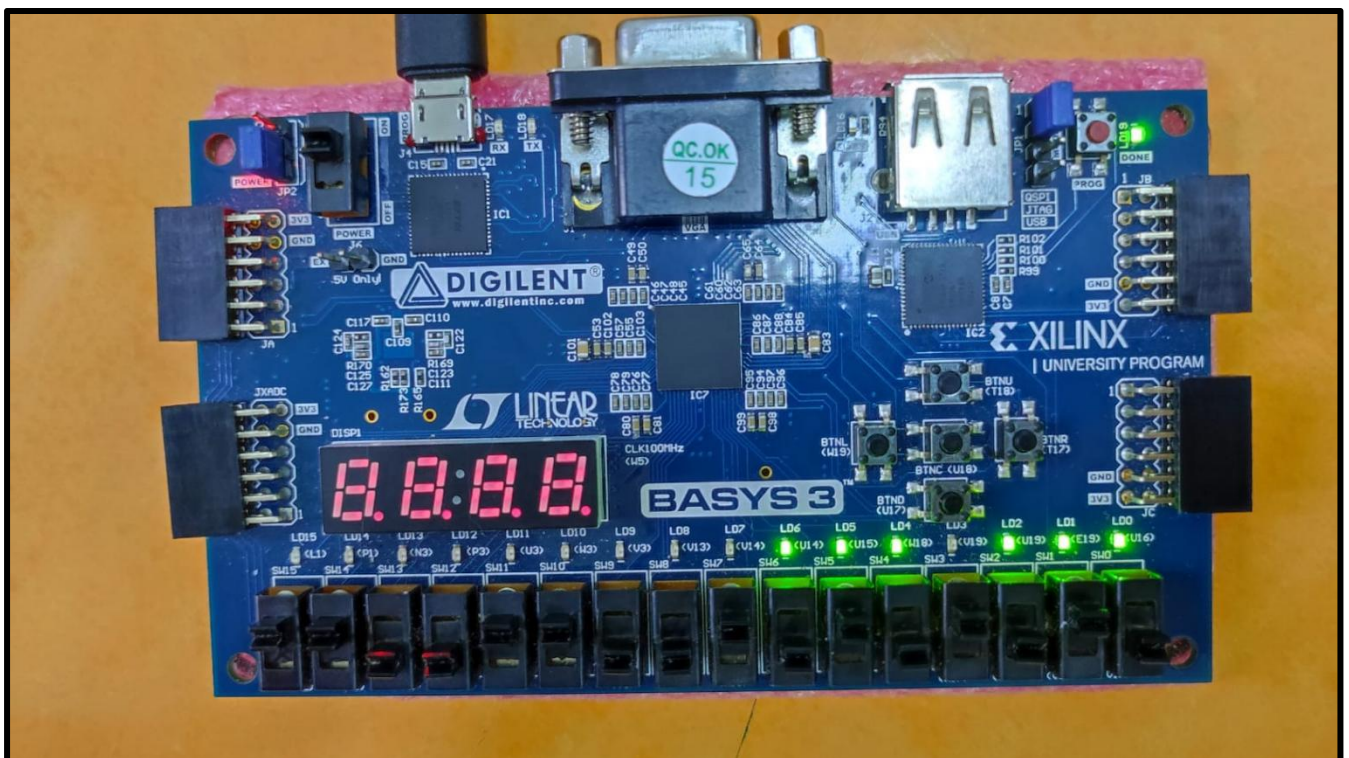
**Operatia “OR” :**

Rezultat: 01110111



**Operatia “NAND” :**

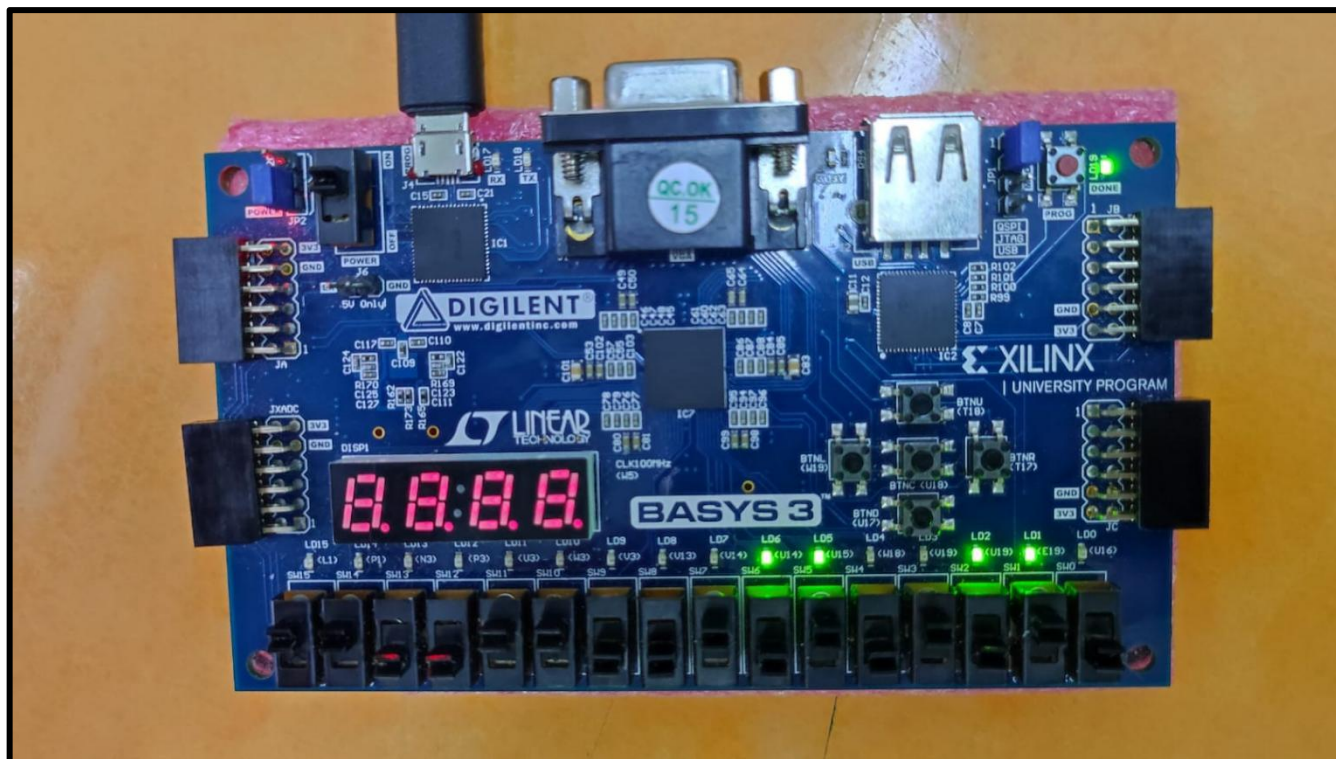
Rezultat: 11101110





**Operatia “XOR” :**

**Rezultat: 01100110**



## 9. Concluzii

In concluzie s-a implementat ca proiect 2 numere pe 8 biti, cu posibilitate de incarcare de la switch-uri a numerelor cu care putem implementa operatiile OR,AND,NAND,XOR , a carui iesire in urma alegerii unei operatii este conectata la led-urile de pe placa de dezvoltare BASYS3.

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>