



## Σχεδίαση Ψηφιακών Συστημάτων

ΕΞΑΜΗΝΙΑΙΑ ΑΣΚΗΣΗ ΘΕΩΡΙΑΣ

ΜΕΡΟΣ 01

Τμήμα Μηχανικών Πληροφορικής  
& Υπολογιστών

(Καθηγητής: Ιωάννης Βογιατζής)

Μάρκος Παντελιδάκης - 18390228 - [ice18390228@uniwa.gr](mailto:ice18390228@uniwa.gr)

**ALU 4-bit**

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  use IEEE.numeric_std.all;
4  USE ieee.std_logic_unsigned.all;
5
6  entity ALU4 is port (
7      ALUin1: in std_logic_vector(3 downto 0);
8      ALUin2: in std_logic_vector(3 downto 0);
9      ALUctrl: in std_logic_vector(3 downto 0);
10     ALUout1: out std_logic_vector(3 downto 0);
11     zero: out std_logic);
12 end ALU4;
13
14 architecture behavioural of ALU4 is
15     signal sig: std_logic_vector(3 downto 0);
16 begin
17     process(ALUin1, ALUin2, ALUctrl)
18     begin
19         case ALUctrl IS
20         when "0000" => sig <= ALUin1 AND ALUin2;
21         when "0001" => sig <= ALUin1 OR ALUin2;
22         when "0010" => sig <= ALUin1 + ALUin2;
23         when "0110" => sig <= ALUin1 - ALUin2 ;
24         when "0111" =>
25             if ALUin1 < ALUin2 then
26                 sig <= "0001";
27             else
28                 sig <= "0000";
29             end if;
30         when others => sig <= sig;
31         end case;
32     end process;
33     zero <='1' when sig="0000" else '0';
34     ALUout1 <= sig;
35 end behavioural;

```

Εικ1.1: Κώδικας αρχείου VHDL κυκλώματος Αριθμητικής και Λογικής μονάδας 4-bit.

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  use IEEE.numeric_std.all;
4
5  entity ALU4_tb is
6  end ALU4_tb;
7
8  architecture test_b of ALU4_tb is
9
10     signal ALUin1l : std_logic_vector (3 downto 0);
11     signal ALUin2l : std_logic_vector (3 downto 0);
12     signal ALUctrl1l : std_logic_vector (3 downto 0);
13     signal ALUout1l : std_logic_vector(3 downto 0);
14     signal zerol : std_logic;
15
16     component ALU4 port(
17         ALUin1 : in std_logic_vector (3 downto 0);
18         ALUin2 : in std_logic_vector (3 downto 0);
19         ALUctrl1 : in std_logic_vector (3 downto 0);
20         ALUout1 : out std_logic_vector(3 downto 0);
21         zero : out std_logic);
22     end component;
23
24     BEGIN
25     M1: ALU4 PORT MAP(ALUin1=>ALUin1l, ALUin2=>ALUin2l, ALUctrl1=>ALUctrl1l,
26         ALUout1=>ALUout1l, zero=>zerol);
27     process
28     begin
29         ALUin1l <= "0010";
30         ALUin2l <= "0100";
31         ALUctrl1l<= "0010";
32         wait for 50 ps;
33         ALUin1l <= "0100";
34         ALUin2l <= "1111";
35         ALUctrl1l<= "0000";
36         wait for 50 ps;
37         ALUin1l <= "0100";
38         ALUin2l <= "1111";
39         ALUctrl1l<= "0001";
40         wait for 50 ps;
41         ALUin1l <= "0100";
42         ALUin2l <= "0010";
43         ALUctrl1l<= "0110";
44         wait for 50 ps;
45         ALUin1l <= "0100";
46         ALUin2l <= "0110";
47         ALUctrl1l<= "0110";
48         wait for 50 ps;
49     end process;
50 end test_b;

```

**Εικ1.2:** Κώδικας αρχείου VHDL testbench κυκλώματος Αριθμητικής και Λογικής μονάδας 4-bit.

**Πίν1.1:** Πίνακας ζητούμενων τιμών ALU.

ALUin1	ALUin2	ALUCTRL	ALUout	zero
0010	0100	0010	0110	0
0100	1111	0000	0100	0
0100	1111	0001	1111	0
0100	0010	0110	0010	0
0100	0110	0110	1110	0

Οι τιμές των εξόδων ALUout και zero της ALU του πίνακα 1.1 επαληθεύονται από την κυματομορφή της Εικόνας 1.3.

	Msgs				
/alu4_tb/ALUin1	0010	0010	0100		
/alu4_tb/ALUin2	0100	0100	1111	0010	0110
/alu4_tb/ALUctrl	0010	0010	0000	0001	0110
/alu4_tb/ALUout	0110	0110	0100	1111	0010
/alu4_tb/zero	0				

Εικ1.3: Παράδειγμα κυματομορφής κυκλώματος Αριθμητικής και Λογικής μονάδας 4-bit.

Με παρόμοιο τρόπο φτιάχνουμε και την ALU των 32 bits:

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  use IEEE.numeric_std.all;
4  USE ieee.std_logic_unsigned.all;
5
6  entity ALU32 is port (
7      ALUin1: in std_logic_vector(31 downto 0);
8      ALUin2: in std_logic_vector(31 downto 0);
9      ALUctrl: in std_logic_vector(3 downto 0);
10     ALUout1: out std_logic_vector(31 downto 0);
11     zero: out std_logic);
12 end ALU32;
13
14 architecture behavioural of ALU32 is
15     signal sig: std_logic_vector(31 downto 0);
16
17 BEGIN
18     process(ALUin1, ALUin2, ALUctrl)
19     begin
20         case ALUctrl IS
21             when "0000" => sig <= ALUin1 AND ALUin2;
22             when "0001" => sig <= ALUin1 OR  ALUin2;
23             when "0010" => sig <= ALUin1 + ALUin2;
24             when "0110" => sig <= ALUin1 - ALUin2;
25             when others => sig <= x"FFFFFFFF";
26         end case;
27     end process;
28     zero <= '1' when sig=x"00000000" else '0';
29     ALUout1 <= sig;
30 end behavioural;

```

Εικ1.4: Κώδικας αρχείου VHDL κυκλώματος Αριθμητικής και Λογικής μονάδας 32-bit.

