Πανεπιστήμιο Δυτικής Αττικής
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών
Σχεδίαση Ψηφιακών Συστημάτων (ICE – 4005) Project_MIPS _2020
Γκούσαρης Κωνσταντίνος – 711171073 – cs171073@uniwa.gr

1: Arithmetic and Logic Unit

1.1 ALU.vhd

```
--MIPS Part_1
--A.L.U. 32bits
--27/05/2020, Konstantinos Gkousaris, 711171073, UniWA--


LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;
use IEEE.std_logic_unsigned.all;

ENTITY alu32 IS PORT (
      aluin1 : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
      aluin2 : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
      aluctrl: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      aluout : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
      zero   : OUT STD_LOGIC);
END alu32;

-- OP mapping :
-- 0000 AND
-- 0001 OR
-- 0010 ADD
-- 0110 SUBTRACT

ARCHITECTURE alu_1 OF alu32 IS
signal result :std_logic_vector(31 downto 0);
BEGIN
PROCESS(aluctrl,aluin1,aluin2,result)
      BEGIN
            IF (aluctrl = "0000") THEN
                  result <= aluin1 AND aluin2;
            ELSIF (aluctrl = "0001") THEN
                  result <= aluin1 OR aluin2;
            ELSIF (aluctrl = "0010") THEN
                  result <= aluin1 + aluin2;
            ELSIF (aluctrl = "0110") THEN
                  result <= aluin1 - aluin2;
            ELSE
                  result <= "00000000000000000000000000000000";
            END IF;

            IF (result="00000000000000000000000000000000") THEN
                  zero <= '1';
            ELSE
                  zero <= '0';
            END IF;
```

```
END PROCESS;
      --zero<='1' WHEN result="00000000000000000000000000000000" ELSE '0';
      aluout <= result;
END;
```

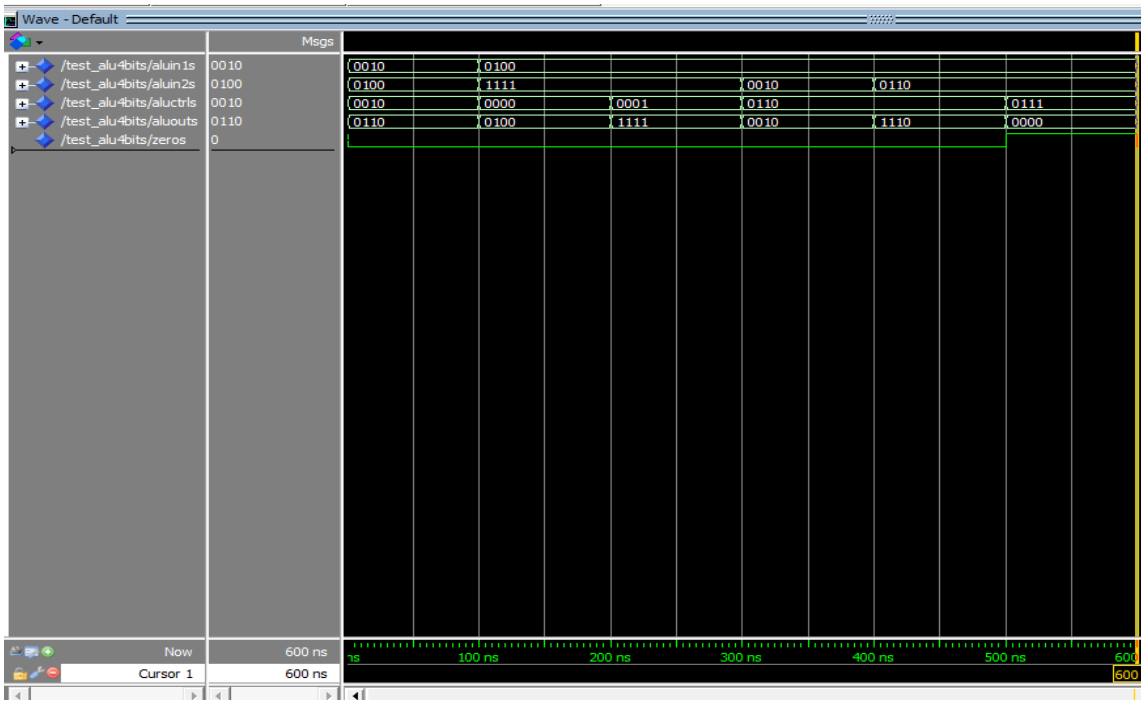
*Παρακάτω θα δοκιμαστεί ο κώδικας και η λειτουργία της άλλου για 4bit αριθμού και των αντίστοιχων πράξεων. Το test bench λειτουργεί για 600 nanosecond.

1.2_ALU_4bits.vhd

```
--MIPS Part_1b
--A.L.U. 4bit
--24/05/2020, Konstantinos Gkousaris, 711171073, UniWA--

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;
use IEEE.std_logic_unsigned.all;

ENTITY alu4 IS PORT (
      aluin1 : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      aluin2 : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      aluctrl: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      aluout : OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
      zero   : OUT STD_LOGIC);
END alu4;

-- OP mapping :
-- 0000 AND
-- 0001 OR
-- 0010 ADD
-- 0110 SUBTRACT
-- 0000 ELSE

ARCHITECTURE alu_4 OF alu4 IS
signal result :std_logic_vector(3 downto 0);
BEGIN
PROCESS(aluctrl,aluin1,aluin2,result)
      BEGIN
      IF (aluctrl = "0000") THEN
            result <= aluin1 AND aluin2;
      ELSIF (aluctrl = "0001") THEN
            result <= aluin1 OR aluin2;
      ELSIF (aluctrl = "0010") THEN
            result <= aluin1 + aluin2;
      ELSIF (aluctrl = "0110") THEN
            result <= aluin1 - aluin2;
      ELSE
            result <= "0000";
      END IF;

      IF (result="0000") THEN
            zero <= '1';
      ELSE
            zero <= '0';
      END IF;

END PROCESS;
      --zero<='1' WHEN result="0000" ELSE '0';
      aluout <= result;
END;
```

1.3 ALU_4bits_testbench.vhd

```
--test bench code
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;
use IEEE.std_logic_unsigned.all;

ENTITY test_ALU4bits IS
END test_ALU4bits;

ARCHITECTURE behavioral OF test_ALU4bits IS

      COMPONENT alu4 IS PORT (
            aluin1 : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
            aluin2 : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
            aluctrl: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
            aluout : OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
            zero   : OUT STD_LOGIC);
      END COMPONENT;

      --component signal
signal  aluin1s, aluin2s, aluctrls, aluouts : std_logic_vector(3 downto 0);
signal  zeros : std_logic;

BEGIN
      ALU4bit1: alu4 PORT MAP (aluin1=>aluin1s,
                        aluin2=>aluin2s,
                        aluctrl=>aluctrls,
                        aluout=>aluouts,
                        zero=>zeros);

      aluin1_process : PROCESS
      BEGIN
            aluin1s <= "0010"; wait for 100 ns; --1
            aluin1s <= "0100"; wait for 100 ns; --2
            aluin1s <= "0100"; wait for 100 ns; --3
            aluin1s <= "0100"; wait for 100 ns; --4
            aluin1s <= "0100"; wait for 100 ns; --5
            --with this extra situation, test the zero output
            aluin1s <= "0100"; wait for 100 ns; --6
      END PROCESS;
      aluin2_process : PROCESS
      BEGIN
            aluin2s <= "0100"; wait for 100 ns; --1
            aluin2s <= "1111"; wait for 100 ns; --2
            aluin2s <= "1111"; wait for 100 ns; --3
            aluin2s <= "0010"; wait for 100 ns; --4
            aluin2s <= "0110"; wait for 100 ns; --5
            --with this extra situation, test the zero output
            aluin2s <= "0110"; wait for 100 ns; --6
      END PROCESS;
      aluctrl_process : PROCESS
      BEGIN
            aluctrls <= "0010"; wait for 100 ns; --1
            aluctrls <= "0000"; wait for 100 ns; --2
            aluctrls <= "0001"; wait for 100 ns; --3
            aluctrls <= "0110"; wait for 100 ns; --4
            aluctrls <= "0110"; wait for 100 ns; --5
            --with this extra situation, test the zero output
            aluctrls <= "0111"; wait for 100 ns; --6
      END PROCESS;
END;
```



***Πινακάς λειτουργίας TEST_BENCH ALU_4bit**

Aluin1	Aluin2	Aluctrl	Aluout	zero
0010	0100	0010 (ADD)	0110	0
0100	1111	0000 (AND)	0100	0
0100	1111	0001 (OR)	1111	0
0100	0010	0110 (SUB)	0010	0
0100	0110	0110 (SUB)	1110	0
0100	0110	0111 (nothing -testing)	0000	1