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

3: Control Units, Sign Extender, Shifter

3.1_ControlUnit.vhd

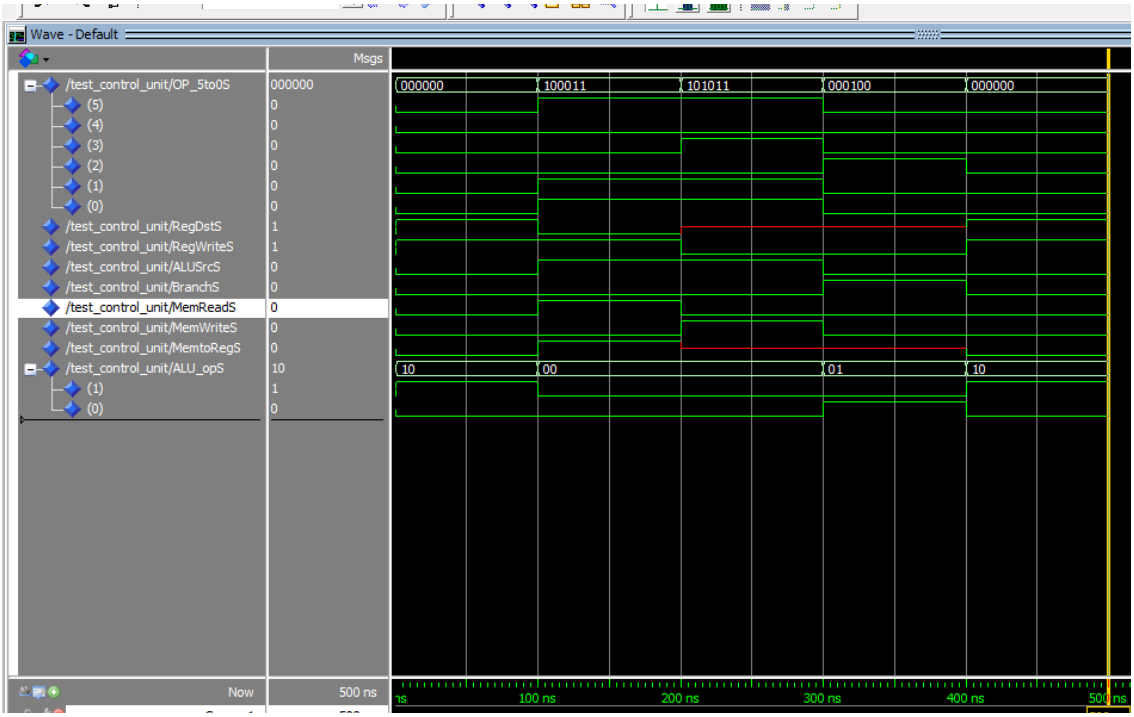
```
--MIPS Part_3
--Control_Unit
--15/06/2020, Konstantinos Gkousaris, 711171073, UniWA

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY Control IS PORT (
      OP_5to0 : IN STD_LOGIC_VECTOR(5 DOWNTO 0);
      RegDst, RegWrite, ALUSrc, Branch: OUT STD_LOGIC;
      MemRead, MemWrite, MemtoReg : OUT STD_LOGIC;
      ALU_op: OUT STD_LOGIC_VECTOR(1 DOWNTO 0));
END Control;

--CONTROL UNIT MAPPING
--INSTRUCTION REGDST ALUSRC MEMTOREG REGWRITE MEMREAD MEMWRITE BRANCH ALUOP1 ALUOP2
--     R        1      0       0        1        0       0        0      1      0
--    LW        0      1       1        1        1       0        0      0      0
--    SW        X      1       X        0        0       1        0      0      0
--   BEQ        X      0       X        0        0       0        1      0      1

--INSTRUCTION
--TYPE R            OPCODE       -> 000000
--LOAD WORD         OPCODE       -> 100011
--STORE WORD        OPCODE       -> 101011
--BRANCH EQUAL      OPCODE       -> 000100
--ELSE

ARCHITECTURE behavioral OF Control IS
BEGIN
      PROCESS(Op_5to0)
      BEGIN
      IF (Op_5to0 = "000000") THEN  --TYPE R, ADD, SUB, AND, OR
            RegDst        <= '1';
            ALUsrc        <= '0';
            MemToReg      <= '0';
            RegWrite      <= '1';
            MemRead       <= '0';
            MemWrite      <= '0';
            Branch        <= '0';
            ALU_op(1)     <= '1';
            ALU_op(0)     <= '0';
      ELSIF (Op_5to0 = "100011") THEN  --LOAD WORD
            RegDst        <= '0';
            ALUsrc        <= '1';
            MemToReg      <= '1';
```

```
            RegWrite      <= '1';
            MemRead       <= '1';
            MemWrite      <= '0';
            Branch        <= '0';
            ALU_op(1)     <= '0';
            ALU_op(0)     <= '0';
      ELSIF (Op_5to0 = "101011") THEN  --STORE WORD
            RegDst        <= 'X';
            ALUsrc        <= '1';
            MemToReg      <= 'X';
            RegWrite      <= '0';
            MemRead       <= '0';
            MemWrite      <= '1';
            Branch        <= '0';
            ALU_op(1)     <= '0';
            ALU_op(0)     <= '0';
      ELSIF (Op_5to0 = "000100") THEN--BRANCH EQUAL
            RegDst        <= 'X';
            ALUsrc        <= '0';
            MemToReg      <= 'X';
            RegWrite      <= '0';
            MemRead       <= '0';
            MemWrite      <= '0';
            Branch        <= '1';
            ALU_op(1)     <= '0';
            ALU_op(0)     <= '1';
      ELSE                              --EVERYTHING ELSE
            RegDst        <= '0';
            ALUsrc        <= '0';
            MemToReg      <= '0';
            RegWrite      <= '0';
            MemRead       <= '0';
            MemWrite      <= '0';
            Branch        <= '0';
            ALU_op(1)     <= '0';
            ALU_op(0)     <= '0';

      END IF;
      END PROCESS;
END;
```

3.1 ControlUnit_testbench.vhd

```
--test bench code
--runs for 400 ns
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY test_control_unit IS
END test_control_unit;

ARCHITECTURE BEHAVIRAL OF test_control_unit IS

COMPONENT Control IS PORT (
      OP_5to0 : IN STD_LOGIC_VECTOR(5 DOWNTO 0);
      RegDst, RegWrite, ALUSrc, Branch: OUT STD_LOGIC;
      MemRead, MemWrite, MemtoReg : OUT STD_LOGIC;
      ALU_op: OUT STD_LOGIC_VECTOR(1 DOWNTO 0));
END COMPONENT;
```

```
	SIGNAL OP_5to0S : STD_LOGIC_VECTOR(5 DOWNTO 0);
	SIGNAL RegDstS, RegWriteS, ALUSrcS, BranchS,MemReadS, MemWriteS, MemtoRegS
			: STD_LOGIC;
	SIGNAL ALU_opS : STD_LOGIC_VE(1 DOWNTO 0);

BEGIN

	TESTCONTROLUNIT1 : Control PORT MAP (OP_5to0S,
			RegDstS, RegWriteS, ALUSrcS, BranchS,
			MemReadS, MemWriteS, MemtoRegS,
			ALU_opS);

	simulation : PROCESS
	BEGIN
		OP_5to0s <= "000000"; wait for 100 ns;
		OP_5to0s <= "100011"; wait for 100 ns;
		OP_5to0s <= "101011"; wait for 100 ns;
		OP_5to0s <= "000100"; wait for 100 ns;
	END PROCESS;
END;
```



Πίνακας Λειτουργίας

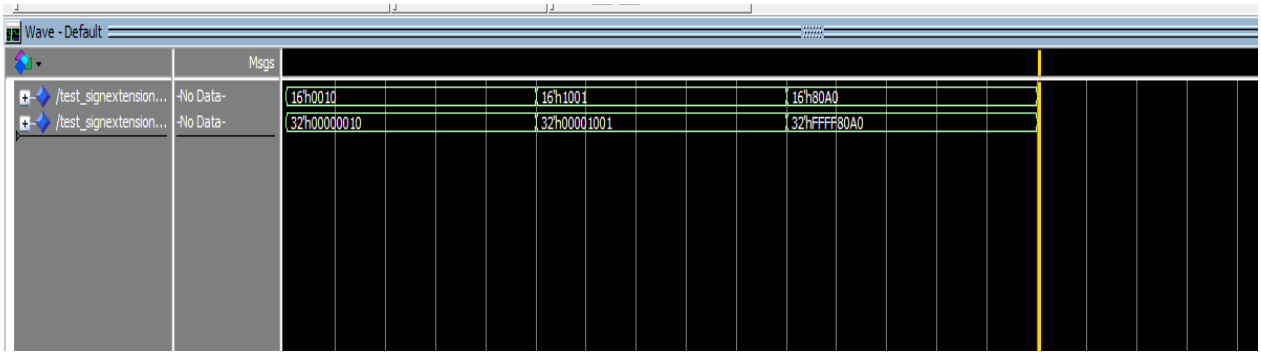
Εντολή	RegDst	ALUSrc	MemToReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	AluOp2
R	1	0	0	1	0	0	0	1	0
LW	0	1	1	1	0	0	0	0	0
SW	X	1	X	0	1	0	0	0	0
BEQ	X	0	X	0	0	1	1	0	1

3.2 SignExtension.vhd

```
--MIPS Part_3
--SIGN EXTANSION
--15/06/2020, Konstantinos Gkousaris, 711171073, UniWA

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY SIGN_Extension IS PORT (
      Insrt_15to0 : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
      Sign_extended: OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
END SIGN_Extension;

--code base on implemantation "ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΣΧΕΔΙΑΣΗ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ"
ARCHITECTURE signextend_1 OF SIGN_Extension IS
      SIGNAL ones : STD_LOGIC_VECTOR(15 DOWNTO 0):= (OTHERS=>'1');
      SIGNAL zeros : STD_LOGIC_VECTOR(15 DOWNTO 0):= (OTHERS=>'0');

BEGIN
       Sign_extended <= ones & Insrt_15to0 when Insrt_15to0(15) = '1'
       else
                    zeros & Insrt_15to0 when Insrt_15to0(15) = '0';
END;
```

3.2 SignExtension_testbench.vhd

```
--test bench
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY test_SignExtension IS
END test_SignExtension;

ARCHITECTURE behavioral OF test_SignExtension IS

COMPONENT SIGN_Extension PORT (
      Insrt_15to0 : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
      Sign_extended: OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
END COMPONENT;


      SIGNAL Insrt_15to0S   : STD_LOGIC_VECTOR(15 DOWNTO 0);
      SIGNAL Sign_extendedS : STD_LOGIC_VECTOR(31 DOWNTO 0);

BEGIN

      SIGNEXTENSIONTEST : SIGN_Extension PORT MAP (
            Insrt_15to0=> Insrt_15to0S,
            Sign_extended =>Sign_extendedS);

      sign_extender_process : PROCESS
      BEGIN
            Insrt_15to0S <= x"0010"; wait for 100 ns;
            Insrt_15to0S <= x"1001"; wait for 100 ns;
            Insrt_15to0S <= x"80A0"; wait for 100 ns;
      END PROCESS;
END;
```



Πίνακας Λειτουργίας

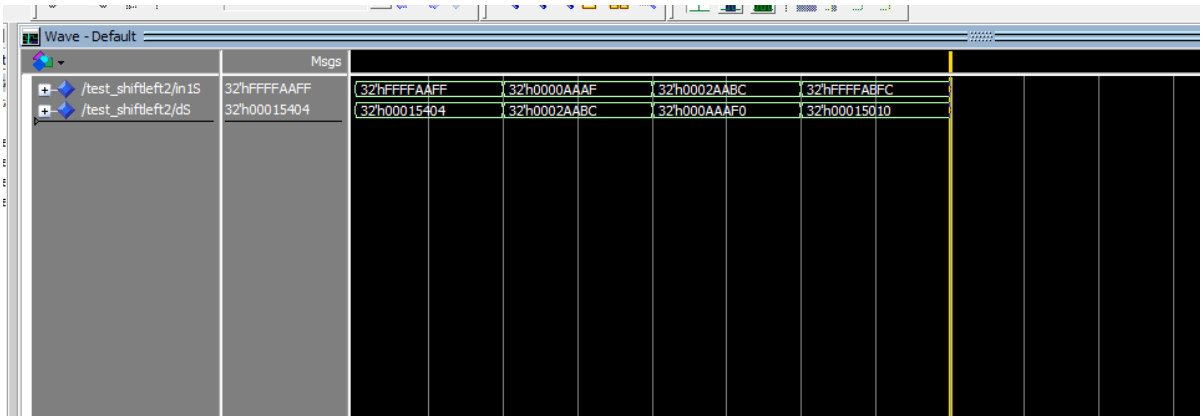
Εισοδός Insrt_15to0	Εξοδός Sign_extended
0x0010	0x0000 0010
0x1001	0x0000 1001
0x80A0	0xFFFF 80A0

3.3 Shiftleft2.vhd

```
--MIPS Part_3
--LEFT SHIFTER '2'
--15/06/2020, Konstantinos Gkousaris, 711171073, UniWA

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

ENTITY shiftleft2 IS PORT (
      in1: IN STD_LOGIC_VECTOR(31 DOWNTO 0);
      d: OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
END shiftleft2;

ARCHITECTURE behavioral OF shiftleft2 IS
--code base on implemantation "ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΣΧΕΔΙΑΣΗ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ"

   signal tmp : unsigned(31 DOWNTO 0);
   --put another signal if someone want to work with shift left higher than '2'
   --signal num : std_logic_vector(3 DOWNTO 0):= "0010";-- shift left 2


BEGIN
      tmp <= to_unsigned(to_integer(signed(in1)),tmp'length) sll 2;
            --to_integer(signed(num));

       d <= std_logic_vector(to_signed(to_integer(tmp),d'length));



END behavioral;
```

3.3 Shiftleft2_testbench.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY test_shiftleft2 IS
END test_shiftleft2;

ARCHITECTURE behavioral OF test_shiftleft2 IS

COMPONENT shiftleft2 IS PORT (
      in1: IN STD_LOGIC_VECTOR(31 DOWNTO 0);
      d: OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
END COMPONENT;

      SIGNAL in1S, dS : STD_LOGIC_VECTOR(31 DOWNTO 0);

BEGIN

      TESTSHIFTLEFT2 : shiftleft2 PORT MAP (in1S,dS);

      simulation : PROCESS
      BEGIN

            in1S  <=    x"0000aaaf"; wait for 100 ns;
            in1S  <=    x"0002aabc"; wait for 100 ns;
            in1S  <=    x"ffffaaff"; wait for 100 ns;
            in1S   <=   x"ffffabfc"; wait for 100 ns;

      END PROCESS;
END;
```



***Πίνακας Λειτουργίας**

INPUT	OUTPUT
0x0000AAAF	0x0002AABC
0xFFFFAAFF	0x00015404
0x0002AABC	0x000AAAF0
0xFFFEABFC	0x00015010

***Παρατηρείται Overflow, δεν παράγονται οι σωστές τιμές σύμφωνα με την εκφώνηση. Θεωρώ σωστές τις τιμές εξαιτίας του πεπερασμένου αριθμού θέσεων που περιεχέι ο std_vector (32). Έγινε εκτέλεση παραπάνω παραδειγμάτων για να διαπιστωθεί η ορθότητά της παρατήρησης.**