Σχεδίαση Ψηφιακών Συστημάτων

ΕΞΑΜΗΝΙΑΙΑ ΑΣΚΗΣΗ ΘΕΩΡΙΑΣ

ΜΕΡΟΣ 04

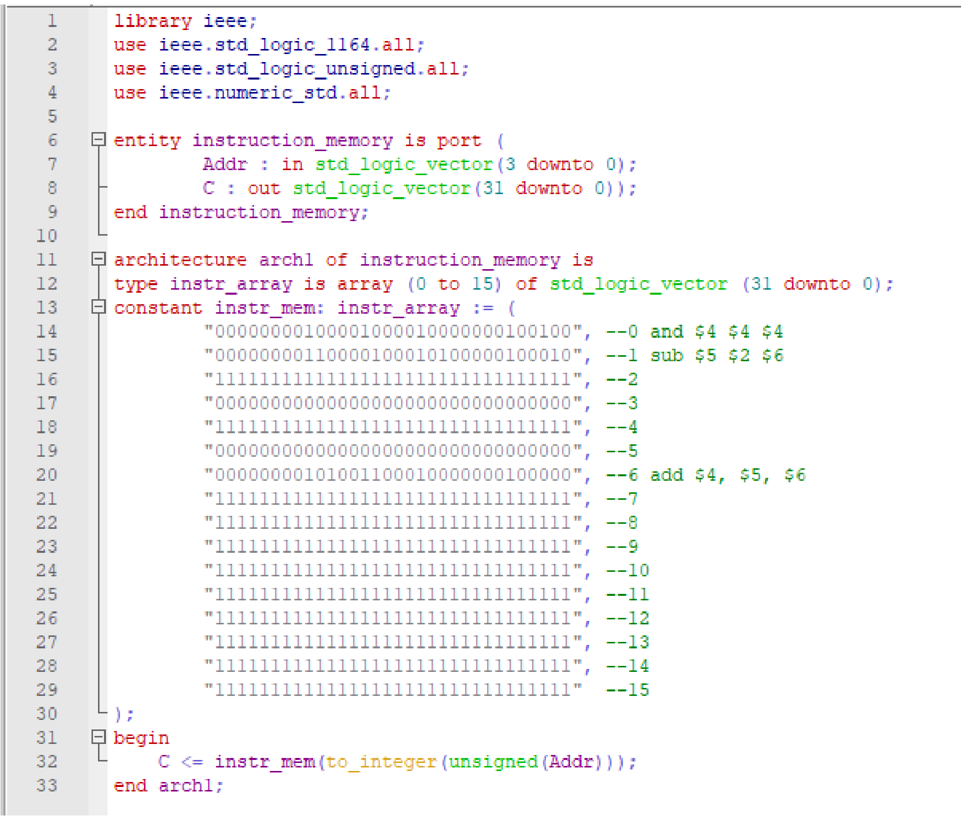
Τμήμα Μηχανικών Πληροφορικής

& Υπολογιστών

(Καθηγητής: Ιωάννης Βογιατζής)

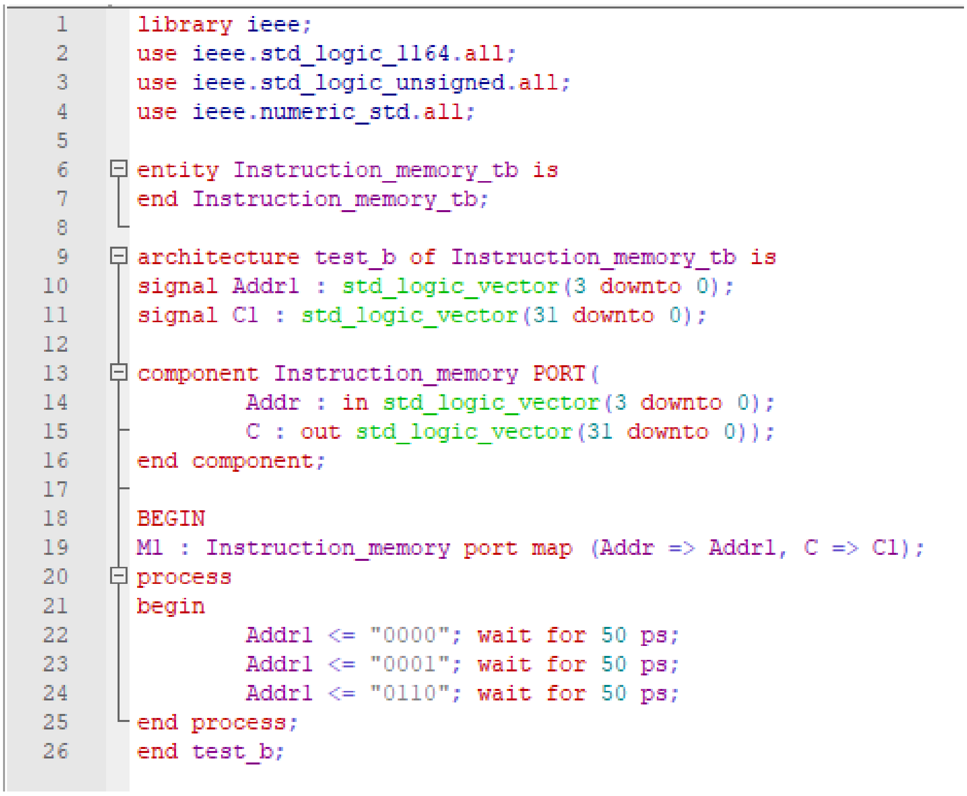
Μάρκος Παντελιδάκης - 18390228 - ice18390228@uniwa.gr

1. Μνήμη Εντολών



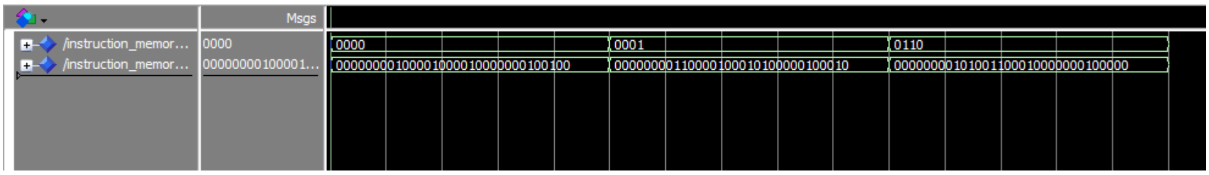
```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;

entity instruction_memory is port (
        Addr : in std_logic_vector(3 downto 0);
        C : out std_logic_vector(31 downto 0));
end instruction_memory;

architecture archl of instruction_memory is
type instr_array is array (0 to 15) of std_logic_vector (31 downto 0);
constant instr_mem: instr_array := (
        "00000000100001000010000000100100", --0 and $4 $4 $4
        "00000000110000100010100000100010", --1 sub $5 $2 $6
        "11111111111111111111111111111111", --2
        "00000000000000000000000000000000", --3
        "11111111111111111111111111111111", --4
        "00000000000000000000000000000000", --5
        "00000000101001100010000000100000", --6 add $4, $5, $6
        "11111111111111111111111111111111", --7
        "11111111111111111111111111111111", --8
        "11111111111111111111111111111111", --9
        "11111111111111111111111111111111", --10
        "11111111111111111111111111111111", --11
        "11111111111111111111111111111111", --12
        "11111111111111111111111111111111", --13
        "11111111111111111111111111111111", --14
        "11111111111111111111111111111111"  --15
);
begin
    C <= instr_mem(to_integer(unsigned(Addr)));
end archl;
```

Εικ4.1.1: Κώδικας αρχείου VHDL κυκλώματος Μνήμης Εντολών MIPS.



```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.numeric_std.all;

entity Instruction_memory_tb is
end Instruction_memory_tb;

architecture test_b of Instruction_memory_tb is
signal Addr1 : std_logic_vector(3 downto 0);
signal C1 : std_logic_vector(31 downto 0);

component Instruction_memory PORT(
        Addr : in std_logic_vector(3 downto 0);
        C : out std_logic_vector(31 downto 0));
end component;

BEGIN
M1 : Instruction_memory port map (Addr => Addr1, C => C1);
process
begin
        Addr1 <= "0000"; wait for 50 ps;
        Addr1 <= "0001"; wait for 50 ps;
        Addr1 <= "0110"; wait for 50 ps;
end process;
end test_b;
```

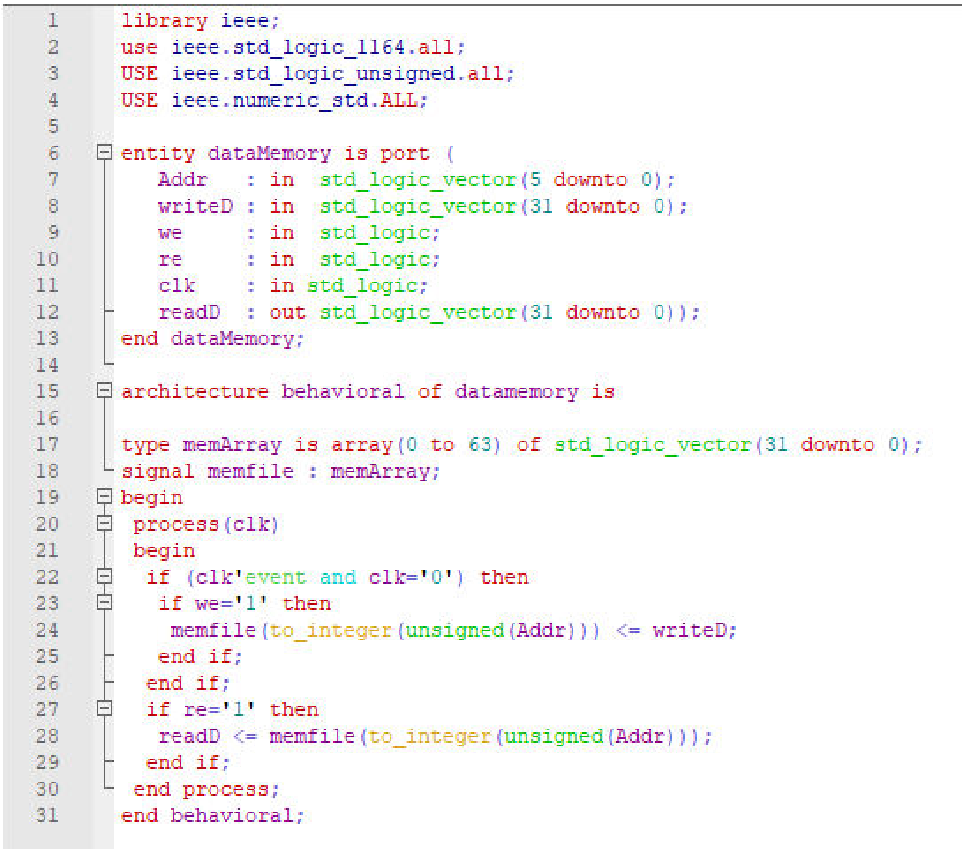
Εικ4.1.2: Κώδικας αρχείου VHDL testbench κυκλώματος Μνήμης Εντολών MIPS.



```
                          Msgs
/instruction_memor...     0000                 0000                               0001                               0110
/instruction_memor...     00000000100001...    00000000100001000010000000100100   00000000110000100010100000100010   00000000101001100010000000100000
```

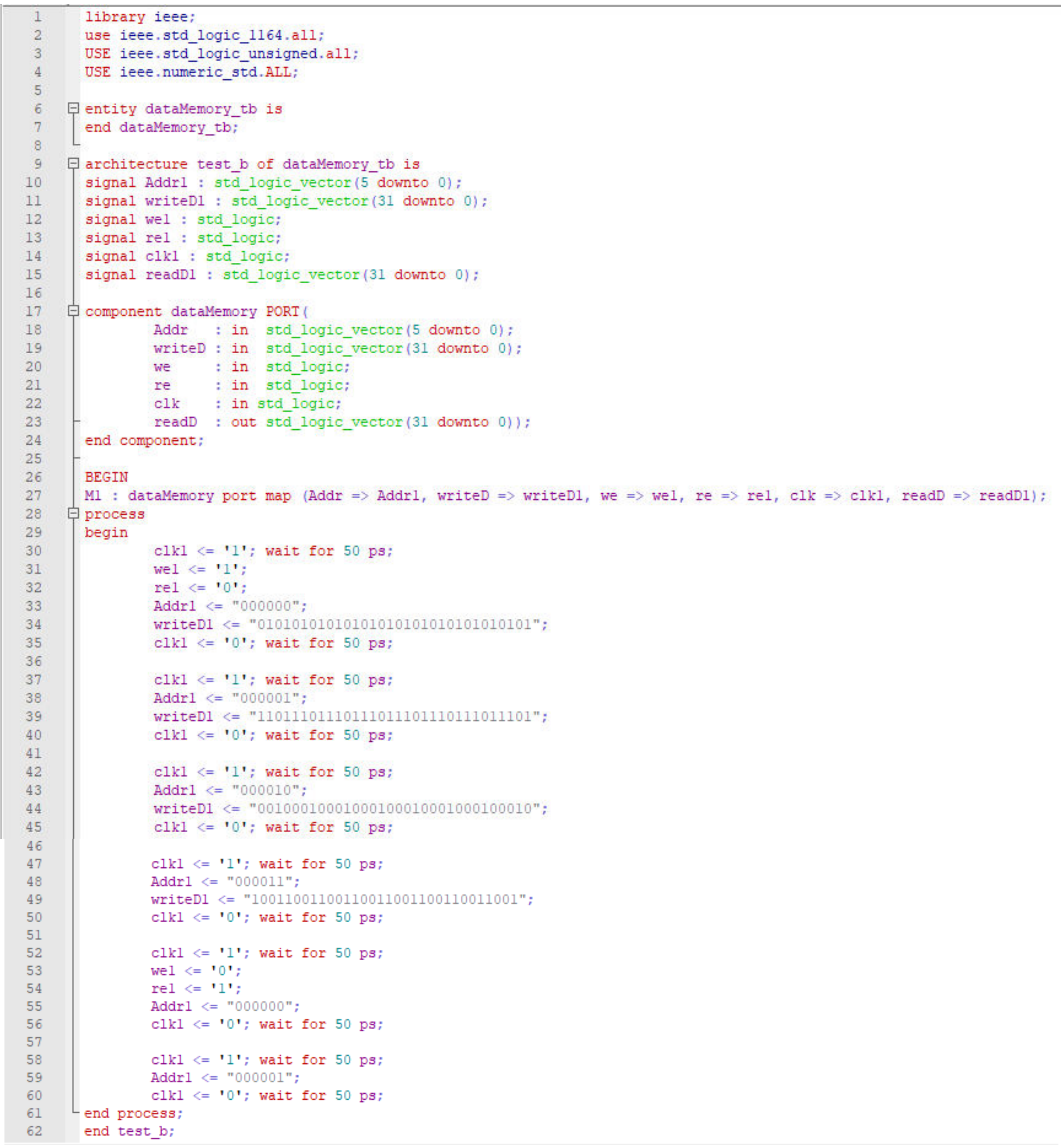
Εικ4.1.3: Παράδειγμα κυματομορφής κυκλώματος Μνήμης Εντολών MIPS.

2. Μνήμη δεδομένων



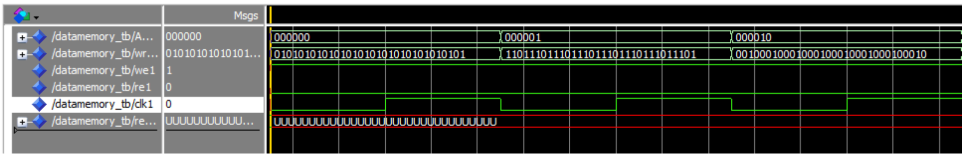
```
library ieee;
use ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;

entity dataMemory is port (
   Addr   : in  std_logic_vector(5 downto 0);
   writeD : in  std_logic_vector(31 downto 0);
   we     : in  std_logic;
   re     : in  std_logic;
   clk    : in std_logic;
   readD  : out std_logic_vector(31 downto 0));
end dataMemory;

architecture behavioral of datamemory is

type memArray is array(0 to 63) of std_logic_vector(31 downto 0);
signal memfile : memArray;
begin
 process(clk)
 begin
  if (clk'event and clk='0') then
   if we='1' then
    memfile(to_integer(unsigned(Addr))) <= writeD;
   end if;
  end if;
  if re='1' then
   readD <= memfile(to_integer(unsigned(Addr)));
  end if;
 end process;
end behavioral;
```

Εικ4.2.1: Κώδικας αρχείου VHDL κυκλώματος Μνήμης δεδομένων MIPS.

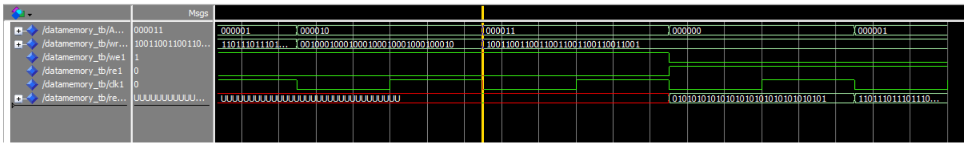


```
library ieee;
use ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;

entity dataMemory_tb is
end dataMemory_tb;

architecture test_b of dataMemory_tb is
signal Addr1 : std_logic_vector(5 downto 0);
signal writeD1 : std_logic_vector(31 downto 0);
signal we1 : std_logic;
signal re1 : std_logic;
signal clk1 : std_logic;
signal readD1 : std_logic_vector(31 downto 0);

component dataMemory PORT(
        Addr   : in  std_logic_vector(5 downto 0);
        writeD : in  std_logic_vector(31 downto 0);
        we     : in  std_logic;
        re     : in  std_logic;
        clk    : in std_logic;
        readD  : out std_logic_vector(31 downto 0));
end component;

BEGIN
M1 : dataMemory port map (Addr => Addr1, writeD => writeD1, we => we1, re => re1, clk => clk1, readD => readD1);
process
begin
        clk1 <= '1'; wait for 50 ps;
        we1 <= '1';
        re1 <= '0';
        Addr1 <= "000000";
        writeD1 <= "01010101010101010101010101010101";
        clk1 <= '0'; wait for 50 ps;

        clk1 <= '1'; wait for 50 ps;
        Addr1 <= "000001";
        writeD1 <= "11011101110111011101110111011101";
        clk1 <= '0'; wait for 50 ps;

        clk1 <= '1'; wait for 50 ps;
        Addr1 <= "000010";
        writeD1 <= "00100010001000100010001000100010";
        clk1 <= '0'; wait for 50 ps;

        clk1 <= '1'; wait for 50 ps;
        Addr1 <= "000011";
        writeD1 <= "10011001100110011001100110011001";
        clk1 <= '0'; wait for 50 ps;

        clk1 <= '1'; wait for 50 ps;
        we1 <= '0';
        re1 <= '1';
        Addr1 <= "000000";
        clk1 <= '0'; wait for 50 ps;

        clk1 <= '1'; wait for 50 ps;
        Addr1 <= "000001";
        clk1 <= '0'; wait for 50 ps;
end process;
end test_b;
```

Εικ4.2.2: Κώδικας αρχείου VHDL testbench κυκλώματος Μνήμης δεδομένων MIPS.



Εικ4.2.3: Αναπαράσταση εγγραφής δεδομένων σε κυματομορφή στη μνήμη δεδομένων MIPS.



Εικ4.2.4: Αναπαράσταση ανάγνωσης δεδομένων σε κυματομορφή από τη μνήμη δεδομένων MIPS.