*****Instituto Politécnico Nacional***

***Escuela Superior de Cómputo***

*Arquitectura de Computadoras*

***Practica 8: Memoria de Programa***

***Nombre:*** *Sampayo Hernández Mauro*

***Grupo:*** *3CV8*

***Profesor:*** *Nayeli Vega García*

***Fecha de entrega:*** *9 de abril del 2020*

**Código de implementación:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.std\_logic\_arith.all;

use IEEE.std\_logic\_unsigned.all;

entity MemoriaPrograma is

generic (d : integer := 25;

a : integer := 16);

Port (PC : in STD\_LOGIC\_VECTOR (a-1 downto 0);

Inst : out STD\_LOGIC\_VECTOR (d-1 downto 0));

end MemoriaPrograma;

architecture Behavioral of MemoriaPrograma is

--INSTRUCCIONES

--Tipo I

constant LI : std\_logic\_vector(4 downto 0) := "00001";

constant LWI : std\_logic\_vector(4 downto 0) := "00010";

constant LW : std\_logic\_vector(4 downto 0) := "10111";

constant SWI : std\_logic\_vector(4 downto 0) := "00011";

constant SW : std\_logic\_vector(4 downto 0) := "00100";

constant ADDI : std\_logic\_vector(4 downto 0):= "00101";

constant SUBI : std\_logic\_vector(4 downto 0):= "00110";

constant ANDI: std\_logic\_vector(4 downto 0) := "00111";

constant ORI : std\_logic\_vector(4 downto 0) := "01000";

constant XORI : std\_logic\_vector(4 downto 0) := "01001";

constant NANDI : std\_logic\_vector(4 downto 0):= "01010";

constant NORI: std\_logic\_vector(4 downto 0) := "01011";

constant XNORI : std\_logic\_vector(4 downto 0):= "01100";

constant BEQI : std\_logic\_vector(4 downto 0):= "01101";

constant BNEI : std\_logic\_vector(4 downto 0):= "01110";

constant BLTI : std\_logic\_vector(4 downto 0):= "01111";

constant BLETI : std\_logic\_vector(4 downto 0):= "10000";

constant BGTI : std\_logic\_vector(4 downto 0):= "10001";

constant BGETI : std\_logic\_vector(4 downto 0):= "10010";

--Tipo R

constant TR : std\_logic\_vector(4 downto 0) := "00000";--Operación Tipo R

constant ADD : std\_logic\_vector(3 downto 0) := "0000";

constant SUB : std\_logic\_vector(3 downto 0) := "0001";

constant OpAND : std\_logic\_vector(3 downto 0) := "0010";

constant OpOR : std\_logic\_vector(3 downto 0) := "0011";

constant OpXOR : std\_logic\_vector(3 downto 0) := "0100";

constant OpNAND: std\_logic\_vector(3 downto 0) := "0101";

constant OpNOR : std\_logic\_vector(3 downto 0) := "0110";

constant OpXNOR : std\_logic\_vector(3 downto 0) := "0111";

constant OpNOT : std\_logic\_vector(3 downto 0) := "1000";

constant OpSLL : std\_logic\_vector(3 downto 0) := "1001";

constant OpSRL : std\_logic\_vector(3 downto 0) := "1010";

--Tipo J

constant B: std\_logic\_vector(4 downto 0):= "10011";

constant CALL : std\_logic\_vector(4 downto 0):= "10100";

--Otros

constant RET : std\_logic\_vector(4 downto 0):= "10101";

constant NOP : std\_logic\_vector(4 downto 0):= "10110";

--Sin Uso

constant SU : std\_logic\_vector(3 downto 0) := "0000";--Sin Uso

--REGISTROS

constant R0 : std\_logic\_vector(3 downto 0) := "0000";

constant R1 : std\_logic\_vector(3 downto 0) := "0001";

constant R2 : std\_logic\_vector(3 downto 0) := "0010";

constant R3 : std\_logic\_vector(3 downto 0) := "0011";

constant R4 : std\_logic\_vector(3 downto 0) := "0100";

constant R5 : std\_logic\_vector(3 downto 0) := "0101";

constant R6 : std\_logic\_vector(3 downto 0) := "0110";

constant R7 : std\_logic\_vector(3 downto 0) := "0111";

constant R8 : std\_logic\_vector(3 downto 0) := "1000";

constant R9 : std\_logic\_vector(3 downto 0) := "1001";

constant R10 : std\_logic\_vector(3 downto 0) := "1010";

constant R11 : std\_logic\_vector(3 downto 0) := "1011";

constant R12 : std\_logic\_vector(3 downto 0) := "1100";

constant R13 : std\_logic\_vector(3 downto 0) := "1101";

constant R14 : std\_logic\_vector(3 downto 0) := "1110";

constant R15 : std\_logic\_vector(3 downto 0) := "1111";

--COMANDOS :0

type banco is array (0 to (2\*\*a)-1) of std\_logic\_vector(d-1 downto 0);

constant memProg : banco := (

LI&R0&x"0000", --0 LI R0, #0

LI&R1&x"0001", --1 LI R1, #1

LI&R2&x"0000", --2 LI R2, #0

LI&R3&x"000c", --3 LI R3, #12

TR&R4&R0&R1&SU&ADD, --4 Ciclo: ADD R4, R0, R1(R4 = R0 + R1)

SWI&R4&x"0048", --5 SWI R4, 72(mem[72] = R4)

ADDI&R0&R1&x"000", --6 ADDI R0, R1, #0 (R0 = R1)

SUBI&R1&R4&x"000", --7 SUBI R1, R4, #0 (R1 = R4)

ADDI&R2&R2&x"001", --8 ADDI R2, R2, #1 (R2 = R2 +1)

BNEI&R3&R2&x"004", --9 BNEI R3, R2, Ciclo (R2 != R3)

NOP&SU&SU&SU&SU&SU, --10 fin: NOP

B&SU&x"000a", --11 B fin

others => (others => '0'));

begin

Inst <= memProg(conv\_integer(PC));

end Behavioral;

**Código de simulación:**

library IEEE;

LIBRARY STD;

USE STD.TEXTIO.ALL;

USE IEEE.std\_logic\_TEXTIO.ALL; --PERMITE USAR STD\_LOGIC

USE IEEE.std\_logic\_1164.ALL;

USE IEEE.std\_logic\_UNSIGNED.ALL;

USE IEEE.std\_logic\_ARITH.ALL;

entity MemoriaPrograma\_tb is

end MemoriaPrograma\_tb;

architecture Behavioral of MemoriaPrograma\_tb is

component MemoriaPrograma is

Port (PC : in STD\_LOGIC\_VECTOR (15 downto 0);

Inst : out STD\_LOGIC\_VECTOR (24 downto 0));

end component;

--Inputs

signal PC : std\_logic\_vector(15 downto 0);

--Outputs

signal Inst : std\_logic\_vector(24 downto 0);

begin

uut: MemoriaPrograma Port Map (

PC => PC,

Inst => Inst);

-- Stimulus process

stim\_proc: process

file ARCH\_RES : TEXT;--archivo de resultados

variable LINEA\_RES : line;--linea de resultado

file ARCH\_VEC : TEXT;--archivo de vectores

variable LINEA\_VEC : line;--linea de vectores

--Variables

variable V\_PC: STD\_LOGIC\_VECTOR(15 downto 0);

variable V\_INST: STD\_LOGIC\_VECTOR(24 downto 0);

--Cadena

variable CADENA : string(1 to 4);

begin

file\_open(ARCH\_VEC, "VECTORES.txt", READ\_MODE);

file\_open(ARCH\_RES, "RESULTADO.txt", WRITE\_MODE);

--Imprension del Encabezado

CADENA := " A";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH+4);

CADENA := " COD";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH);

CADENA := "19..";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH);

CADENA := "15..";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH);

CADENA := "11..";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH);

CADENA := "7..4";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH);

CADENA := "3..0";

write(LINEA\_RES, CADENA, right, CADENA'LENGTH+1);

writeline(ARCH\_RES,LINEA\_RES);--Escribe la linea en el archivo

--Imprimiendo RESULTADOS

wait for 10 ns;

for i in 0 to 85 loop

readline(ARCH\_VEC,LINEA\_VEC);

Hread(LINEA\_VEC, V\_PC);

PC<= V\_PC;

wait for 10 ns;

V\_INST := Inst; -- asignando salida

Hwrite(LINEA\_RES, V\_PC, right, 5);

--write(LINEA\_RES, " ", right, 1);

write(LINEA\_RES, V\_INST, right, V\_INST'LENGTH+1);

writeline(ARCH\_RES,LINEA\_RES);--Escribe la linea en el archivo

end loop;

file\_close(ARCH\_VEC);--Cierra el archivo

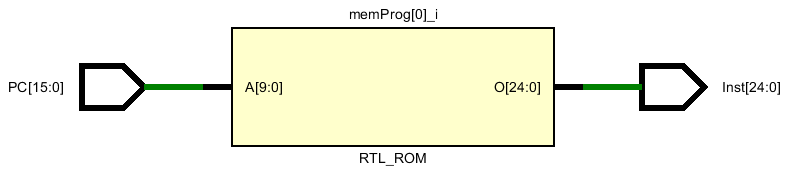
file\_close(ARCH\_RES);--Cierra el archivo

wait;

end process;-- Stimulus process

end Behavioral;

**Diagrama RTL:**



**Simulación:**

