*****Instituto Politécnico Nacional***

***Escuela Superior de Cómputo***

*Arquitectura de Computadoras*

***ESCOMIPS***

***Nombre:*** *Martinez Ortega Juan Yael*

***Grupo:*** *3CV2*

***Profesor:*** *Nayeli Vega García*

***Fecha de entrega:*** *3 de julio del 2020*

**Código de Implementación del procesador:**

**library** IEEE**;**

**use** IEEE**.**STD\_LOGIC\_1164**.ALL;**

**entity** ESCOMIPS **is**

**generic** **(** d **:** integer **:=** 25**;**

x **:** integer **:=** 16**;**

n**:** integer **:=** 4**;**

tam\_Microinstruccion**:** integer**:=**20**;**

tam\_OPCODE**:** integer**:=**5**;**

tam\_FUNCODE**:** integer**:=**4**);**

**Port** **(** clr0**,** clk**:** **in** STD\_LOGIC**;**

PC**,** RD1**,** RD2**,** resALU**,** BusSR**:** **out** STD\_LOGIC\_VECTOR **(**15 **downto** 0**));**

**end** ESCOMIPS**;**

**architecture** Behavioral **of** ESCOMIPS **is**

**component** Registro\_clr **is**

**Port** **(** clk**,** rclr**:** **in** STD\_LOGIC**;**

clr**:** **out** STD\_LOGIC**);**

**end** **component;**

**component** ALU\_Nbits **is**

**Port** **(** a**,** b **:** **in** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**);**

aluop **:** **in** STD\_LOGIC\_VECTOR **(**n**-**1 **downto** 0**);**--3 downto 0

res **:** **out** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**);**

bn**,** z**,** co**,** ov **:** **out** STD\_LOGIC**);**

**end** **component;**

**component** ArchivodeRegistros **is**

**Port** **(** readReg1**,** readReg2**,** writeReg**,** shamt **:** **in** STD\_LOGIC\_VECTOR **(**n**-**1 **downto** 0**);**

writeData **:** **in** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**);**

readData1**,** readData2 **:** **inout** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**);**

WR**,** SHE**,** dir**,** clk **,** clr **:** **in** STD\_LOGIC**);**

**end** **component;**

**component** MemoriaDatos **is**

**Port** **(**dataIn**:** **in** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**);**

dir **:** **in** STD\_LOGIC\_VECTOR **(**9 **downto** 0**);**

clk**,** WD **:** **in** STD\_LOGIC**;**

dataOut **:** **out** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**));**

**end** **component;**

**component** Pila\_MemoriaPrograma **is**

**Port(** PC\_in **:** **in** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**);**

clk**,** clr**,** UP**,** DW**,** WPC **:** **in** STD\_LOGIC**;**

PC\_out **:** **out** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**);**

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

**end** **component;**

**component** UnidadControl **is**

**Port** **(** funCode **:** **in** STD\_LOGIC\_VECTOR **(**tam\_FUNCODE**-**1 **downto** 0**);**

opCode **:** **in** STD\_LOGIC\_VECTOR **(**tam\_OPCODE**-**1 **downto** 0**);**

banderas **:** **in** STD\_LOGIC\_VECTOR **(**n**-**1 **downto** 0**);**

clk**,** clr**,** LF**:** **in** STD\_LOGIC**;**

Microinstruccion **:** **out** STD\_LOGIC\_VECTOR **(**tam\_Microinstruccion**-**1 **downto** 0**));**

**end** **component;**

**component** Mux2a1\_16bits **is**

**Port** **(** e0**,** e1 **:** **in** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**);**

condicion **:** **in** std\_logic**;**

salida **:** **out** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**));**

**end** **component;**

**component** Mux2a1\_4bits **is**

**Port** **(** e0**,** e1 **:** **in** STD\_LOGIC\_VECTOR **(**n**-**1 **downto** 0**);**

condicion **:** **in** STD\_LOGIC**;**

salida **:** **out** STD\_LOGIC\_VECTOR **(**n**-**1 **downto** 0**));**

**end** **component;**

**component** extSigno **is**

**Port** **(** entrada**:** **in** STD\_LOGIC\_VECTOR **(**11 **downto** 0**);**

salida**:** **out** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**));**

**end** **component;**

**component** extDireccion **is**

**Port** **(** entrada**:** **in** STD\_LOGIC\_VECTOR **(**11 **downto** 0**);**

salida**:** **out** STD\_LOGIC\_VECTOR **(**x**-**1 **downto** 0**));**

**end** **component;**

**signal** clr**,** bn**,** z**,** co**,** ov**,** LF**,** SHE**,** DIR**,** WR**,** WD**,** UP**,** DW**,** WPC**:** STD\_LOGIC**;**

**signal** SDMP**,** SR2**,** SWD**,** SEXT**,** SOP1**,** SOP2**,** SDMD**,** SR**:** STD\_LOGIC**;**

**signal** a**,** b**,** res**,** writeData**,** readData1**,** readData2**,** PC\_in**,** PC\_out**,** sal\_Signo**,** sal\_Dir**,** extensor**,** dirMemData**,** dataOut**,** SR\_Out**:** STD\_LOGIC\_VECTOR**(**x**-**1 **downto** 0**);**

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

**signal** readReg1**,** readReg2**,** writeReg**,** shamt**,** aluop**:** STD\_LOGIC\_VECTOR **(**n**-**1 **downto** 0**);**

**begin**

CLR\_reg**:** Registro\_clr

**Port** **Map** **(** clk **=>** clk**,**

rclr **=>**clr0**,**

clr **=>** clr**);**

Unidad\_Control**:** UnidadControl --(0, N, Z, C)

**Port** **Map** **(** banderas**(**3**)** **=>** ov**,**

banderas**(**2**)** **=>** bn**,**

banderas**(**1**)** **=>** z**,**

banderas**(**0**)** **=>** co**,**

LF **=>** LF**,**

clk **=>** clk**,**

clr **=>** clr**,**

opCode **=>** Inst**(**24 **downto** 20**),**

funCode **=>** Inst**(**3 **downto** 0**),**

Microinstruccion**(**19**)** **=>** SDMP**,**

Microinstruccion**(**18**)** **=>** UP**,**

Microinstruccion**(**17**)** **=>** DW**,**

Microinstruccion**(**16**)** **=>** WPC**,**

Microinstruccion**(**15**)** **=>** SR2**,**

Microinstruccion**(**14**)** **=>** SWD**,**

Microinstruccion**(**13**)** **=>** SEXT**,**

Microinstruccion**(**12**)** **=>** SHE**,**

Microinstruccion**(**11**)** **=>** DIR**,**

Microinstruccion**(**10**)** **=>** WR**,**

Microinstruccion**(**9**)** **=>** SOP1**,**

Microinstruccion**(**8**)** **=>** SOP2**,**

Microinstruccion**(**7**)** **=>** aluop**(**3**),**

Microinstruccion**(**6**)** **=>** aluop**(**2**),**

Microinstruccion**(**5**)** **=>** aluop**(**1**),**

Microinstruccion**(**4**)** **=>** aluop**(**0**),**

Microinstruccion**(**3**)** **=>** SDMD**,**

Microinstruccion**(**2**)** **=>** WD**,**

Microinstruccion**(**1**)** **=>** SR**,**

Microinstruccion**(**0**)** **=>** LF**);**

Pila\_y\_MemoriaPrograma**:** Pila\_MemoriaPrograma

**Port** **map(** PC\_in **=>** PC\_in**,**

clk **=>** clk**,**

clr **=>** clr**,**

UP **=>** UP**,**

DW **=>** DW**,**

WPC **=>** WPC**,**

PC\_out **=>** PC\_out**,**

Inst **=>** Inst**);**

mux\_SR2**:** mux2a1\_4bits

**Port** **map(** e1 **=>** Inst**(**19 **downto** 16**),**

e0 **=>** Inst**(**11 **downto** 8**),**

condicion **=>** SR2**,**

salida **=>** readReg2**);**

Mux\_SWD**:** Mux2a1\_16bits

**Port** **map(** e1 **=>** SR\_Out**,**

e0 **=>** Inst**(**15 **downto** 0**),**

condicion **=>** SWD**,**

salida **=>** writeData**);**

Extensor\_Signo**:** extSigno

**Port** **Map** **(** entrada **=>** Inst**(**11 **downto** 0**),**

salida **=>** sal\_Signo**);**

Extensor\_Direccion**:** extDireccion

**Port** **Map** **(** entrada **=>** Inst**(**11 **downto** 0**),**

salida **=>** sal\_Dir**);**

Mux\_SEXT**:** Mux2a1\_16bits

**Port** **map(** e1 **=>** sal\_Dir**,**

e0 **=>** sal\_Signo**,**

condicion **=>** sext**,**

salida **=>** extensor**);**

Achivo\_Registros**:** ArchivodeRegistros

**Port** **map(** readReg1 **=>** Inst**(**15 **downto** 12**),**

readReg2 **=>** readReg2**,**

writeReg **=>** Inst**(**19 **downto** 16**),**

shamt **=>** Inst**(**7 **downto** 4**),**

writeData **=>** writeData**,**

readData1 **=>** readData1**,**

readData2 **=>** readData2**,**

WR **=>** WR**,**

SHE **=>** SHE**,**

dir **=>** dir**,**

clk **=>** clk**,**

clr **=>** clr**);**

Mux\_SOP1**:** Mux2a1\_16bits

**Port** **map(** e1 **=>** PC\_out**,**

e0 **=>** readData1**,**

condicion **=>** SOP1**,**

salida **=>** a**);**

muxSOP2**:** Mux2a1\_16bits

**Port** **map(** e1 **=>** extensor**,**

e0 **=>** readData2**,**

condicion **=>** SOP2**,**

salida **=>** b**);**

ALU**:** ALU\_Nbits

**Port** **map(** a **=>** a**,**

b **=>** b**,**

aluop **=>** aluop**,**

res **=>** res**,**

co **=>** co**,**

bn **=>** bn**,**

z **=>** z**,**

ov **=>** ov**);**

muxSDMD**:** Mux2a1\_16bits

**Port** **map(** e1 **=>** Inst**(**15 **downto** 0**),**

e0 **=>** res**,**

condicion **=>** SDMD**,**

salida **=>** dirMemData**);**

Memoria\_Datos**:** MemoriaDatos

**Port** **map(** dataIn **=>** readData2**,**

dir **=>** dirMemData**(**9 **downto** 0**),**

clk **=>** clk**,**

WD **=>** WD**,**

dataOut **=>** dataOut**);**

muxSR**:** Mux2a1\_16bits

**Port** **map(** e1 **=>** res**,**

e0 **=>** dataOut**,**

condicion **=>** SR**,**

salida **=>** SR\_Out**);**

Mux\_SDMP**:** Mux2a1\_16bits

**Port** **map(** e1 **=>** SR\_Out**,**

e0 **=>** Inst**(**15 **downto** 0**),**

condicion **=>** SDMP**,**

salida **=>** PC\_in**);**

--ASIGNACION DE SALIDAS

PC **<=** PC\_Out**;**

RD1 **<=** readData1**;**

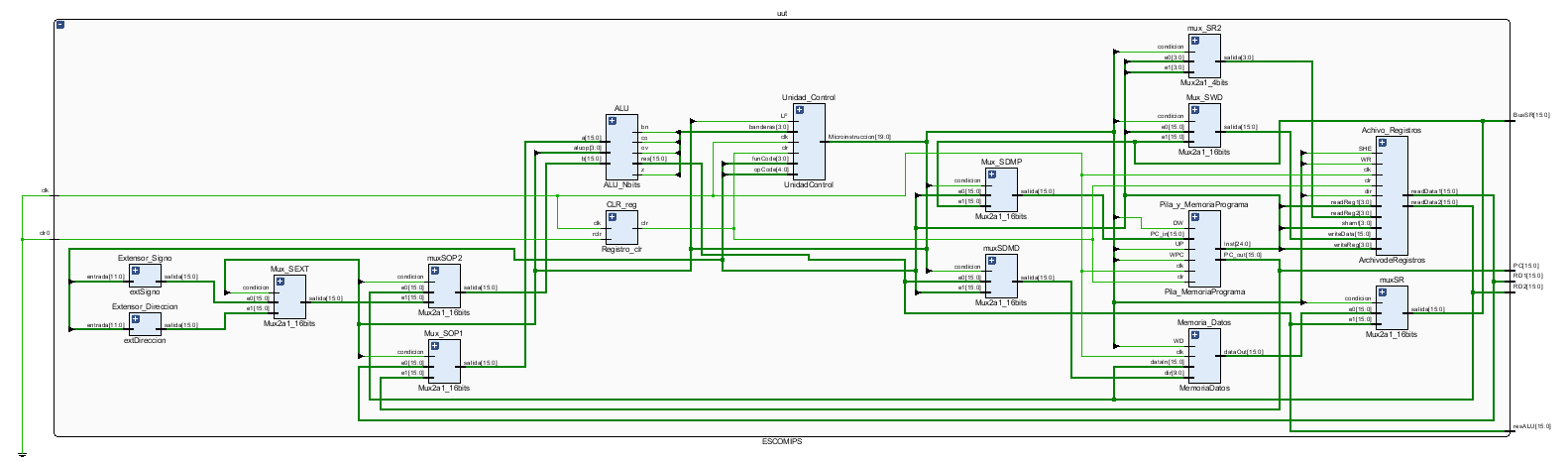
RD2 **<=** readData2**;**

resALU **<=** res**;**

BusSr **<=** SR\_Out**;**

**end** Behavioral**;**

**Diagrama RTL:**



**Código para llenar a la memoria con el problema del punto 2:**

**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 **:=** 10**);**

**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"**;**--Operaci󮠔ipo 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 **:=** **(**

--SUMA

LI **&** R0 **&** x"0001"**,** --0. LI R0, #1

LI **&** R1 **&** x"0007"**,** --1. LI R1, #7

TR **&** R1 **&** R1 **&** R0 **&** SU **&** ADD**,** --2. SUMA: ADD R1, R1, R0

SWI **&** R1 **&** x"0005"**,** --3. SWI R1, 5

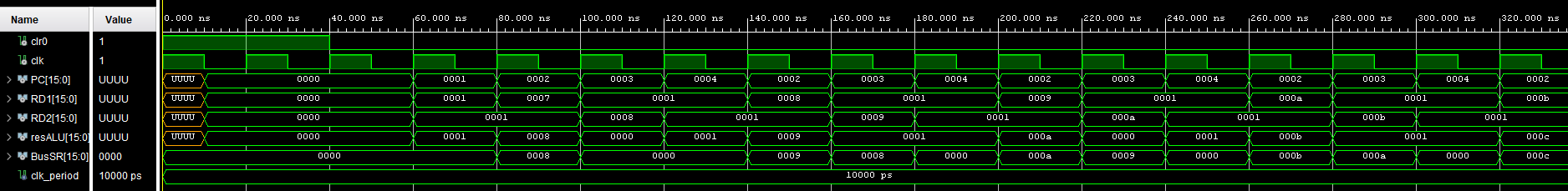
B **&** SU **&** x"0002"**,** --4. B CICLO

**begin**

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

**end** Behavioral**;**

**Simulación:**



**Tabla de resultados de la ejecución:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bus | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 |
| PC | 0 | 1 | 2 | 3 | 4 | 2 | 3 | 4 | 2 | 3 | 4 |
| Instrucción |  |  |  |  |  |  |  |  |  |  |  |
| ReadData1 | 0 | 1 | 7 | 1 | 1 | 8 | 1 | 1 | 9 | 1 | 1 |
| ReadData2 | 0 | 1 | 1 | 8 | 1 | 1 | 9 | 1 | 1 | a | 1 |
| ResALU | 0 | 1 | 8 | 0 | 1 | 9 | 1 | 1 | a | 0 | 1 |
| BusSR | 0 | 0 | 8 | 0 | 0 | 9 | 8 | 0 | a | 9 | 0 |

**Código para llenar a la memoria con el problema del punto 5:**

**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 **:=** 10**);**

**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"**;**--Operaci󮠔ipo 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 **:=** **(**

--ARREGLO 10 LOCALIDADES IMPARES INICIANDO CON 5

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

LI**&**R1**&**x"0005"**,** --1. LI R1 #5

LI**&**R2**&**x"000f"**,** --2. LI R2 #15

SW**&**R1**&**R0**&**X"042"**,** --3. AQUI: SW R1 R0 42

ADDI**&**R0**&**R0**&**X"001"**,** --4. ADDI R0 R0 1

ADDI**&**R1**&**R1**&**X"002"**,** --5. ADDI R1 R1 2

BLTI**&**R2**&**R0**&**X"ffd"**,** --6. BLTI R2, R0, -3 (1111 1111 1101)

NOP**&**X"00000"**,** --7. NOP

B**&**SU**&**x"0007"**,** --8. B NOP

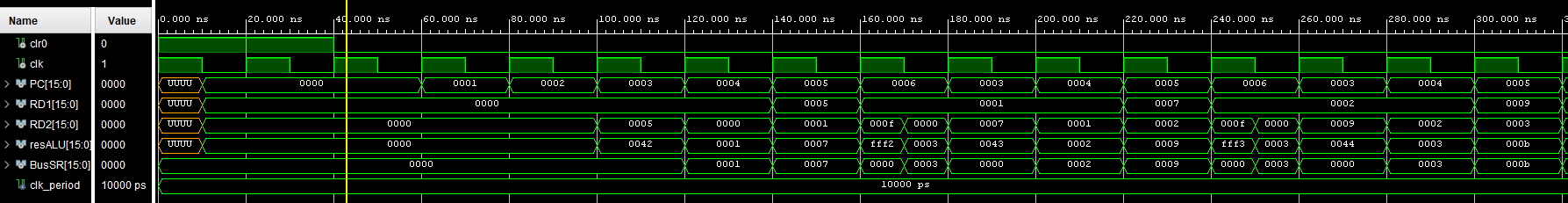
**others** **=>** **(others** **=>** '0'**));**

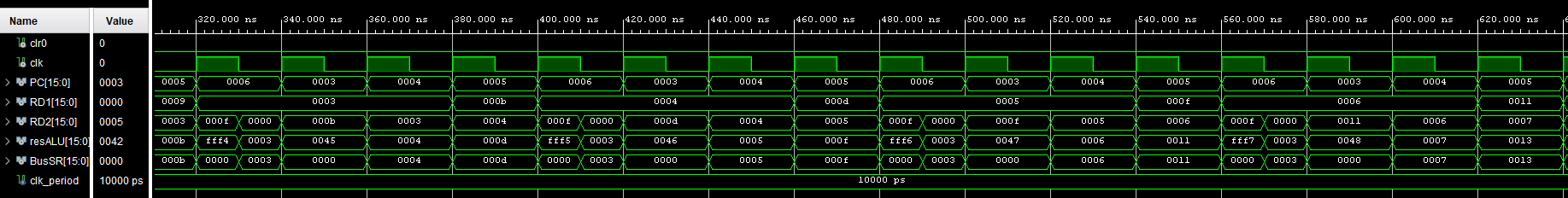
**begin**

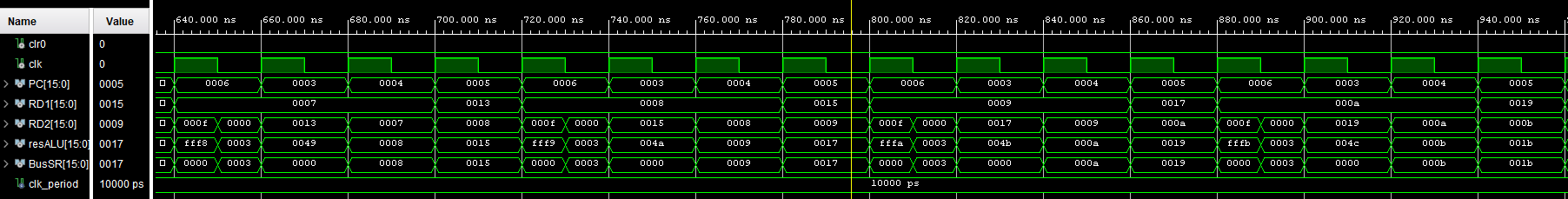
Inst **<=** memProg**(**conv\_integer**(**PC**));**

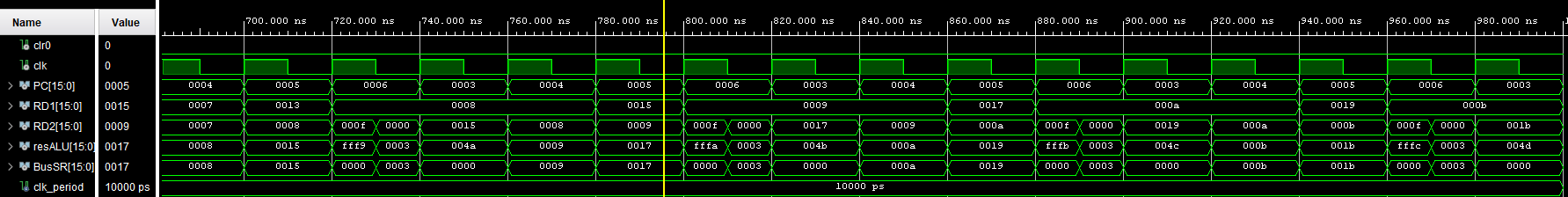
**end** Behavioral**;**

**Simulación:**









**Tabla de resultados de la ejecución:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bus | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 |
| PC | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 3 | 4 | 5 | 6 |
| Instrucción |  |  |  |  |  |  |  |  |  |  |  |
| ReadData1 | 0 | 0 | 0 | 0 | 0 | 5 | 1 | 1 | 1 | 7 | 2 |
| ReadData2 | 0 | 0 | 0 | 5 | 0 | 1 | f | 7 | 1 | 2 | f |
| ResALU | 0 | 0 | 0 | 42 | 1 | 7 | fff2 | 43 | 2 | 9 | fff3 |
| BusSR | 0 | 0 | 0 | 0 | 1 | 7 | 0 | 0 | 2 | 9 | 0 |