*****Instituto Politécnico Nacional***

***Escuela Superior de Cómputo***

*Arquitectura de Computadoras*

***Practica 14: Unidad de Control***

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

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

***Profesor:*** *Nayeli Vega García*

***Fecha de entrega:*** *28 de junio del 2020*

**Elementos de la Arquitectura**

**Código de Implementación**

* **MfunCode**

**library** IEEE**;**

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

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

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

**entity** MfunCode **is**

**generic** **(**

D **:** integer **:=** 20**;**

A **:** integer **:=** 4**);**

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

salidaD **:** **out** STD\_LOGIC\_VECTOR **(**D**-**1 **downto** 0**));**

**end** MfunCode**;**

**architecture** Behavioral **of** MfunCode **is**

**constant** ADD**:** std\_logic\_vector **:=** X"04433"**;**

**constant** SUB**:** std\_logic\_vector **:=** X"04473"**;**

**constant** OpAND**:** std\_logic\_vector **:=** X"04403"**;**

**constant** OpOR**:** std\_logic\_vector **:=** X"04413"**;**

**constant** OpXOR**:** std\_logic\_vector **:=** X"04423"**;**

**constant** OpNAND**:** std\_logic\_vector **:=** X"044d3"**;**

**constant** OpNOR**:** std\_logic\_vector **:=** X"044c3"**;**

**constant** OpXNOR**:** std\_logic\_vector **:=** X"044a3"**;**

**constant** OpNOT**:** std\_logic\_vector **:=** X"044d3"**;**

**constant** OpSLL**:** std\_logic\_vector **:=** X"01400"**;**

**constant** OpSRL**:** std\_logic\_vector **:=** X"01c00"**;**

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

**constant** memoria **:** banco **:=** **(**

ADD**,** --00

SUB**,** --01

OpAND**,** --02

OpOR**,** --03

OpXOR**,** --04

OpNAND**,** --05

OpNOR**,** --06

OpXNOR**,** --07

OpNOT**,** --08

OpSLL**,** --09

OpSRL**,** --10

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

**begin**

salidaD **<=** memoria**(**conv\_integer**(**funCode**));**

**end** Behavioral**;**

* **MopCode**

**library** IEEE**;**

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

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

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

**entity** MopCode **is**

**generic** **(**

D **:** integer **:=** 20**;**

A **:** integer **:=** 5**);**

**Port** **(** opCode **:** **in** STD\_LOGIC\_VECTOR **(**A**-**1 **downto** 0**);**

salidaD **:** **out** STD\_LOGIC\_VECTOR **(**D**-**1 **downto** 0**));**

**end** MopCode**;**

**architecture** Behavioral **of** MOpcode **is**

--OPCODES--

--Tipo I

**constant** LI**:** std\_logic\_vector **:=** x"00400"**;**

**constant** LWI**:** std\_logic\_vector **:=** x"04408"**;**

**constant** LW**:** std\_logic\_vector **:=** x"06531"**;**

**constant** SWI**:** std\_logic\_vector **:=** x"0800c"**;**

**constant** SW**:** std\_logic\_vector **:=** x"0A135"**;**

**constant** ADDI**:** std\_logic\_vector **:=** x"04533"**;**

**constant** SUBI**:** std\_logic\_vector **:=** x"04573"**;**

**constant** ANDI**:** std\_logic\_vector **:=** x"04503"**;**

**constant** ORI**:** std\_logic\_vector **:=** x"04513"**;**

**constant** XORI**:** std\_logic\_vector **:=** x"04523"**;**

**constant** NANDI**:** std\_logic\_vector **:=** x"045d3"**;**

**constant** NORI**:** std\_logic\_vector **:=** x"045c3"**;**

**constant** XNORI**:** std\_logic\_vector **:=** x"045a3"**;**

**constant** BEQI**:** std\_logic\_vector **:=** x"08071"**;**

**constant** BNEI**:** std\_logic\_vector **:=** x"08071"**;**

**constant** BLTI**:** std\_logic\_vector **:=** x"08071"**;**

**constant** BLETI**:** std\_logic\_vector **:=** x"08071"**;**

**constant** BGTI**:** std\_logic\_vector **:=** x"08071"**;**

**constant** BGETI**:** std\_logic\_vector **:=** x"08071"**;**

**constant** SALTO**:** std\_logic\_vector **:=** x"98333"**;**--Saltos condicionales de los B's tipo I

--Tipo J

**constant** B**:** std\_logic\_vector **:=** x"10000"**;**

**constant** CALL**:** std\_logic\_vector **:=** x"50000"**;**

--Otras Instrucciones

**constant** RET**:** std\_logic\_vector **:=** x"20000"**;**

**constant** NOP**:** std\_logic\_vector **:=** x"00000"**;**

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

**constant** memoria**:** banco **:=** **(**

SALTO**,** --00

LI**,** --01

LWI**,** --02

SWI**,** --03

SW**,** --04

ADDI**,** --05

SUBI**,** --06

ANDI**,** --07

ORI**,** --08

XORI**,** --09

NANDI**,** --10

NORI**,** --11

XNORI**,** --12

BEQI**,** --13

BNEI**,** --14

BLTI**,** --15

BLETI**,** --16

BGTI**,** --17

BGETI**,** --18

B**,** --19

CALL**,** --20

RET**,** --21

NOP**,** --22

LW**,** --23

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

**begin**

salidaD **<=** memoria**(**conv\_integer**(**opCode**));**

**end** Behavioral**;**

* **Mux\_SM**

**library** IEEE**;**

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

**entity** Mux\_SM **is**

**Port** **(** MfunCode**,** MopCode **:** **in** STD\_LOGIC\_VECTOR **(**19 **downto** 0**);**

SM **:** **in** STD\_LOGIC**;**

Microinstruccion **:** **out** STD\_LOGIC\_VECTOR **(**19 **downto** 0**));**

**end** Mux\_SM**;**

**architecture** Behavioral **of** Mux\_SM **is**

**begin**

Microinstruccion **<=** MopCode **when** SM **=** '1' **else** MfunCode**;**

**end** Behavioral**;**

* **Mux\_SODPC**

**library** IEEE**;**

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

**entity** Mux\_SODPC **is**

**Port** **(** opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

SDOPC **:** **in** STD\_LOGIC**;**

salida **:** **out** STD\_LOGIC\_VECTOR **(**4 **downto** 0**));**

**end** Mux\_SODPC**;**

**architecture** Behavioral **of** Mux\_SODPC **is**

**begin**

salida **<=** opCode **when** SDOPC **=** '1' **else** "00000"**;**

**end** Behavioral**;**

* **Decodificador\_Instruccion**

**library** IEEE**;**

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

**entity** Decodificador\_Instruccion **is**

**Port** **(** opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

TIPOR**,** BEQI**,** BNEQI**,** BLTI**,** BLETI**,** BGTI**,** BGETI **:** **out** STD\_LOGIC**);**

**end** Decodificador\_Instruccion**;**

**architecture** Behavioral **of** Decodificador\_Instruccion **is**

**begin**

TIPOR **<=** '1' **when** opCode **=** "00000" **else** '0'**;**

BEQI **<=** '1' **when** opCode **=** "01101" **else** '0'**;**

BNEQI **<=** '1' **when** opCode **=** "01110" **else** '0'**;**

BLTI **<=** '1' **when** opCode **=** "01111" **else** '0'**;**

BLETI **<=** '1' **when** opCode **=** "10000" **else** '0'**;**

BGTI **<=** '1' **when** opCode **=** "10001" **else** '0'**;**

BGETI **<=** '1' **when** opCode **=** "10010" **else** '0'**;**

**end** Behavioral**;**

* **Nivel**

**library** IEEE**;**

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

**entity** Nivel **is**

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

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

**end** Nivel**;**

**architecture** Behavioral **of** nivel **is**

**signal** A**:** std\_logic **:=** '0'**;**

**signal** B**:** std\_logic **:=** '0'**;**

**begin**

**process** **(**clk**)** **begin**

**if** **(**clk'**event** and clk **=** '1'**)** **then** --rising\_edge(clk)

A **<=** NOT A**;**

**end** **if;**

**if** **(**clk'**event** and clk **=** '0'**)** **then** --falling\_edge(clk)

B **<=** NOT B**;**

**end** **if;**

**end** **process;**

NA **<=** '0' **when** clr**=**'1' **else** **(**A XOR B**);**

**end** Behavioral**;**

* **Regitro\_Estado**

**library** IEEE**;**

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

**entity** Registro\_Estado **is**

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

banderas **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

Q **:** **out** STD\_LOGIC\_VECTOR **(**3 **downto** 0**));**

**end** Registro\_Estado**;**

**architecture** Behavioral **of** Registro\_Estado **is**

**begin**

**process(**clk**,** clr**)** **begin**

**if** clr **=** '1' **then**

Q **<=** "0000"**;**

**elsif** **(**clk'**event** and clk**=**'0'**)** **then**

**if** LF **=** '1' **then**

Q **<=** banderas**;**

**end** **if;**

**end** **if;**

**end** **process;**

**end** Behavioral**;**

* **Condicion**

**library** IEEE**;**

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

**entity** Condicion **is**

**Port** **(** Q **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

EQ**,** NEQ**,** LT**,** LE**,** GTI**,** GET **:** **out** STD\_LOGIC**);**

**end** Condicion**;**

**architecture** Behavioral **of** Condicion **is**

**begin**

--(0, N, Z, C)

EQ **<=** '1' **when** Q**=** "0010" **else** '0'**;**--Z

NEQ **<=** not Q**(**1**);**--not(Z)

LT **<=** Q**(**2**);**--not(C)

LE **<=(**Q**(**2**)** or Q**(**1**));**--Z + not(C)

GTI **<=** not Q**(**2**);**--not(Z) and C

GET **<=((**not Q**(**2**))** or Q**(**1**));**--C

**end** Behavioral**;**

* **Carta\_ASM**

**library** IEEE**;**

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

**entity** Carta\_ASM **is**

**Port** **(** TIPOR**,** BEQ**,** BNEQ**,** BLT**,** BLE**,** BGT**,** BGET**,** NA **:** **in** STD\_LOGIC**;**

EQ**,** NEQ**,** LT**,** LE**,** GTI**,** GET **:** **in** STD\_LOGIC**;**

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

SDOPC**,** SM **:** **out** STD\_LOGIC**);**

**end** Carta\_ASM**;**

**architecture** Behavioral **of** Carta\_ASM **is**

**begin**

estados**:** **process** **(**clk**,** clr**,** TIPOR**,** BEQ**,** BNEQ**,** BLT**,** BLE**,** BGT**,** BGET**,** NA**,** EQ**,** NEQ**,** LT**,** LE**,** GTI**,** GET**)**

**begin**

**if** clr **=** '1' **then**

SDOPC **<=** '0'**;**

SM **<=** '0'**;**

**elsif** **rising\_edge(**clk**)** **then**

**if** TIPOR **=** '1' **then**

SDOPC **<=** '0'**;**

SM **<=** '0'**;**

**else**

**if** BEQ **=** '1' **then** --BEQ

**if** NA **=** '1' **then** --VERIFICACION

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**elsif** EQ **=** '1' **then** --SALTO

SDOPC **<=** '1'**;**

SM **<=** '1'**;**

**else**

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**end** **if;**

**elsif** BNEQ **=** '1' **then**--BNEQ

**if** NA **=** '1' **then** --VERIFICACION

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**elsif** NEQ **=** '1' **THEN** --SALTO

SDOPC **<=** '1'**;**

SM **<=** '1'**;**

**else**

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**end** **if;**

**elsif** BLT **=** '1' **then** --BLT

**if** NA **=** '1' **then** --VERIFICACION

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**elsif** LT **=** '1' **then** --SALTO

SDOPC **<=** '1'**;**

SM **<=** '1'**;**

**else**

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**end** **if;**

**elsif** BLE **=** '1' **then** --BLE

**if** NA **=** '1' **then** --VERIFICACION

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**elsif** LE **=** '1' **then** --SALTO

SDOPC **<=** '1'**;**

SM **<=** '1'**;**

**else**

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**end** **if;**

**elsif** BGT **=** '1' **then** --BGT

**if** NA **=** '1' **then** --VERIFICACION

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**elsif** GTI **=** '1' **then** --SALTO

SDOPC **<=** '1'**;**

SM **<=** '1'**;**

**else**

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**end** **if;**

**elsif** BGET **=** '1' **then** --BGET

**if** NA **=** '1' **then** --VERIFICACION

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**elsif** GET **=** '1' **then** --SALTO

SDOPC **<=** '1'**;**

SM **<=** '1'**;**

**else**

SDOPC **<=** '0'**;**

SM **<=** '1'**;**

**end** **if;**

**end** **if;**

**end** **if;**

**end** **if;**

**end** **process;**

**end** Behavioral**;**

**Código de Simulación**

* **MfunCode**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.ALL;**

**entity** MfunCode\_tb **is**

**end** MfunCode\_tb**;**

**architecture** Behavioral **of** MfunCode\_tb **is**

**component** MfunCode **is**

**Port** **(** funCode **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

salidaD **:** **out** STD\_LOGIC\_VECTOR **(**19 **downto** 0**));**

**end** **component;**

--Inputs

**signal** funCode **:** STD\_LOGIC\_VECTOR**(**3 **downto** 0**)** **:=** **(others** **=>** '0'**);**

--Outputs

**signal** salidaD **:** STD\_LOGIC\_VECTOR**(**19 **downto** 0**);**

**begin**

-- Instantiate the Unit Under Test (UUT)

uut**:** MfunCode

**Port** **Map** **(** funCode **=>** funCode**,**

salidaD **=>** salidaD**);**

-- Stimulus process

stim\_proc**:** **process**

**begin**

funCode **<=** x"0"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"1"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"2"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"3"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"4"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"5"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"6"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"7"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"8"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"9"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"A"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"B"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"C"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"D"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"E"**;**

**wait** **for** 50 ns**;**

funCode **<=** x"F"**;**

**wait** **for** 50 ns**;**

**wait;**

**end** **process;**

**end** Behavioral**;**

* **MopCode**

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.ALL;**

**entity** MopCode\_tb **is**

**end** MopCode\_tb**;**

**architecture** Behavioral **of** MopCode\_tb **is**

**component** MopCode **is**

**Port** **(** opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

salidaD **:** **out** STD\_LOGIC\_VECTOR **(**19 **downto** 0**));**

**end** **component;**

--Inputs

**signal** opCode **:** STD\_LOGIC\_VECTOR**(**4 **downto** 0**)** **:=** **(others** **=>** '0'**);**

--Outputs

**signal** salidaD **:** STD\_LOGIC\_VECTOR**(**19 **downto** 0**);**

**begin**

-- Instantiate the Unit Under Test (UUT)

uut**:** MopCode

**Port** **Map** **(** opCode **=>** opCode**,**

salidaD **=>** salidaD**);**

-- Stimulus process

stim\_proc**:** **process**

**begin**

opCode **<=** "00001"**;**--LI

**wait** **for** 50 ns**;**

opCode **<=** "00011"**;**--SWI

**wait** **for** 50 ns**;**

opCode **<=** "00101"**;**--ADDI

**wait** **for** 50 ns**;**

opCode **<=** "00110"**;**--SUBI

**wait** **for** 50 ns**;**

opCode **<=** "01001"**;**--XORI

**wait** **for** 50 ns**;**

opCode **<=** "01100"**;**--XNORI

**wait** **for** 50 ns**;**

opCode **<=** "01111"**;**--BLTI

**wait** **for** 50 ns**;**

opCode **<=** "10010"**;**--BGETI

**wait** **for** 50 ns**;**

opCode **<=** "10011"**;**--B

**wait** **for** 50 ns**;**

opCode **<=** "10100"**;**--CALL

**wait** **for** 50 ns**;**

opCode **<=** "10101"**;**--RET

**wait** **for** 50 ns**;**

opCode **<=** "10110"**;**--NOP

**wait** **for** 50 ns**;**

**wait;**

**end** **process;**

**end** Behavioral**;**

* **Decodificador\_Instruccion**

**library** IEEE**;**

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

**entity** Decodificador\_Instruccion\_tb **is**

**end** Decodificador\_Instruccion\_tb**;**

**architecture** Behavioral **of** Decodificador\_Instruccion\_tb **is**

**component** Decodificador\_Instruccion **is**

**Port** **(** opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

TIPOR**,** BEQI**,** BNEQI**,** BLTI**,** BLETI**,** BGTI**,** BGETI **:** **out** STD\_LOGIC**);**

**end** **component;**

--Inputs

**signal** opCode **:**STD\_LOGIC\_VECTOR**(**4 **downto** 0**)** **:=** **(others** **=>** '0'**);**

--Outputs

**signal** TIPOR **:** STD\_LOGIC**;**

**signal** BEQI **:** STD\_LOGIC**;**

**signal** BNEQI **:** STD\_LOGIC**;**

**signal** BLTI **:** STD\_LOGIC**;**

**signal** BLETI **:** STD\_LOGIC**;**

**signal** BGTI **:** STD\_LOGIC**;**

**signal** BGETI **:** STD\_LOGIC**;**

**begin**

uut**:** Decodificador\_Instruccion

**Port** **Map** **(** opCode **=>** opCode**,**

TIPOR **=>** TIPOR**,**

BEQI **=>** BEQI**,**

BNEQI **=>** BNEQI**,**

BLTI **=>** BLTI**,**

BLETI **=>** BLETI**,**

BGTI **=>** BGTI**,**

BGETI **=>** BGETI**);**

-- Stimulus process

stim\_proc**:** **process**

**begin**

opCode **<=** "00001"**;**--LI

**wait** **for** 50 ns**;**

opCode **<=** "00000"**;**--Tipo R

**wait** **for** 50 ns**;**

opCode **<=** "01101"**;**--BEQI

**wait** **for** 50 ns**;**

opCode **<=** "01110"**;**--BNEI

**wait** **for** 50 ns**;**

opCode **<=** "01111"**;**--BLTI

**wait** **for** 50 ns**;**

opCode **<=** "10000"**;**--BLETI

**wait** **for** 50 ns**;**

opCode **<=** "10001"**;**--BGTI

**wait** **for** 50 ns**;**

opCode **<=** "10010"**;**--BGETI

**wait** **for** 50 ns**;**

**wait;**

**end** **process;**

**end** Behavioral**;**

* **Nivel**

**library** IEEE**;**

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

**entity** Nivel\_tb **is**

**end** Nivel\_tb**;**

**architecture** Behavioral **of** Nivel\_tb **is**

**component** Nivel **is**

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

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

**end** **component;**

--Inputs

**signal** clk **:** STD\_LOGIC **:=** '0'**;**

**signal** clr **:** STD\_LOGIC **:=** '0'**;**

--Outputs

**signal** NA **:** STD\_LOGIC**;**

-- Clock period definitions

**constant** clk\_period **:** time **:=** 10 ns**;**

**begin**

-- Instantiate the Unit Under Test (UUT)

uut**:** Nivel

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

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

NA **=>** NA**);**

-- Clock process definitions

clk\_process **:process**

**begin**

clk **<=** '0'**;**

**wait** **for** clk\_period**/**2**;**

clk **<=** '1'**;**

**wait** **for** clk\_period**/**2**;**

**end** **process;**

-- Stimulus process

stim\_proc**:** **process**

**begin**

**wait** **for** 20 ns**;**

clr **<=** '1'**;**

**wait** **for** 20 ns**;**

clr **<=** '0'**;**

**wait** **for** 40 ns**;**

**wait** **for** 20 ns**;**

clr **<=** '1'**;**

**wait** **for** 25 ns**;**

clr **<=** '0'**;**

**wait** **for** 40 ns**;**

**wait;**

**end** **process;**

**end** Behavioral**;**

* **Regitro\_Estado**

**library** IEEE**;**

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

**entity** Registro\_Estado\_tb **is**

**end** Registro\_Estado\_tb**;**

**architecture** Behavioral **of** Registro\_Estado\_tb **is**

**component** Registro\_Estado **is**

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

banderas **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

Q **:** **out** STD\_LOGIC\_VECTOR **(**3 **downto** 0**));**

**end** **component;**

--Inputs

**signal** clk **:** STD\_LOGIC **:=** '0'**;**

**signal** clr **:** STD\_LOGIC **:=** '0'**;**

**signal** LF **:** STD\_LOGIC **:=** '0'**;**

**signal** banderas **:** STD\_LOGIC\_VECTOR**(**3 **downto** 0**)** **:=** **(others** **=>** '0'**);**

--Outputs

**signal** Q **:** STD\_LOGIC\_VECTOR**(**3 **downto** 0**);**

-- Clock period definitions

**constant** CLK\_period **:** time **:=** 10 ns**;**

**begin**

-- Instantiate the Unit Under Test (UUT)

uut**:** Registro\_Estado

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

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

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

banderas **=>** banderas**,**

Q **=>** Q**);**

-- Clock process definitions

CLK\_process **:process**

**begin**

clk **<=** '0'**;**

**wait** **for** CLK\_period**/**2**;**

clk **<=** '1'**;**

**wait** **for** CLK\_period**/**2**;**

**end** **process;**

-- Stimulus process

stim\_proc**:** **process**

**begin**

clr **<=** '1'**;**

**wait** **for** 20 ns**;**

clr **<=** '0'**;**

banderas **<=** "0110"**;**

LF **<=** '1'**;**

**wait** **for** 50 ns**;**

banderas **<=** "1010"**;**

LF **<=** '0'**;**

**wait** **for** 50 ns**;**

LF **<=** '1'**;**

**wait** **for** 50 ns**;**

banderas **<=** "1001"**;**

LF **<=** '0'**;**

**wait** **for** 50 ns**;**

LF **<=** '1'**;**

**wait** **for** 50 ns**;**

**wait;**

**end** **process;**

**end** Behavioral**;**

* **Condicion**

**library** IEEE**;**

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

**entity** Condicion\_tb **IS**

**end** Condicion\_tb**;**

**architecture** Behavioral **of** Condicion\_tb **is**

**component** Condicion **is**

**Port** **(** Q **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

EQ**,** NEQ**,** LT**,** LE**,** GTI**,** GET **:** **out** STD\_LOGIC**);**

**end** **component;**

--Inputs

**signal** Q **:** STD\_LOGIC\_VECTOR **(**3 **downto** 0**)** **:=** **(others** **=>** '0'**);**

--Outputs

**signal** EQ **:** STD\_LOGIC**;**

**signal** NEQ **:** STD\_LOGIC**;**

**signal** LT **:** STD\_LOGIC**;**

**signal** LE **:** STD\_LOGIC**;**

**signal** GTI **:** STD\_LOGIC**;**

**signal** GET **:** STD\_LOGIC**;**

**begin**

-- Instantiate the Unit Under Test (UUT)

uut**:** Condicion

**Port** **Map** **(** Q **=>** Q**,**

EQ **=>** EQ**,**

NEQ **=>** NEQ**,**

LT **=>** LT**,**

LE **=>** LE**,**

GTI **=>** GTI**,**

GET **=>** GET**);**

-- Stimulus process

stim\_proc**:** **process**

**begin**

Q **<=** "0000"**;**

**wait** **for** 50 ns**;**

Q **<=** "0010"**;**

**wait** **for** 50 ns**;**

Q **<=** "0100"**;**

**wait** **for** 50 ns**;**

Q **<=** "1000"**;**

**wait** **for** 50 ns**;**

Q **<=** "0110"**;**

**wait** **for** 50 ns**;**

Q **<=** "1001"**;**

**wait** **for** 50 ns**;**

Q **<=** "1010"**;**

**wait** **for** 50 ns**;**

**wait;**

**end** **process;**

**end** Behavioral**;**

* **Carta\_ASM**

**library** IEEE**;**

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

**entity** Carta\_ASM\_tb **is**

**end** Carta\_ASM\_tb**;**

**architecture** Behavioral **of** Carta\_ASM\_tb **is**

**component** Carta\_ASM **is**

**Port** **(** TIPOR**,** BEQ**,** BNEQ**,** BLT**,** BLE**,** BGT**,** BGET**,** NA **:** **in** STD\_LOGIC**;**

EQ**,** NEQ**,** LT**,** LE**,** GTI**,** GET **:** **in** STD\_LOGIC**;**

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

SDOPC**,** SM **:** **out** STD\_LOGIC**);**

**end** **component;**

--Inputs

**signal** TIPOR **:** STD\_LOGIC **:=** '0'**;**

**signal** BEQ **:** STD\_LOGIC **:=** '0'**;**

**signal** BNEQ **:** STD\_LOGIC **:=** '0'**;**

**signal** BLT **:** STD\_LOGIC **:=** '0'**;**

**signal** BLE **:** STD\_LOGIC **:=** '0'**;**

**signal** BGT **:** STD\_LOGIC **:=** '0'**;**

**signal** BGET **:** STD\_LOGIC **:=** '0'**;**

**signal** NA **:** STD\_LOGIC **:=** '0'**;**

**signal** EQ **:** STD\_LOGIC **:=** '0'**;**

**signal** NEQ **:** STD\_LOGIC **:=** '0'**;**

**signal** LT **:** STD\_LOGIC **:=** '0'**;**

**signal** LE **:** STD\_LOGIC **:=** '0'**;**

**signal** GTI **:** STD\_LOGIC **:=** '0'**;**

**signal** GET **:** STD\_LOGIC **:=** '0'**;**

**signal** clk **:** STD\_LOGIC **:=** '0'**;**

**signal** clr **:** STD\_LOGIC **:=** '0'**;**

--Outputs

**signal** SDOPC **:** STD\_LOGIC**;**

**signal** SM **:** STD\_LOGIC**;**

-- Clock period definitions

**constant** CLK\_period **:** time **:=** 10 ns**;**

**BEGIN**

-- Instantiate the Unit Under Test (UUT)

uut**:** Carta\_ASM

**Port** **Map** **(** TIPOR **=>** TIPOR**,**

BEQ **=>** BEQ**,**

BNEQ **=>** BNEQ**,**

BLT **=>** BLT**,**

BLE **=>** BLE**,**

BGT **=>** BGT**,**

BGET **=>** BGET**,**

NA **=>** NA**,**

EQ **=>** EQ**,**

NEQ **=>** NEQ**,**

LT **=>** LT**,**

LE **=>** LE**,**

GTI **=>** GTI**,**

GET **=>** GET**,**

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

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

SDOPC **=>** SDOPC**,**

SM **=>** SM**);**

-- Clock process definitions

CLK\_process **:process**

**begin**

clk **<=** '0'**;**

**wait** **for** CLK\_period**/**2**;**

clk **<=** '1'**;**

**wait** **for** CLK\_period**/**2**;**

**end** **process;**

stim\_proc**:** **process**

**begin**

clr **<=** '1'**;**

**wait** **for** 20 ns**;**

clr **<=** '0'**;**

BEQ **<=** '1'**;**

EQ **<=** '1'**;**

**wait** **for** CLK\_period**\***5**;**

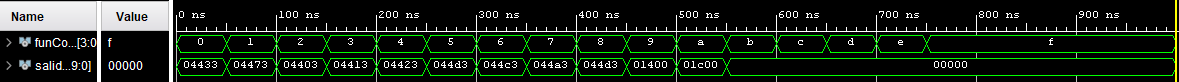
**wait;**

**end** **process;**

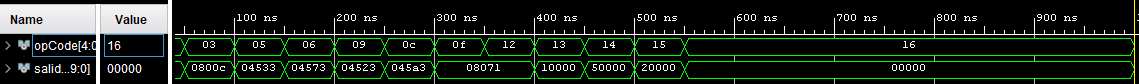
**end** Behavioral**;**

**Simulación:**

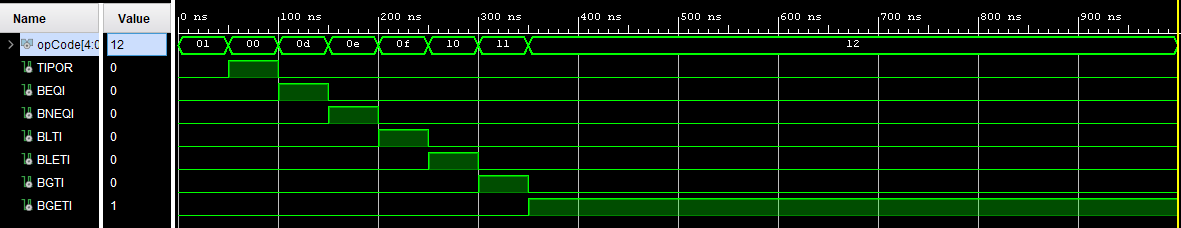
* **MfunCode**



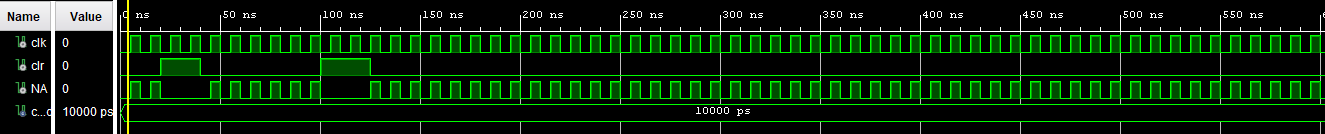
* **MopCode**

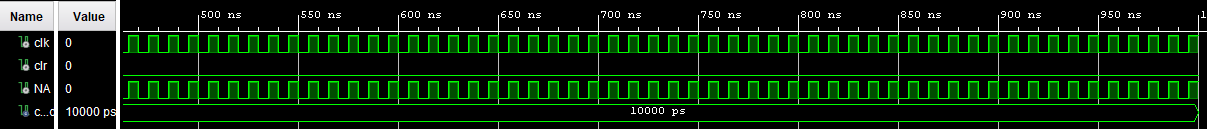


* **Decodificador\_Instruccion**

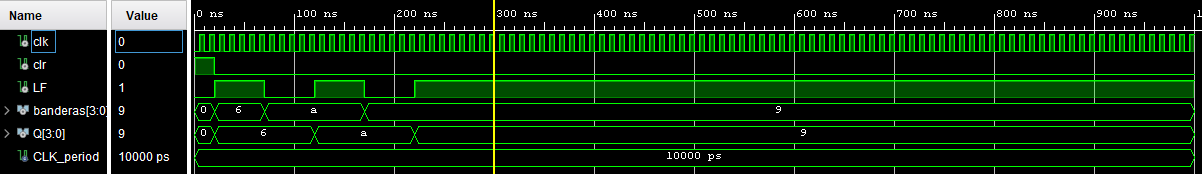


* **Nivel**

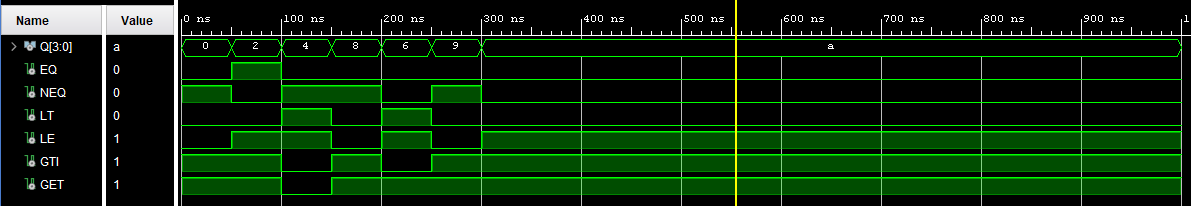




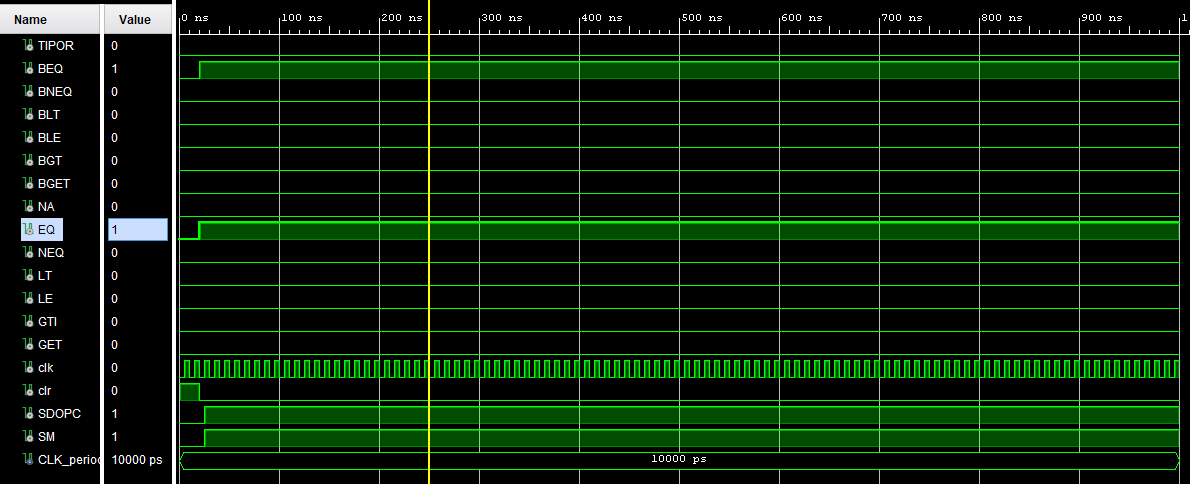
* **Regitro\_Estado**



* **Condición**

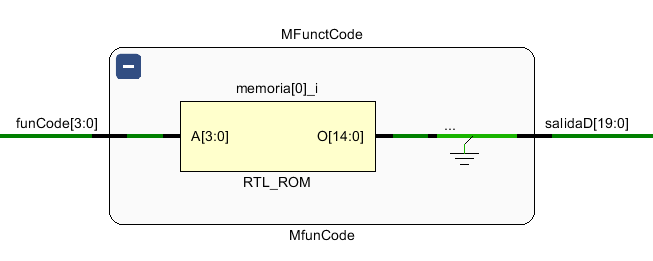


* **Carta\_ASM**

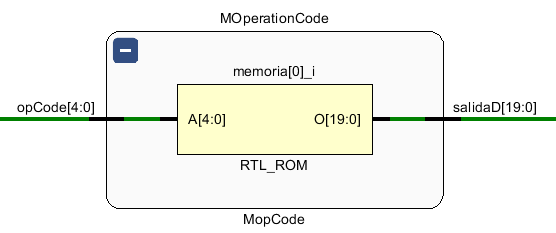


**Diagrama RTL:**

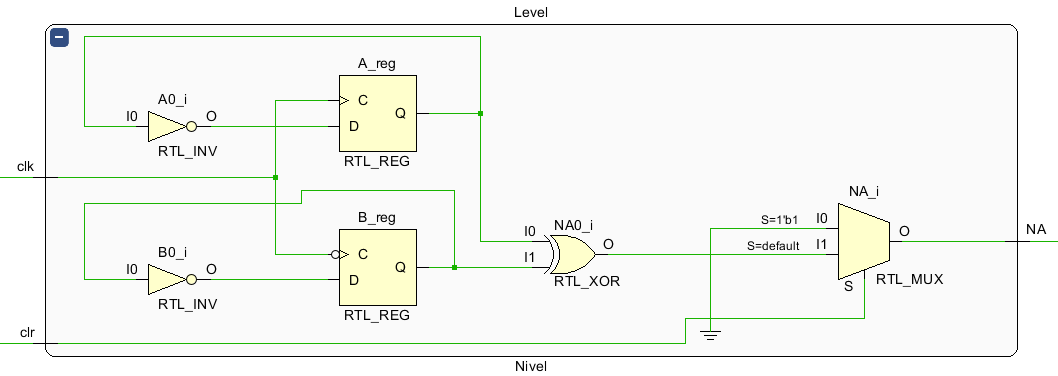
* **MfunCode**



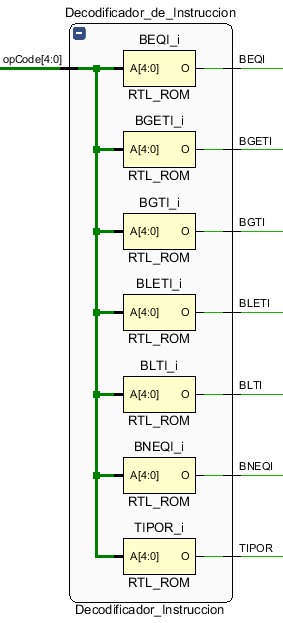
* **MopCode**



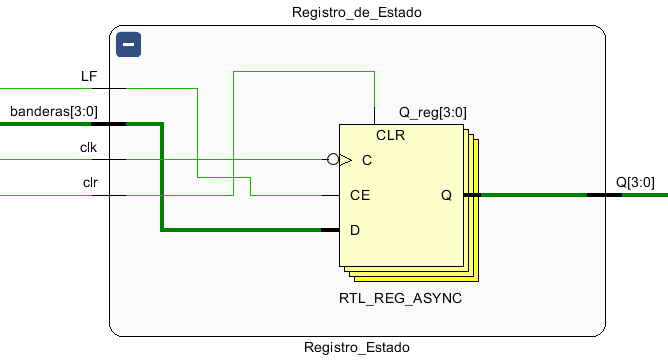
* **Nivel**



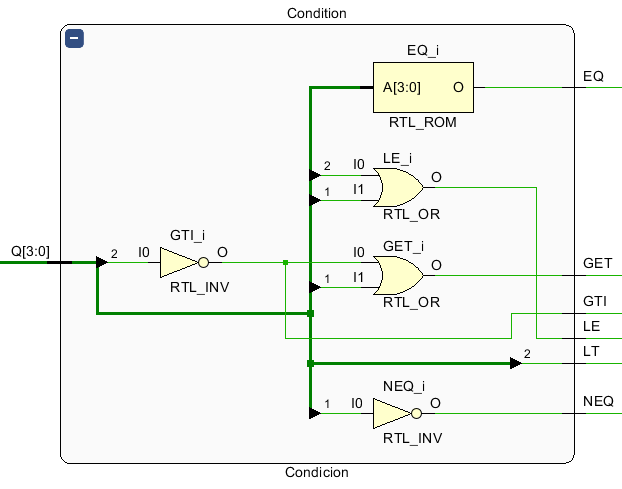
* **Decodificador\_Instruccion**

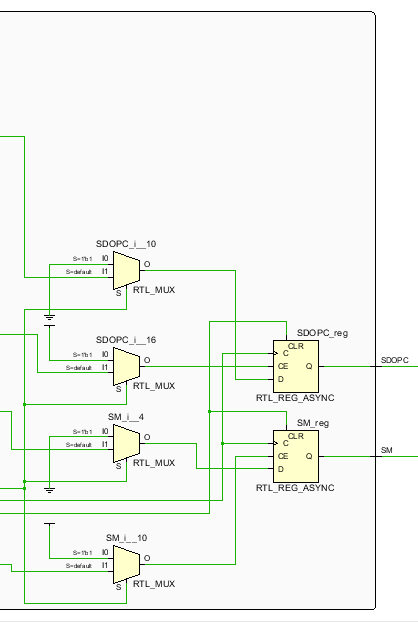
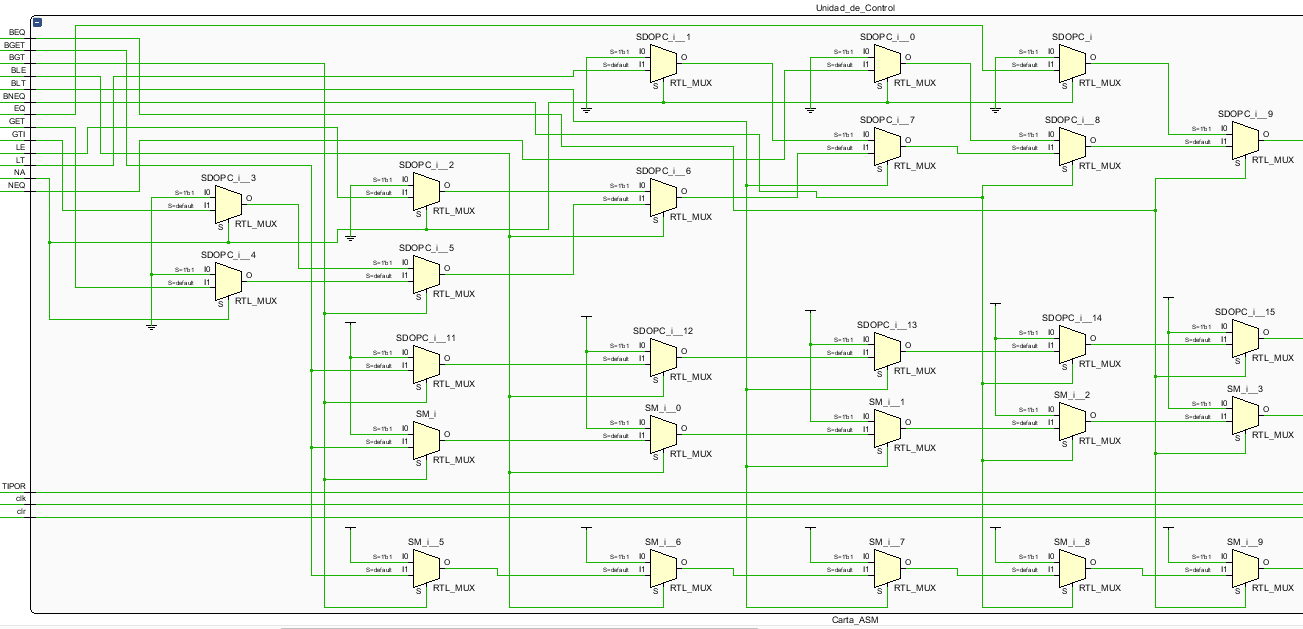


* **Regitro\_Estado**



* **Condicion**



* **Carta\_ASM**

**Arquitectura Completa**

**Código de Implementación**

**library** IEEE**;**

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

**entity** Unidad\_Control **is**

**Port** **(** funCode**,** banderas **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

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

Microinstruccion **:** **out** STD\_LOGIC\_VECTOR **(**19 **downto** 0**);**

LVL**:** **OUT** STD\_LOGIC**);**

**end** Unidad\_Control**;**

**architecture** Behavioral **of** Unidad\_Control **is**

**component** Carta\_ASM **is**

**Port** **(** TIPOR**,** BEQ**,** BNEQ**,** BLT**,** BLE**,** BGT**,** BGET**,** NA **:** **in** STD\_LOGIC**;**

EQ**,** NEQ**,** LT**,** LE**,** GTI**,** GET **:** **in** STD\_LOGIC**;**

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

SDOPC**,** SM **:** **out** STD\_LOGIC**);**

**end** **component;**

**component** MfunCode **is**

**Port** **(** funCode **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

salidaD **:** **out** STD\_LOGIC\_VECTOR **(**19 **downto** 0**));**

**end** **component;**

**component** MopCode **is**

**Port** **(** opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

salidaD **:** **out** STD\_LOGIC\_VECTOR **(**19 **downto** 0**));**

**end** **component;**

**component** Mux\_SODPC **is**

**Port** **(** opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

SDOPC **:** **in** STD\_LOGIC**;**

salida **:** **out** STD\_LOGIC\_VECTOR **(**4 **downto** 0**));**

**end** **component;**

**component** Mux\_SM **is**

**Port** **(** MfunCode**,** MopCode **:** **in** STD\_LOGIC\_VECTOR **(**19 **downto** 0**);**

SM **:** **in** STD\_LOGIC**;**

Microinstruccion **:** **out** STD\_LOGIC\_VECTOR **(**19 **downto** 0**));**

**end** **component;**

**component** Decodificador\_Instruccion **is**

**Port** **(** opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

TIPOR**,** BEQI**,** BNEQI**,** BLTI**,** BLETI**,** BGTI**,** BGETI **:** **out** STD\_LOGIC**);**

**end** **component;**

**component** Nivel **is**

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

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

**end** **component;**

**component** Registro\_Estado **is**

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

banderas **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

Q **:** **out** STD\_LOGIC\_VECTOR **(**3 **downto** 0**));**

**end** **component;**

**component** Condicion **is**

**Port** **(** Q **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

EQ**,** NEQ**,** LT**,** LE**,** GTI**,** GET **:** **out** STD\_LOGIC**);**

**end** **component;**

**signal** OpR**,** OpNoR**:** STD\_LOGIC\_VECTOR**(**19 **DOWNTO** 0**);**

**signal** SM\_Mux**:** STD\_LOGIC\_VECTOR**(**4 **DOWNTO** 0**);**

**signal** Q**:** STD\_LOGIC\_VECTOR**(**3 **DOWNTO** 0**);**

**signal** SDOPC**,** NA**,** SM**:** STD\_LOGIC**;**

**signal** TIPOR**,** BEQI**,** BNEQI**,** BLTI**,** BLETI**,**BGTI**,**BGETI**:** STD\_LOGIC**;**

**signal** EQ**,** NEQ**,** LT**,** LE**,** GTI**,** GET **:** STD\_LOGIC**;**

**begin**

MFunctCode**:** MFunCode

**Port** **map** **(** funCode **=>** funCode**,**

salidaD **=>** OpR**);**

MOperationCode**:** MOpCode

**Port** **map** **(** opCode **=>** SM\_Mux**,**

salidaD **=>** OpNoR**);**

MuxSM**:** Mux\_SM

**Port** **map** **(** MfunCode **=>** OpR**,**

MopCode **=>** OpNoR**,**

SM **=>** SM**,**

Microinstruccion **=>** Microinstruccion**);**

MuxSODPC**:** Mux\_SODPC

**Port** **map** **(** opCode **=>** opCode**,**

SDOPC **=>** SDOPC**,**

salida **=>** SM\_Mux**);**

Decodificador\_de\_Instruccion**:** Decodificador\_Instruccion

**Port** **map(** opCode **=>** opCode**,**

TIPOR **=>** TIPOR**,**

BEQI **=>** BEQI**,**

BNEQI **=>** BNEQI**,**

BLTI **=>** BLTI**,**

BLETI **=>** BLETI**,**

BGTI **=>** BGTI**,**

BGETI **=>** BGETI**);**

Level**:** Nivel

**Port** **map** **(** clk **=>** clk**,**

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

NA **=>** NA**);**

Registro\_de\_Estado**:** Registro\_Estado

**Port** **map(** clk **=>** clk**,**

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

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

banderas **=>** banderas**,**

Q **=>** Q**);**

Condition**:** Condicion

**Port** **map** **(** Q **=>** Q**,**

EQ **=>** EQ**,**

NEQ **=>** NEQ**,**

LT **=>** LT**,**

LE **=>** LE**,**

GTI **=>** GTI**,**

GET **=>** GET**);**

Unidad\_de\_Control**:** Carta\_ASM

**Port** **map** **(** TIPOR **=>** TIPOR**,**

BEQ **=>** BEQI**,**

BNEQ **=>** BNEQI**,**

BLT **=>** BLTI**,**

BLE **=>** BLETI**,**

BGT **=>** BGTI**,**

BGET **=>** BGETI**,**

NA **=>** NA**,**

EQ **=>** EQ**,**

NEQ **=>** NEQ**,**

LT **=>** LT**,**

LE **=>** LE**,**

GTI **=>** GTI**,**

GET **=>** GET**,**

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

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

SDOPC **=>** SDOPC**,**

SM **=>** SM**);**

LVL **<=** NA**;**

**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** Unidad\_Control\_tb **is**

**end** Unidad\_Control\_tb**;**

**architecture** Behavioral **of** Unidad\_Control\_tb **is**

**component** Unidad\_Control **is**

**Port** **(** funCode**,** banderas **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

opCode **:** **in** STD\_LOGIC\_VECTOR **(**4 **downto** 0**);**

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

Microinstruccion **:** **out** STD\_LOGIC\_VECTOR **(**19 **downto** 0**);**

LVL**:** **OUT** STD\_LOGIC**);**

**end** **component;**

--Inputs

**signal** funCode **:** std\_logic\_vector**(**3 **downto** 0**)** **:=** **(others** **=>** '0'**);**

**signal** banderas **:** std\_logic\_vector**(**3 **downto** 0**)** **:=** **(others** **=>** '0'**);**

**signal** opCode **:** std\_logic\_vector**(**4 **downto** 0**)** **:=** **(others** **=>** '0'**);**

**signal** clk **:** std\_logic **:=** '0'**;**

**signal** clr **:** std\_logic **:=** '0'**;**

**signal** LF **:** std\_logic **:=** '0'**;**

--Outputs

**signal** Microinstruccion **:** std\_logic\_vector**(**19 **downto** 0**);**

**signal** LVL**:** STD\_LOGIC**;**

-- Clock period definitions

**constant** CLK\_period **:** time **:=** 10 ns**;**

**begin**

-- Instantiate the Unit Under Test (UUT)

uut**:** Unidad\_Control

**Port** **Map** **(** funCode **=>** funCode**,**

banderas **=>** banderas**,**

opCode **=>** opCode**,**

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

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

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

Microinstruccion **=>** Microinstruccion**,**

LVL **=>** LVL**);**

-- Clock process definitions

CLK\_process **:process**

**begin**

CLK **<=** '0'**;**

**wait** **for** CLK\_period**/**2**;**

CLK **<=** '1'**;**

**wait** **for** CLK\_period**/**2**;**

**end** **process;**

-- 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\_funCode**,** V\_banderas**:** STD\_LOGIC\_VECTOR**(**3 **DOWNTO** 0**);**

**variable** V\_opCode**:** STD\_LOGIC\_VECTOR**(**4 **DOWNTO** 0**);**

**variable** V\_clr**,** V\_LF**,** V\_LVL**:** STD\_LOGIC**;**

**variable** V\_Microinstruccion**:** std\_logic\_vector**(**19 **downto** 0**);**

--Cadena

**VARIABLE** CADENA **:** STRING**(**1 **TO** 7**);**

**begin**

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

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

--Impresion de Cadenas

CADENA **:=** "OP\_CODE"**;**

**write(**LINEA\_RES**,** CADENA**,** **right,** CADENA'**LENGTH+**1**);** --OP\_CODE

CADENA **:=** "FUNCODE"**;**

**write(**LINEA\_RES**,** CADENA**,** **right,** CADENA'**LENGTH+**1**);** --FUNC\_CODE

CADENA **:=** " FLAGS"**;**

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

CADENA **:=** " CLR"**;**

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

CADENA **:=** " LF"**;**

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

CADENA **:=** "MICROIN"**;**

**write(**LINEA\_RES**,** CADENA**,** **right,** CADENA'**LENGTH+**14**);** --MICROINSTRUCCION

CADENA **:=** " NIVEL"**;**

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

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

--Impresion de Resultados

**wait** **for** 100 ns**;**

**for** j **in** 0 **to** 46 **loop**

--Lectura de VECTORES.TXT

**readline(**ARCH\_VEC**,**LINEA\_VEC**);** -- lee una linea completa

**read(**LINEA\_VEC**,** V\_opCode**);**

opCode **<=** V\_opCode**;**

**read(**LINEA\_VEC**,** V\_funCode**);**

funCode **<=** V\_funCode**;**

**read(**LINEA\_VEC**,** V\_banderas**);**

banderas **<=** V\_banderas**;**

**read(**LINEA\_VEC**,** v\_clr**);**

clr **<=** V\_clr**;**

**read(**LINEA\_VEC**,** V\_LF**);**

LF **<=** V\_LF**;**

**wait** **until** **RISING\_EDGE(**CLK**);**--Flanco de subida

V\_Microinstruccion **:=** Microinstruccion**;** -- Asignacion de Salida

V\_LVL **:=** LVL**;**

--Escritura de Resultados

**write(**LINEA\_RES**,** V\_opCode**,** **right,** 8**);**

**write(**LINEA\_RES**,** V\_funCode**,** **right,** 8**);**

**write(**LINEA\_RES**,** V\_banderas**,** **right,** 8**);**

**write(**LINEA\_RES**,** V\_CLR**,** **right,** 8**);**

**write(**LINEA\_RES**,** V\_LF**,** **right,** 8**);**

**write(**LINEA\_RES**,** V\_Microinstruccion**,** **right,** 21**);**

**if(**V\_LVL **=** '1'**)** **then**

CADENA **:=** " ALTO"**;**

**write(**LINEA\_RES**,** CADENA**,** **right,** 8**);**

**else**

CADENA **:=** " BAJO"**;**

**write(**LINEA\_RES**,** CADENA**,** **right,** 8**);**

**end** **if;**

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

**end** **loop;**

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

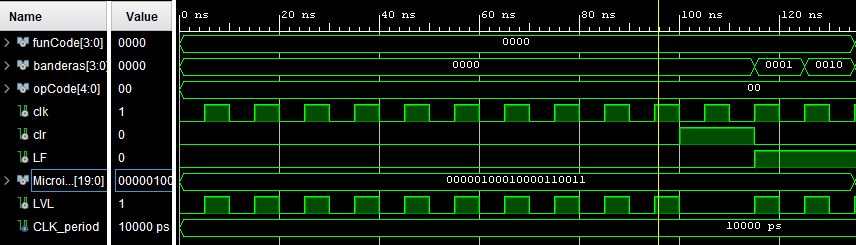
file\_close**(**ARCH\_RES**);**--Cierra el archivo

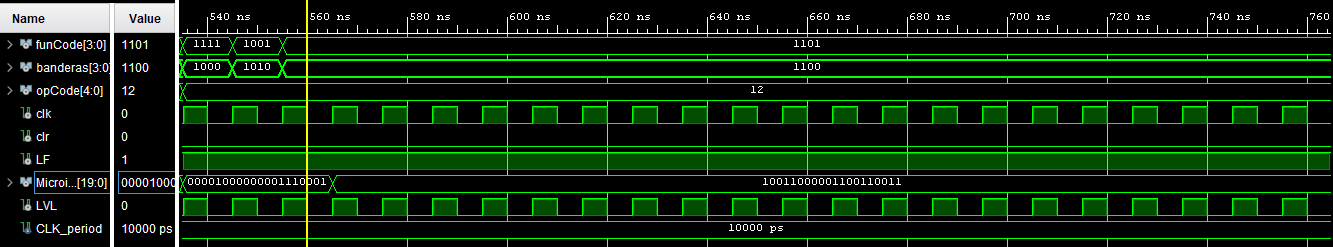
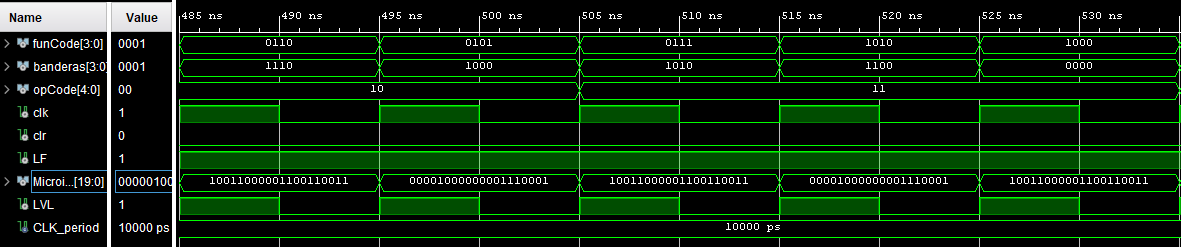
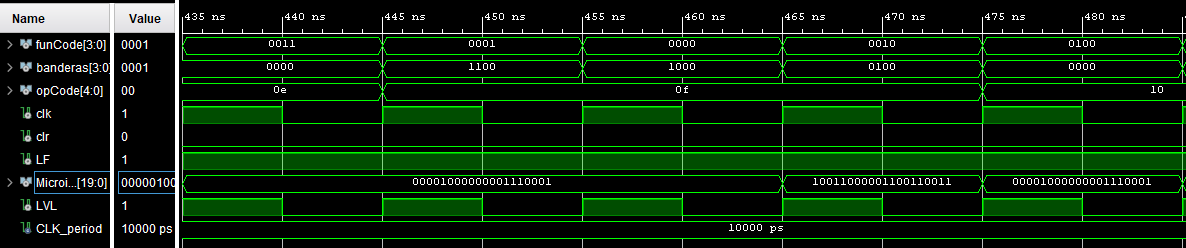
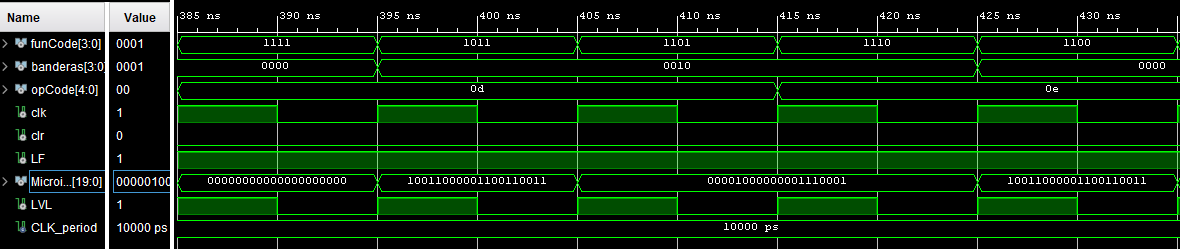
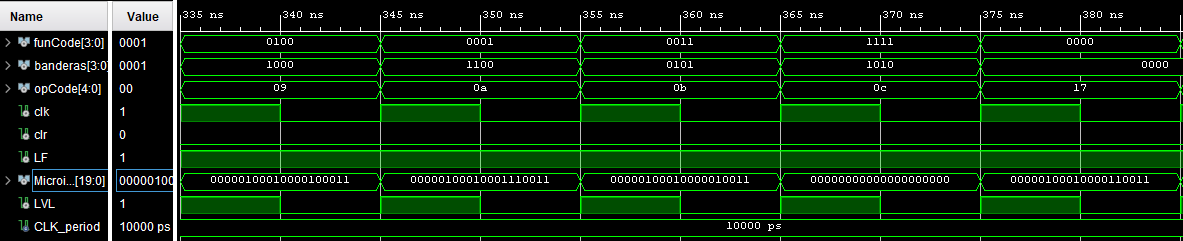
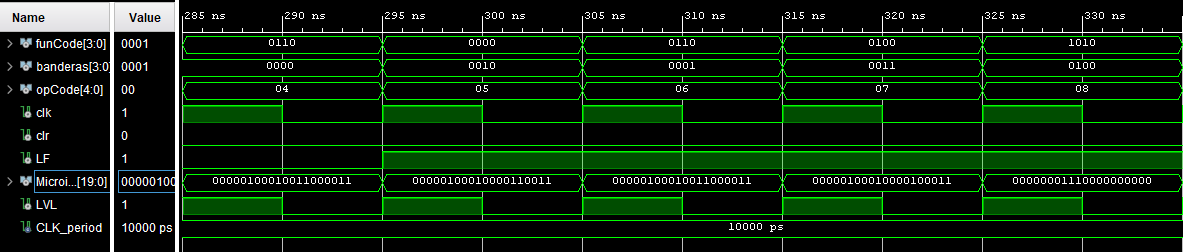
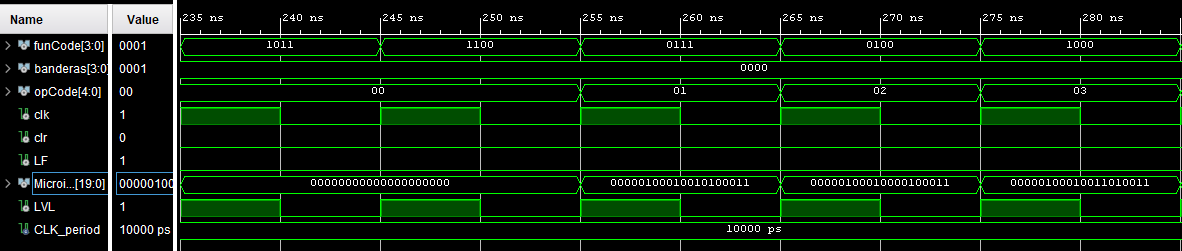
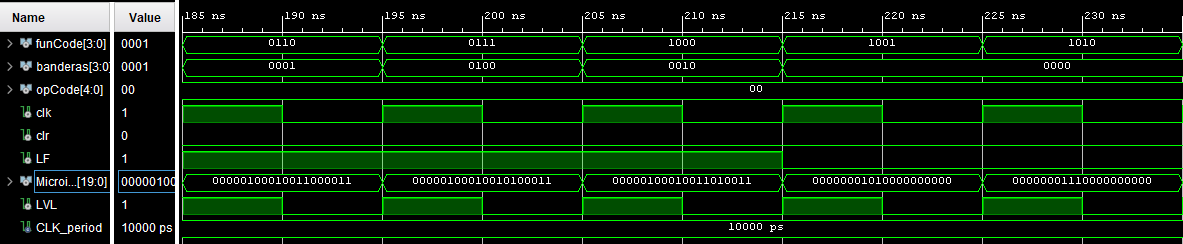
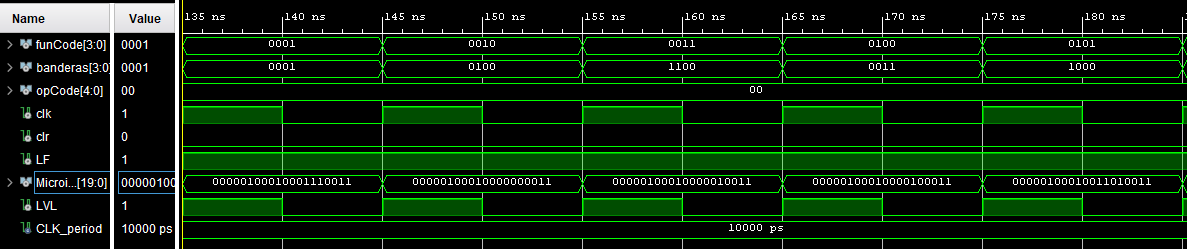
**wait;**

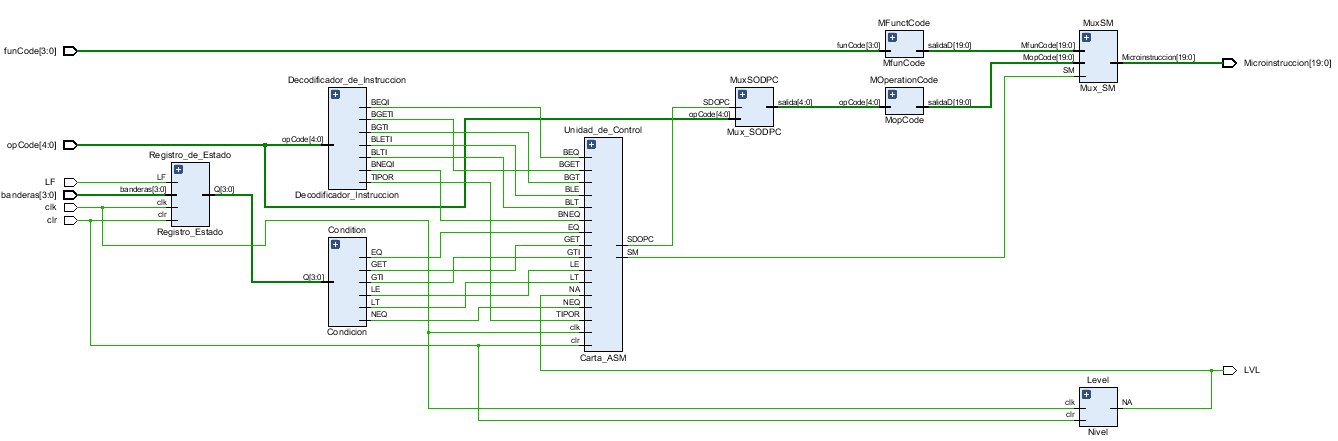
**end** **process;**-- Stimulus process

**end** Behavioral**;**

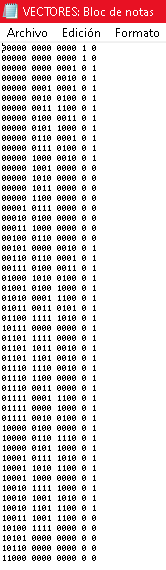
**Simulación:**





**Diagrama RTL:**

**Archivo de Entradas:**



**Archivo de Salidas:**

