*****Instituto Politécnico Nacional***

***Escuela Superior de Cómputo***

*Arquitectura de Computadoras*

***Practica 10: Pila Hardware 2***

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

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

***Profesor:*** *Nayeli Vega García*

***Fecha de entrega:*** *17 de abril del 2020*

**Código de Implementación:**

* **Pila**

**library** IEEE**;**

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

**use** IEEE**.**STD\_LOGIC\_arith**.ALL;**--Datos con signos y sin signo, y operaciones aritmeticas

**use** IEEE**.**STD\_LOGIC\_unsigned**.ALL;**--Realizar operaciones sin signo para los ST\_LOGIC\_VECTOR

**entity** Pila **is**

**generic** **(** N**:** integer **:=**3**;**

M**:** integer **:=**16**);**

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

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

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

SP**:** **out** std\_logic\_vector**(**N**-**1 **downto** 0**));**

**end** Pila**;**

**architecture** Behavioral **of** Pila **is**

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

**signal** aux**:** banco**;**

**signal** SP1**:** integer **range** 0 **to** **(**2**\*\***N**)-**1**;**

**begin**

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

**variable** SPout **:** integer **range** 0 **to** **(**2**\*\***N**)-**1**;**

**begin**

**if(**clr **=** '1'**)then**

SPout **:=** 0**;**

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

**elsif(**clk'**event** and clk **=** '1'**)then**

**if(**WPC **=** '0' and UP **=** '0' and DW **=** '0'**)then**--incremento

aux**(**SPout**)** **<=** aux**(**SPout**)+**1**;**

**elsif(**WPC **=** '1' and UP **=** '1' and DW **=** '0'**)then**--CALL

SPout **:=** SPout **+** 1**;**

aux**(**SPout**)** **<=** PC\_in**;**

**elsif(**WPC **=** '1' and UP **=** '0' and DW **=** '0'**)then**--JUMP

aux**(**SPout**)** **<=** PC\_in**;**

**elsif(**WPC **=** '0' and UP **=** '0' and DW **=** '1'**)then**--RET

SPout **:=** SPout **-** 1**;**

aux**(**SPout**)** **<=** aux**(**SPout**)+**1**;**

**end** **if;**

**end** **if;**

SP1 **<=** SPout**;**

**end** **process;**

SP **<=** conv\_std\_logic\_vector**(**SP1**,** 3**);**

PC\_out **<=** aux**(**SP1**);**

**end** Behavioral**;**

* **StackPointer**

**library** IEEE**;**

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

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

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

**entity** StackPointer **is**

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

SP**:** **inout** STD\_LOGIC\_VECTOR**(**2 **downto** 0**));**

**end** StackPointer**;**

**architecture** Behavioral **of** StackPointer **is**

**begin**

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

**begin**

**if(**clr **=** '1'**)** **then**

SP **<=** "000"**;**

**elsif(RISING\_EDGE(**clk**))then**

**if(**UP **=** '1'**)then**

SP **<=** SP **+** 1**;**

**elsif(**DW **=** '1'**)then**

SP **<=** SP **-** 1**;**

**end** **if;**

**end** **if;**

**end** **process;**

**end** Behavioral**;**

* **Demux**

**library** IEEE**;**

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

**entity** Demux **is**

**Port** **(** WPC **:** **in** STD\_LOGIC**;**

SP**:** **in** STD\_LOGIC\_VECTOR**(**2 **downto** 0**);**

dex**:** **out** STD\_LOGIC\_VECTOR**(**7 **downto** 0**));**

**end** Demux**;**

**architecture** Behavioral **of** Demux **is**

**begin**

dex**(**0**)** **<=** WPC **when** SP **=** "000" **else** '0'**;**

dex**(**1**)** **<=** WPC **when** SP **=** "001" **else** '0'**;**

dex**(**2**)** **<=** WPC **when** SP **=** "010" **else** '0'**;**

dex**(**3**)** **<=** WPC **when** SP **=** "011" **else** '0'**;**

dex**(**4**)** **<=** WPC **when** SP **=** "100" **else** '0'**;**

dex**(**5**)** **<=** WPC **when** SP **=** "101" **else** '0'**;**

dex**(**6**)** **<=** WPC **when** SP **=** "110" **else** '0'**;**

dex**(**7**)** **<=** WPC **when** SP **=** "111" **else** '0'**;**

**end** Behavioral**;**

* **Mux**

**library** IEEE**;**

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

**entity** Mux **is**

**Port(** PCin0**:** **in** std\_logic\_vector**(**15 **downto** 0**);**

PCin1**:** **in** std\_logic\_vector**(**15 **downto** 0**);**

PCin2**:** **in** std\_logic\_vector**(**15 **downto** 0**);**

PCin3**:** **in** std\_logic\_vector**(**15 **downto** 0**);**

PCin4**:** **in** std\_logic\_vector**(**15 **downto** 0**);**

PCin5**:** **in** std\_logic\_vector**(**15 **downto** 0**);**

PCin6**:** **in** std\_logic\_vector**(**15 **downto** 0**);**

PCin7**:** **in** std\_logic\_vector**(**15 **downto** 0**);**

SP **:** **in** STD\_LOGIC\_VECTOR **(**2 **downto** 0**);**

PC\_out**:** **out** STD\_LOGIC\_VECTOR **(**15 **downto** 0**));**

**end** Mux**;**

**architecture** Behavioral **of** Mux **is**

**begin**

PC\_out **<=** PCin0 **when** SP **=** "000" **else**

PCin1 **when** SP **=** "001" **else**

PCin2 **when** SP **=** "010" **else**

PCin3 **when** SP **=** "011" **else**

PCin4 **when** SP **=** "100" **else**

PCin5 **when** SP **=** "101" **else**

PCin6 **when** SP **=** "110" **else**

PCin7**;**

**end** Behavioral**;**

**Código de Simulación:**

**library** IEEE**;**

**LIBRARY** STD**;**

**USE** STD**.**TEXTIO**.ALL;**

**use** IEEE**.**STD\_LOGIC\_1164**.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** Pila\_tb **is**

**end** Pila\_tb**;**

**architecture** Behavioral **of** Pila\_tb **is**

**component** Pila

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

PC\_out **:** **out** STD\_LOGIC\_VECTOR **(**15 **downto** 0**);**

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

SP**:** **out** STD\_LOGIC\_VECTOR **(**2 **downto** 0**));**

**end** **component;**

--Inputs

**signal** PC\_in **:** STD\_LOGIC\_VECTOR **(**15 **downto** 0**)** **:=** **(others** **=>** '0'**);**

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

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

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

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

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

--Outputs

**signal** PC\_out **:** STD\_LOGIC\_VECTOR **(**15 **downto** 0**);**

**signal** SP**:** STD\_LOGIC\_VECTOR **(**2 **downto** 0**);**

-- Clock period definitions

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

**begin**

-- Instantiate the Unit Under Test (UUT)

uut**:** pila **PORT** **MAP** **(**

PC\_in **=>** PC\_in**,**

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

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

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

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

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

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

SP **=>** SP

**);**

-- 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\_PC\_in**,** V\_PC\_out**:** STD\_LOGIC\_VECTOR**(**15 **DOWNTO** 0**);**

**variable** v\_SP **:** std\_logic\_vector**(**3 **downto** 0**);**

**variable** V\_clr**,** V\_UP**,** V\_DW**,** V\_WPC**:** STD\_LOGIC**;**

--Cadena

**variable** CADENA **:** STRING**(**1 **TO** 4**);**

**begin**

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

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

--Impresion de Cadenas

CADENA **:=** " IN"**;**

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

CADENA **:=** " UP"**;**

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

CADENA **:=** " DW"**;**

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

CADENA **:=** " WPC"**;**

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

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

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

CADENA **:=** " SP"**;**

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

CADENA **:=** " PC"**;**

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

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

--Impresión de Resultados

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

**for** i **in** 0 **to** 25 **loop**

--Lectura de cadenas de VECTORES.txt

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

hread**(**LINEA\_VEC**,** V\_PC\_in**);**

PC\_in **<=** V\_PC\_in**;**

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

UP **<=** V\_UP**;**

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

DW **<=** V\_DW**;**

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

WPC **<=** V\_WPC**;**

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

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

**wait** **until** **RISING\_EDGE(**clk**);**

V\_PC\_out **:=** PC\_out**;** -- asignando salida

V\_SP **:=** '0' **&** SP**;**

--Escribiendo Resultados

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

**write(**LINEA\_RES**,** V\_UP**,** **right,** 5**);**

**write(**LINEA\_RES**,** V\_DW**,** **right,** 5**);**

**write(**LINEA\_RES**,** V\_WPC**,** **right,** 5**);**

**write(**LINEA\_RES**,** V\_clr**,** **right,** 5**);**

Hwrite**(**LINEA\_RES**,** V\_SP**,** **right,** 5**);**

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

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

**end** **loop;**

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

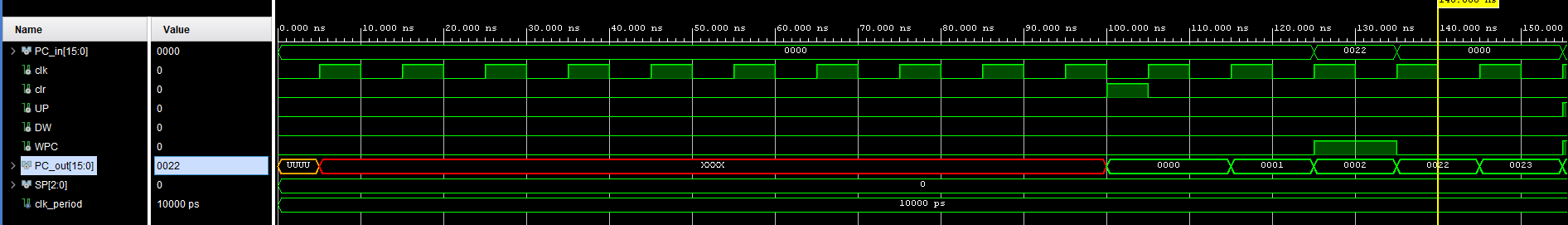
file\_close**(**ARCH\_RES**);**--Cierra el archivo

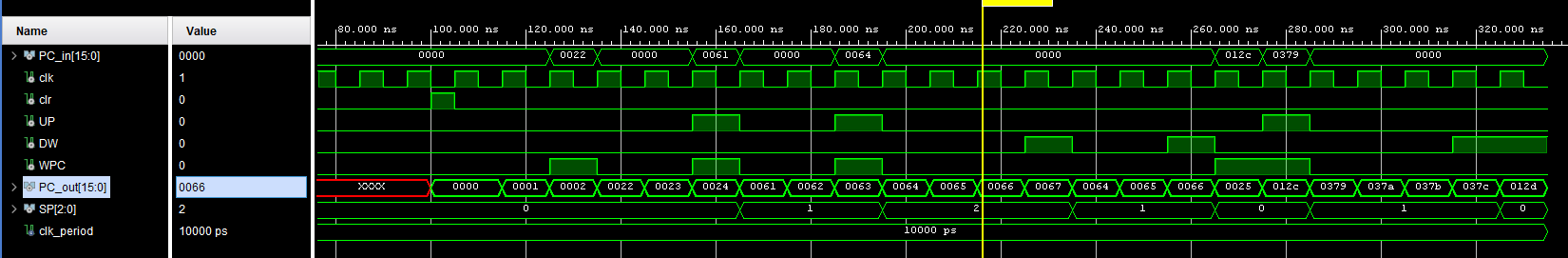
**wait;**

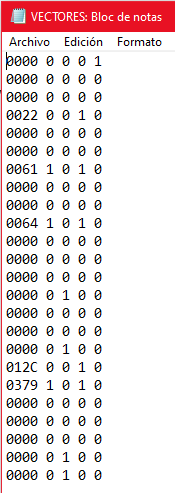
**end** **process;**--Stimulus process

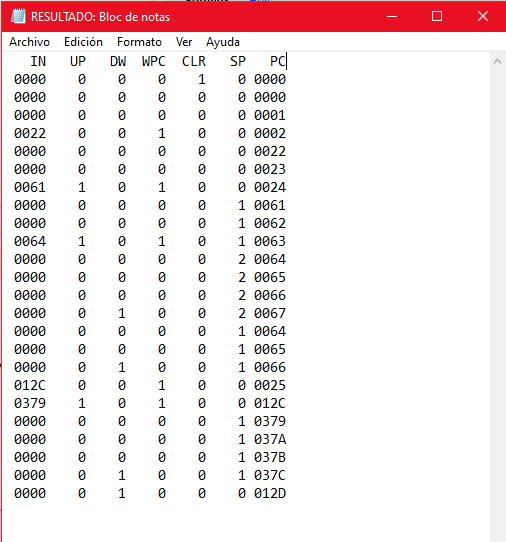
**end** Behavioral**;**

**Simulación:**





**Archivo de entrada: Archivo de salida:**



**Diagrama RTL:**

