

LABORATORIO DE DISEÑO DIGITAL MODERNO

Diseño y construcción de una carta ASM.

Práctica 11

Profesora: M.I. Norma Elva Chávez Rodríguez.

OBJETIVO:

El alumno comprenderá utilizando lenguaje VHDL, el funcionamiento y manejo de cartas ASM para la creación de máquinas de estados.

INTRODUCCIÓN:

Las cartas ASM (Algorithmic State Machine), indican el funcionamiento de un sistema secuencial de forma gráfica y están especialmente enfocadas, como indica su nombre, a representar algoritmos secuenciales, fue creado por Chris Clare.

ESPECIFICACIONES:

Diseñar un contador que cuente en forma ascendente/descendente los números binarios del cero al tres, cuando el valor de la variable A sea uno la cuenta se efectuará de forma ascendente y si el valor de la variable A es cero la cuenta se efectuará en forma descendente. El cambio de cuenta de ascendente a descendente y viceversa se puede efectuar en cualquier estado.

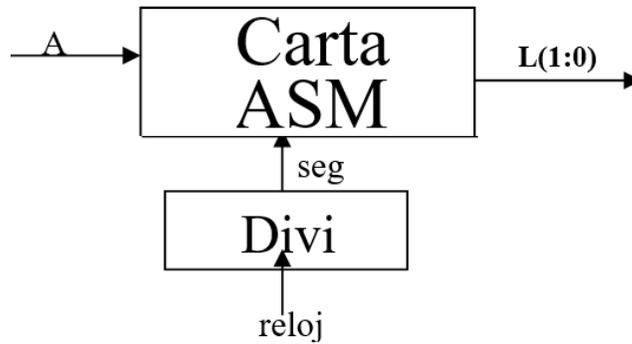
LABORATORIO DE DISEÑO DIGITAL MODERNO

Diseño y construcción de una carta ASM.

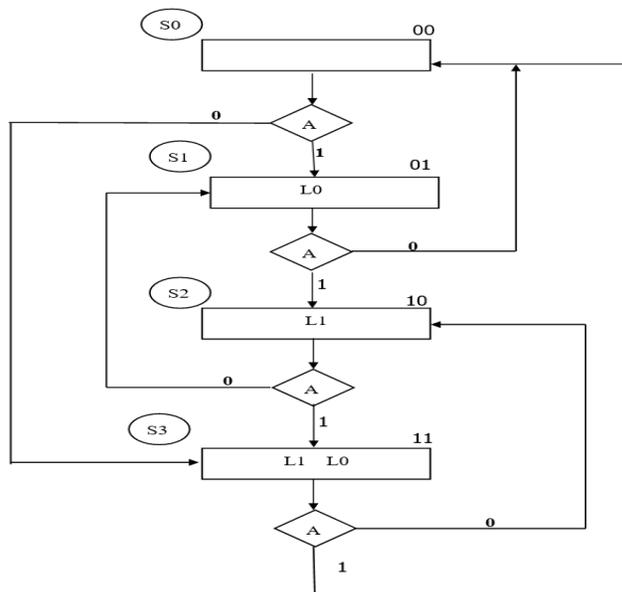
Práctica 11

Profesora: M.I. Norma Elva Chávez Rodríguez.

DIAGRAMA DE BLOQUES:



CARTA ASM:



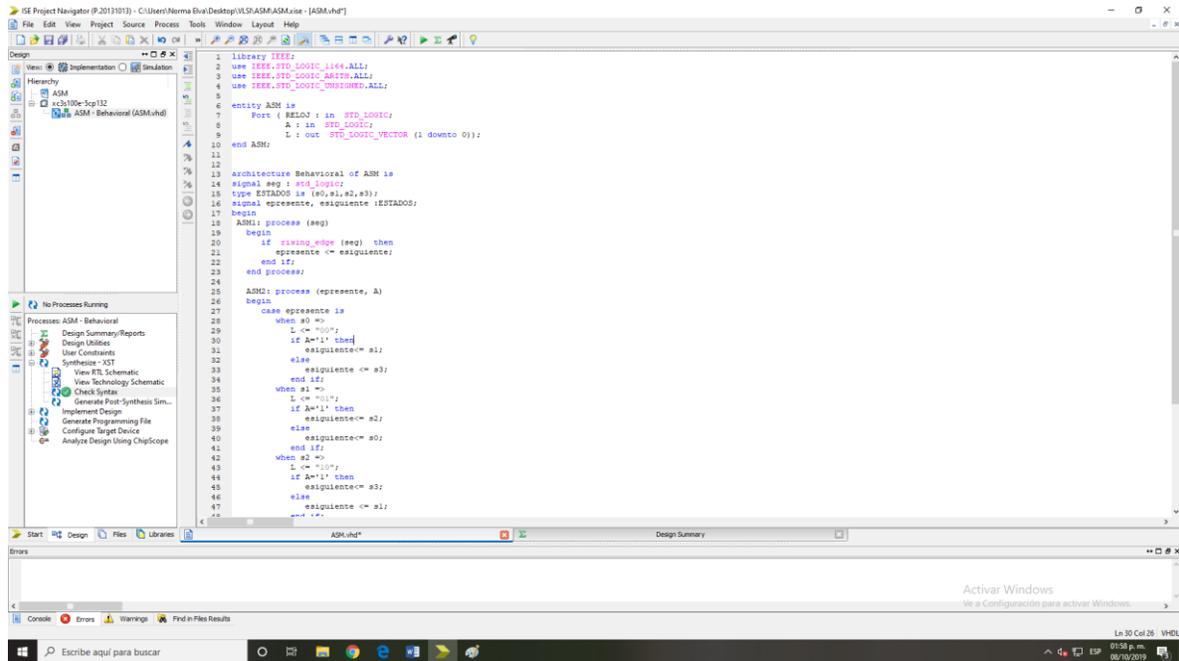
LABORATORIO DE DISEÑO DIGITAL MODERNO

Diseño y construcción de una carta ASM.

Práctica 11

Profesora: M.I. Norma Elva Chávez Rodríguez.

CÓDIGO EN LENGUAJE VHDL:



```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 entity ASM is
7     Port ( RESET : in STD_LOGIC;
8           A : in STD_LOGIC;
9           L : out STD_LOGIC_VECTOR (1 downto 0));
10 end ASM;
11
12
13 architecture Behavioral of ASM is
14     signal seq : STD_LOGIC;
15     type ESTADOS is (e0,e1,e2,e3);
16     signal eprestate, esiguiente : ESTADOS;
17 begin
18     ASM1: process (seq)
19     begin
20         if rising_edge (seq) then
21             eprestate <= esiguiente;
22         end if;
23     end process;
24
25     ASM2: process (eprestate, A)
26     begin
27         case eprestate is
28             when e0 =>
29                 L <= "00";
30                 if A='1' then
31                     esiguiente<= e1;
32                 else
33                     esiguiente <= e3;
34                 end if;
35             when e1 =>
36                 L <= "01";
37                 if A='1' then
38                     esiguiente<= e2;
39                 else
40                     esiguiente<= e0;
41                 end if;
42             when e2 =>
43                 L <= "10";
44                 if A='1' then
45                     esiguiente<= e3;
46                 else
47                     esiguiente <= e1;
48                 end if;
49         end case;
50     end process;
51 end;
```