12-4-2024

Programación de sistemas reconfigurables

ASM Y ME

Ingeniería en electrónica y computación

ANDRADE SALAZAR, IGNACIO
CENTRO UNIVERSITARIO DE LOS VALLES, UNIVERSIDAD DE GUADALAJARA

ASM Y ME

1.- Con que estructuras se define una máquina de estado

Las estructuras de los sistemas secuenciales síncronos basan su funcionamiento en los elementos de memoria conocidos como flip-flop, la palabra sincronía se refiere a que cada uno de estos elementos de memoria que interactúan en un sistema, los cuales están conectados a la misma señal de reloj, de forma tal que solo se producirá un cambio de estado en el sistema cuando ocurra un flanco de disparo o un pulso en la señal de reloj. Esta condición permite sincronizar la activación de las señales de salidas propias de la aplicación dentro de una secuencia. Dichas salidas se producen mediante el uso de dos estructuras: maquina de Moore y máquina de Mealy.

En la estructura de Moore, las señales de salida solo son función del estado actual combinatorio.

En la estructura de Mealy, las señales de salida dependen tanto del estado combinatorio en el que se encuentra el sistema como de la entrada actual.

2.- Ponga un ejemplo del uso del “type”

Como primer paso en nuestro diseño, consideremos los estados d0, d1, d2, y d3, los cuales, para poder ser presentados en código VHDL, deben definirse dentrio de un tipo de datos denominado enumerado y utilizar la declaración type.

type estados **is** (d0, d1, d2, d3);

signal edo_presente, edo_futuro: estados;

3.-Transcriba el listado 3.9, complételo bien.

```
library ieee;
use ieee.std_logic_1164.all;
entity diagrama is
port ( clk,x: in std_logic;
        z: out std_logic);
end diagrama;
architecture arq_diagrama of diagrama ios
type estados is (d0, d1, d2, d3);
signal edo_presente, edo_futuro: estados;
begin
proceso1: process (edo_presente, x) begin
        case edo_presente is
                when d0 => z=> '0';
                        if x='1' then
                                edo:futuro <= d1;
                        else
                                edo_futuro<= d0;
                        end if;
                when d2 => z<='0';
                        if x='1' then
                                edo_futuro <= d3;
                        else
                                edo_futuro <= d0;
                        end if;
                when d3 => z<='0';
                        if x='1' then
                                edo:futuro <= d0;
                                z <= '1'
```

```
                else
                        edo_futuro <= d3;
                        z <='0';
                end if;
        end case;
end process proceso1;
    proceso2: process(clk) begin
        if (clk'event and clk='1') then
                edo:presente <= edo_futuro;
        end if;
    end process proceso2;
end arq_diagrama;
```

4.- Diga de que trata la Introducción del capítulo 4, hasta antes de 4.1

En la vida cotidiana, la descripción de un sistema secuencial de aparente complejidad, como los cajeros automáticos, el control de trenes, entre otros, requiere, en su análisis, de más de una condición de entrada y un numero considerable de variables de salida.

La descripción a través de la carta Algoritmo de la Maquina de Estado (ASM), diseñado en específico para definir algoritmos de hardware, permite un mejor análisis y una mejor interpretación debido generalmente a que las secuencias a describir contienen un mayor numero de variables de entrada y salida.

El aspecto principal de un algoritmo ASM considera las siguientes características.

Primero: el algoritmo debe ser finito, debe tener un número determinado de estados.

Segundo: el algoritmo debe ser definido, en cada estado deben estar por completo especificadas todas las acciones que se llevan a cabo, esto incluye las entradas, las salidas y las decisiones que nos conducen a ese estado.