

CURSO 3R
Fecha: 31/10/2015

Parcial Técnicas Digitales I

Nombre: Sueldo Enrique
Legajo: 62508

4 (Cafido)

EJ.	Puntaje
1	5
2	15
3	20
4	20
Total	60

- 1 (25 puntos) Diseñe un contador Hexadecimal de números pares o impares según el estado de la señal de control X.
X=0 Salen números impares.
X=1 Salen números pares.
Para el diseño utilizar FF tipo JK.
 - 1.1 Tabla de transición de estados.
 - 1.2 Minimización de las funciones.
 - 1.3 Circuito esquemático.
- 2 (25 puntos) Diseñe un con un registro de desplazamiento un sistema secuencial que genere las siguientes salidas.
111 - 011 - 100 - 111 - 000 y se repite.
 - 2.1 Realizar diagrama de brujin e indicar el camino seleccionado.
 - 2.2 Diseñar el sistema y entregar las tablas y funciones correspondientes.
 - 2.3 Utilizar Multiplexores de 8:1 para implementar la lógica de realimentación.
 - 2.4 Realice el circuito esquemático.
- 3 (30 puntos) Hay que controlar las puertas de cristal de un edificio. Disponemos de un detector de personas P, de un detector de puerta abierta A y de un detector de puerta cerrada C. Si viene una persona se abre la puerta. Si la puerta empieza a abrirse, completa el ciclo entero aunque desaparezca la persona. Si sigue apareciendo gente la puerta permanece abierta continuamente. Si cuando estaba cerrándose la puerta aparece una persona, la puerta vuelve a abrirse, completando un nuevo ciclo. Diseñar la maquina de estado, dibujar el circuito y describir en VHDL. La máquina de estado es una máquina de estado de Moore.
 - 3.1 Diagrama de estados.
 - 3.2 Tabla de transición.
 - 3.3 Descripción en VHDL.
- 4 (20 puntos) Diseñe un oscilador con dos inversores de 15Khz con un ciclo de trabajo del 25%. (Ciclo de trabajo = (tiempo en alto)/ periodo)*100)
 - 4.1 Dibuje el circuito.
 - 4.2 Dibuje las formas de onda involucradas.
 - 4.3 Calcule los valores de RC para obtener una frecuencia de 15Khz.

```

type state_type is (st1_<name_state>, st2_<name_state>, ...);
signal state, next_state : state_type;
signal <output>_i : std_logic;

SYNC_PROC: process (<clock>)
begin
  if (<clock>'event and <clock> = '1') then
    if (<reset> = '1') then
      state <= st1_<name_state>;
      <output>_i <= '0';
    else
      state <= next_state;
      <output>_i <= <output>_i;
      -- assign other outputs to internal signals
    end if;
  end if;
end process;

NEXT_STATE_DECODE: process (state, <input1>, <input2>, ...)
begin
  next_state <= state;
  case (state) is
    when st1_<name> =>
      if <input_1> = '1' then
        next_state <= st2_<name>;
      end if;
    when st2_<name> =>
      if <input_2> = '1' then
        next_state <= st3_<name>;
      end if;
    when st3_<name> =>
      next_state <= st1_<name>;
    when others =>
      next_state <= st1_<name>;
  end case;
end process;

OUTPUT_DECODE: process (state, <input1>, <input2>, ...)
begin
  if (state = st3_<name> and <input1> = '1') then
    <output>_i <= '1';
  else
    <output>_i <= '0';
  end if;
end process;

OUTPUT_DECODE: process (state)
begin
  if state = st3_<name> then
    <output>_i <= '1';
  else
    <output>_i <= '0';
  end if;
end process;

```