

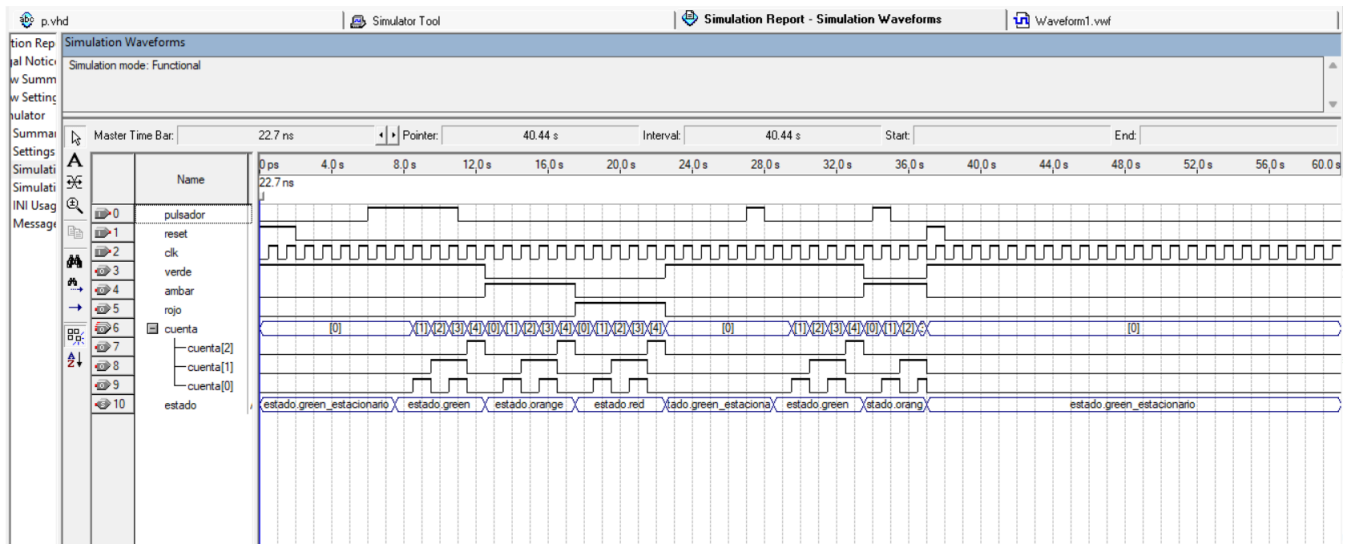
Práctica 4: Introducción diseño digital circuitos combinacionales

Tecnología de Computadores 2021 – 2022

Grupo_9_Jareño_Manzaneque

A. ESTUDIO FINAL PRÁCTICA 4

1. Simulación.



Si presionamos el reset, todas las salidas están apagas excepto el verde_estacionario.

Si se presiona el pulsador, en el momento que hay un flanco de reloj comienza la cuenta atrás y empieza a estar 5 segundos en casa estado de forma que cubre los estados verdes, ámbar y rojo, consecutivamente cada uno durante 5 segundos, y, después vuelve al verde_estacionario.

Si por algún casual se presiona el reset mientras se está en un estado de la FSM que no sea el verde_estacionario, este vuelve al color verde.

2. Código VHDL.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;
```

```
ENTITY p IS
```

```
PORT (
```

```
    pulsador: in STD_logic;
```

```
    clk : IN STD_LOGIC; -- Reloj de 1Hz y 5s.
```

```
    reset : IN STD_LOGIC; -- Reset, activo a nivel alto-
```

```
    verde : OUT STD_LOGIC; -- Luz verde del semáforo.
```

```
    ambar : OUT STD_LOGIC; -- Luz ambar del semáforo.
```

```
    rojo : OUT STD_LOGIC; -- Luz roja del semáforo.
```

```
    cuenta : OUT INTEGER RANGE 0 TO 4 -- Muestra la cuenta del temporizador.
```

```
);
```

```
END p;
```

ARCHITECTURE funcional OF p IS

TYPE estadoMaquina IS (clear,green,orange,red, green_estacionario); --Estados de la FSM

SIGNAL estado, siguiente : estadoMaquina; --Señal actual y siguiente de la FSM

SIGNAL e : STD_LOGIC; --Enable

SIGNAL c : STD_LOGIC; --Clear

SIGNAL f_2 : INTEGER RANGE 0 TO 1; --Salida activa cuando contador llega a 4

signal Q0: STD_LOGic; -- Detector de Flanco

signal Q1: STD_LOGic; --Detector de Flanco

signal pulsador_f: std_logic; --Salida Detector de Flanco

signal contador: INTEGER RANGE 0 TO 4; --Contador

BEGIN

--Detector de Flancos

process(clk, reset, pulsador)

begin

if reset = '1' then

Q0<='0' ;

Q1<='0';

elsif clk'event and clk= '1' then

Q0<= pulsador;

Q1<=Q0;

end if;

end process;

pulsador_f <= not Q1 and Q0;

--Asigno el valor del contador a cuenta para que se muestre en waveform

cuenta<=contador;

--Contador

process(clk, reset, E)

begin

if reset = '1' then

contador <= 0;

elsif clk'event and clk = '1' then

--Si es 4 entonces lo reseteo a 0

if contador >= 4 then

contador <= 0;

--Si enable is 1 entonces empieza a contar

elsif e = '1' then

contador <= contador + 1;

else

contador <= 0;

end if;

end if;

end process;

-- F pasa a valer 1 cuando el contador llega a 4 y, por tanto, ha pasado un estado

f_2 <= 1 when contador >= 4 else 0;

--PARTE COMBINACIONAL

```
PROCESS (clk,reset)
BEGIN
--Estado en reposo
  IF reset='1' THEN

    estado <= green_estacionario;

  ELSIF clk='1' AND clk'event THEN

    estado <= siguiente;
  END IF;
END PROCESS;

--ESTADOS DE LA FSM
PROCESS (estado,f_2)
BEGIN

  CASE estado IS
  --ESTADO VERDE
    WHEN green =>
      verde <= '1';
      ambar <= '0';
      rojo <= '0';
      e <= '1';
      c <= '0';
      --Cambio de estado
      IF f_2 = 0 THEN
        siguiente <= green;
      ELSE
        siguiente <= orange;
      END IF;
    --ESTADO NARANJA
    WHEN orange =>
      verde <= '0';
      ambar <= '1';
      rojo <= '0';
      e <= '1';
      c <= '0';
      --Cambio de estado
      IF f_2 = 0 THEN
        siguiente <= orange;
      ELSE
        siguiente <= red;
      END IF;
    --ESTADO ROJO
    WHEN red =>
      verde <= '0';
      ambar <= '0';
      rojo <= '1';
      e <= '1';
      c <= '0';
      --Cambio de estado
      IF f_2 = 1 THEN
        siguiente <= green_estacionario;
      ELSE
        siguiente <= red;
```

```
END IF;
    --ESTADO ESTACIONARIO
    WHEN green_estacionario =>
        verde <= '1';
        ambar <= '0';
        rojo <= '0';
        e <= '0';
        c <= '0';
        --Si se pulsa el pulsador se inicia el semaforo
        IF f_2 = 0 and pulsador_f='1' THEN
            siguiente <= green;
        else
            siguiente<= green_estacionario;
        END IF;
    --OTRO ESTADO
    WHEN OTHERS =>
        verde <= '1';
        ambar <= '0';
        rojo <= '0';
        e <= '0';
        c <= '0';
        IF f_2 = 0 and pulsador_f='1' THEN
            siguiente <= green;
        else
            siguiente<= green_estacionario;
        END IF;
END CASE;
END PROCESS;
END funcional;
```

3. Video montaje.

El video del montaje está en la subida del alumno José Antonio Jareño García con NIA 100472166.