

CONTENIDO

	Carta de la academia	
Práctica 1	Diseño de un reloj digital	4
Práctica 2	Diseño de registros de corrimiento en cascada	5
Práctica 3	Diseño del control de un tren eléctrico	12
Práctica 4	Diseño del control de servomotores	17
Práctica 5	Diseño del control de intensidad en leds	22
Práctica 6	Diseño del control de motores a pasos	28
Práctica 7	Diseño del control de sensores ultrasónicos	33
Práctica 8	Diseño de un transmisor para comunicación serial	46
Práctica 9	Diseño de un receptor para comunicación serial	50
Práctica 10	Diseño de un generador de video VGA	56
Práctica 11	Emulador de display de 7 segmentos en un monitor	61
Práctica 12	Emulador de contadores en un monitor	69
Práctica 13	Captura de imágenes de cámara digital	75
	Bibliografía	79
	Glosario	95

Carta de la academia



FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA

RECIBIDO
11 SEP 11 AM 10 51

FACULTAD DE INGENIERÍA
DIRECCIÓN

ING. ALEJANDRO SOSA FUENTES
REPRESENTANTE DE LA DIVISIÓN DE INGENIERÍA ELÉCTRICA.
ANTE EL COSLDI
FACULTAD DE INGENIERÍA.

PRESENTE.

Por este conducto le informamos que la academia de Diseño Digital VLSI, concluyó la elaboración y la revisión de las prácticas Diseño VLSI, material requerido para el plan de estudios 2016, que en 2019-1 se impartirá por vez primera.

Esfuerzo que forma parte del proceso de homologación de la gestión de laboratorios, en el marco de acciones del plan de desarrollo 2015-2019 de esta Facultad.

Las prácticas que se le entregan son un testimonio del compromiso y profesionalismo de los académicos, que participan en la academia Diseño Digital VLSI, en la búsqueda de elevar la calidad de la docencia y favorecer el aprendizaje de los estudiantes, plasmando sus experiencias y sus conocimientos en esta publicación.

Agradeciendo de antemano la atención prestada.

ATENTAMENTE.

"POR MI RAZA HABLARÁ EL ESPÍRITU".

Ciudad Universitaria, Cd. Mx., a 29 de Agosto del 2017

COORDINADORA DE LA ACADEMIA DISEÑO DIGITAL VLSI
M.I. CHAVEZ RODRIGUEZ NORMA ELVA Firma
INTEGRANTES DE LA ACADEMIA DISEÑO DIGITAL VLSI
M.I. Flores Olvera Vicente Firma
M.I. Fonseca Chávez Elizabeth Firma
M.I. Guevara Rodríguez María del Socorro Firma
M.I. Ibarra Carrillo Mario Firma
M.I. Prieto Meléndez Rafael Firma
M.I. Ramírez Chavarría Roberto Firma

UNAM **RECIBIDO**
FACULTAD DE INGENIERÍA

11 SEP 2017

SECRETARÍA GENERAL

CCP. DR. CARLOS A. ESCALANTE SANDOVAL, DIRECTOR FACULTAD DE INGENIERÍA, UNAM.

Práctica 1.
DISEÑO DE UN RELOJ DIGITAL

OBJETIVO:

Demostrar a los estudiantes que en un FPGA las declaraciones concurrentes se efectúan al mismo tiempo (en paralelo). Implantar operaciones concurrentes mediante el diseño de un reloj digital, en donde el orden de escritura en las instrucciones concurrentes no afecta el resultado de síntesis o de simulación.

ESPECIFICACIONES:

Utilizando un FPGA y 4 displays de 7 segmentos, diseñar un reloj digital, el cual visualice en los dos primeros displays las horas y en los siguientes dos, los minutos. Cada vez que se llegue a 23 horas con 59 minutos, se reiniciará el conteo de horas y minutos. La figura 1.1 muestra el diagrama del bloque de este sistema.

DIAGRAMA DE BLOQUES:

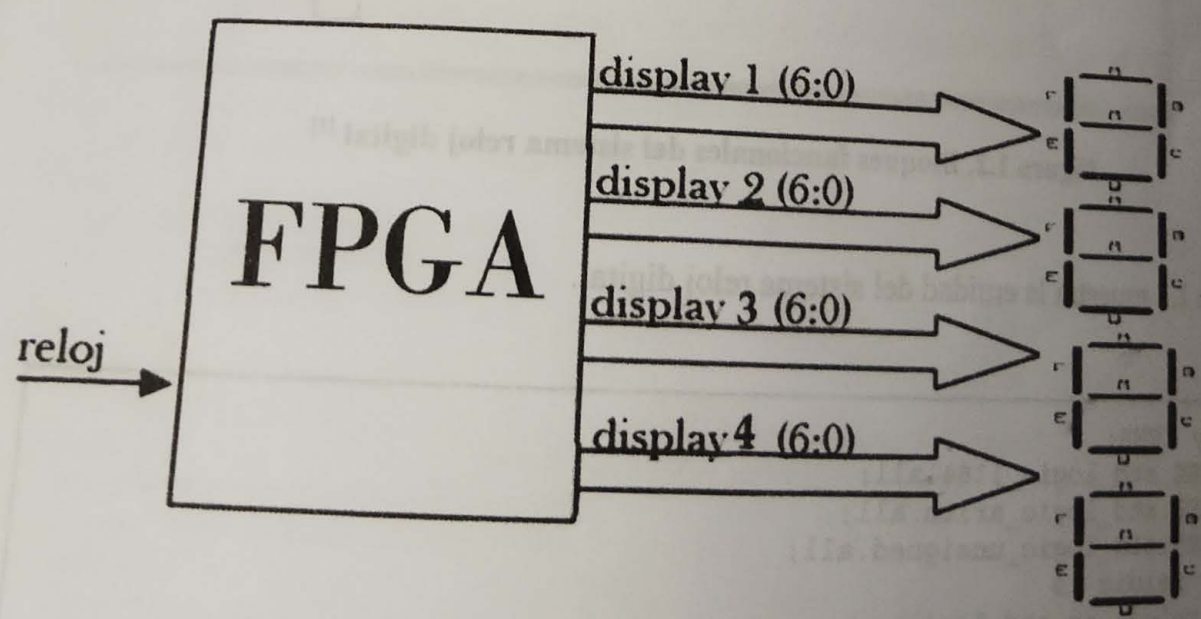


Figura 1.1. Diagrama de bloques del sistema reloj digital^[1]

Un FPGA puede configurarse con muchos bloques funcionales en lenguaje VHDL que estén ejecutando acciones a la vez. A estas acciones ejecutándose al mismo tiempo se le llama ejecución concurrente.

Las señales son declaraciones necesarias cuando se ejecutan instrucciones concurrentes, debido a que ellas unen los bloques funcionales.

La figura 1.2 muestra los bloques funcionales del sistema reloj digital donde las señales se muestran con flechas de color azul.

BLOQUES FUNCIONALES:

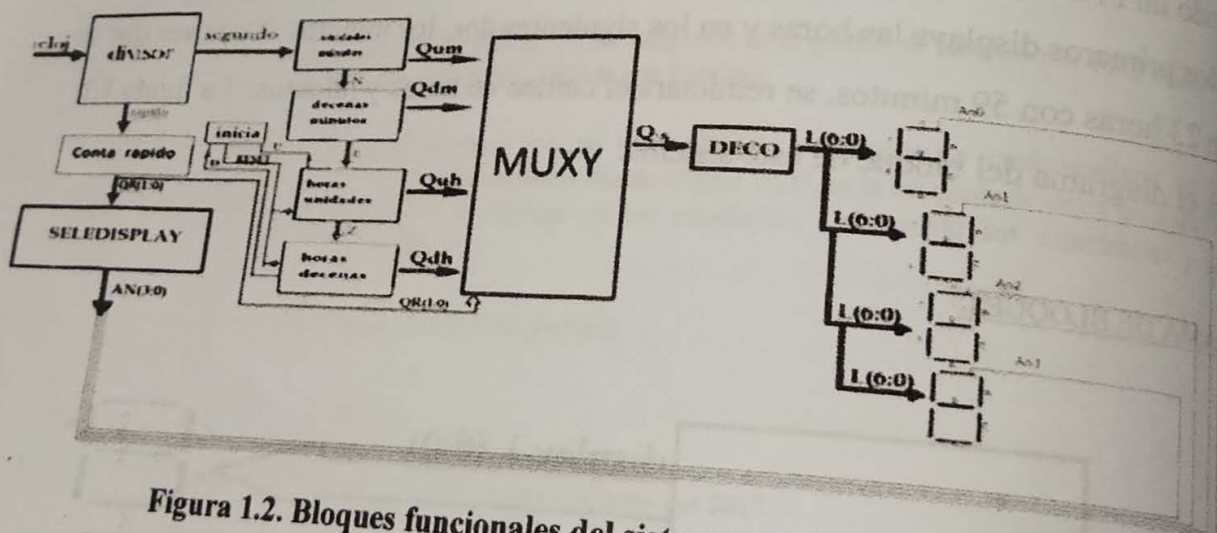


Figura 1.2. Bloques funcionales del sistema reloj digital [1]

La figura 1.3 muestra la entidad del sistema reloj digital.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
entity reldig is
Port (reloj: in std_logic;
      AN: out std_logic_vector (3 downto 0);
      L: out std_logic_vector (6 downto 0));
end reldig;
```

Figura 1.3. Entidad del sistema reloj digital

La figura 1.4 muestra la parte declaratoria de la arquitectura en el sistema reloj digital.

```
architecture behavioral of reldig is
  signal segundo: std_logic;
  signal rapido: std_logic;
  signal n: std_logic;
  signal Qs: std_logic_vector(3 downto 0);
  signal Qum: std_logic_vector(3 downto 0);
  signal Qdm: std_logic_vector(3 downto 0);
  signal e: std_logic;
  signal Qr: std_logic_vector(1 downto 0);
  signal Quh: std_logic_vector(3 downto 0);
  signal Qdh: std_logic_vector(3 downto 0);
  signal z: std_logic;
  signal u: std_logic;
  signal d: std_logic;
  signal reset: std_logic;
```

Figura 1.4. Parte declaratoria en la arquitectura del sistema reloj digital

La figura 1.5 muestra la parte operatoria de la arquitectura en el sistema reloj digital.


```

begin
  divisor: process (reloj)
    variable cuenta: std_logic_vector(27 downto 0) := X"00000000";
  begin
    if rising_edge (reloj) then
      if cuenta=X"48009E0" then
        cuenta:= X"00000000";
      else
        cuenta:= cuenta +1;
      end if;
    end if;
    segundo <= cuenta(22);
    rapido <= cuenta(10);
  end process;

  unidades: process (segundo)
    variable cuenta: std_logic_vector(3 downto 0) := "0000";
  begin
    if rising_edge (segundo) then
      if cuenta ="1001" then
        cuenta:="0000";
        n <= '1';
      else
        cuenta:= cuenta +1;
        n <= '0';
      end if;
    end if;
    qum <= cuenta;
  end process;

```

Figura 1.5. Parte operatoria de la arquitectura del sistema reloj digital


```

decenas: process (n)
  variable cuenta: std_logic_vector(3 downto 0) := "0000";
begin
  if rising_edge (n) then
    if cuenta = "0101" then
      cuenta := "0000";
      e <= '1';
    else
      cuenta := cuenta + 1;
      e <= '0';
    end if;
  end if;
  Qdm <= cuenta;
end process;

```

```

HoraU: Process(E, reset)
  variable cuenta: std_logic_vector(3 downto 0) := "0000";
begin
  if rising_edge(E) then
    if cuenta = "1001" then
      cuenta := "0000";
      Z <= '1';
    else
      cuenta := cuenta + 1;
      Z <= '0';
    end if;
  end if;
  if reset = '1' then
    cuenta := "0000";
  end if;
  Quh <= cuenta;
  U <= cuenta(2);
end Process;

```

24
42
0100-u
0010-d

cuenta = "0000"
u = cuenta(2)

```

HoraD: Process(Z, reset)
  variable cuenta: std_logic_vector(3 downto 0) := "0000";
begin
  if rising_edge(Z) then
    if cuenta = "0010" then
      cuenta := "0000";
    else
      cuenta := cuenta + 1;
    end if;
  end if;
  if reset = '1' then
    cuenta := "0000";
  end if;
  Qdh <= cuenta;
  D <= cuenta(1);
end Process;

```

d <= cuenta(1)

Figura 1.5. (continuación) Parte operatoria de la arquitectura del sistema reloj digital


```

decenas: process (n)
  variable cuenta: std_logic_vector(3 downto 0) := "0000";
begin
  if rising_edge (n) then
    if cuenta = "0101" then
      cuenta := "0000";
      e <= '1';
    else
      cuenta := cuenta + 1;
      e <= '0';
    end if;
  end if;
  Qdm <= cuenta;
end process;

HoraU: Process(E, reset)
  variable cuenta: std_logic_vector(3 downto 0) := "0000";
begin
  if rising_edge(E) then
    if cuenta = "1001" then
      cuenta := "0000";
      Z <= '1';
    else
      cuenta := cuenta + 1;
      Z <= '0';
    end if;
  end if;
  if reset = '1' then
    cuenta := "0000";
  end if;
  Quh <= cuenta;
  U <= cuenta(2);
end Process;

HoraD: Process(Z, reset)
  variable cuenta: std_logic_vector(3 downto 0) := "0000";
begin
  if rising_edge(Z) then
    if cuenta = "0010" then
      cuenta := "0000";
    else
      cuenta := cuenta + 1;
    end if;
  end if;
  if reset = '1' then
    cuenta := "0000";
  end if;
  Qdh <= cuenta;
  D <= cuenta(1);
end Process;

```

24

42

0100-u

0010-d

cuenta = "0000"
u d

u <= cuenta(2)

d <= cuenta(1)

Figura 1.5. (continuación) Parte operatoria de la arquitectura del sistema reloj digital

Uuenta = 0700
 cuenta = 0070
 2

```

inicia: process (U,D)
begin
    reset <= (U and D);
end process;

Contrapid: process (rapido)
variable cuenta: std_logic_vector(1 downto 0) := "00";
begin
    if rising_edge (rapido) then
        cuenta := cuenta + 1;
    end if;
    Qr <= cuenta;
end process;

muxy: process (Qr)
begin
    if Qr = "00" then
        Qs <= Qum;
    elsif Qr = "01" then
        Qs <= Qdm;
    elsif Qr = "10" then
        Qs <= Quh;
    elsif Qr = "11" then
        Qs <= Qdh;
    end if;
end process;

seledisplay: process (Qr)
begin
    case Qr is
        when "00" =>
            AN <= "1110";
        when "01" =>
            AN <= "1101";
        when "10" =>
            AN <= "1011";
        when others =>
            AN <= "0111";
    end case;
end process;

with Qs select
    L <= "1000000" when "0000", --0
        "1111001" when "0001", --1
        "0100100" when "0010", --2
        "0110000" when "0011", --3
        "0011001" when "0100", --4
        "0010010" when "0101", --5
        "0000010" when "0110", --6
    
```

Figura 1.5. (continuación) Parte operatoria de la arquitectura del sistema reloj digital

ACTIVIDADES COMPLEMENTARIAS:

- 1.- El alumno diseñará un reloj digital con alarma, en el cual se active una señal sonora cuya intensidad vaya aumentando que el usuario apague el sistema.
- 2.- El alumno diseñará un reloj digital que trabaje en sentido contrario a las manecillas del reloj, lo que significa que va a descontar tiempo empezando en un valor preseleccionado por el usuario.

