

Manual de prácticas Diseño Digital VLSI
División de Ingeniería Eléctrica
Departamento de Computación

Práctica 12. EMULADOR DE CONTADORES EN UN MONITOR

OBJETIVO:

El alumno aprenderá el diseño de contadores mediante un FPGA y con visualización en un monitor VGA.

ESPECIFICACIONES:

Utilizando un FPGA, un cable VGA y un monitor, diseñar un contador que cuente del cero al nueve. Cuando el conteo llegue a su límite, el contador deberá reiniciarse. La figura 12.1 muestra el diagrama de bloques y la figura 12.2 muestra los bloques funcionales del sistema emulador de contadores en un monitor.

DIAGRAMA DE BLOQUES:

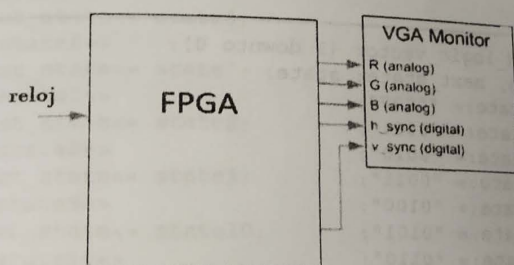


Figura 12.1. Diagrama de bloques del sistema emulador de contadores en un monitor

DIAGRAMA DE BLOQUES FUNCIONALES:

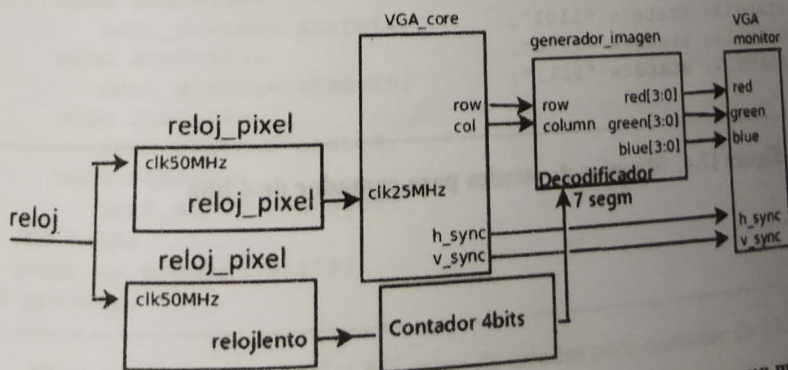


Figura 12.2 Diagrama de bloques del sistema emulador de contadores en un monitor

El proceso del divisor de frecuencia se muestra en la figura 12.3, donde puede observarse la entrada de reloj de 50 mHz y a su salida la señal de reloj "clklow" a muy baja frecuencia.

```

process (clk50mhz)
begin
    if (clk50mhz'event and (clk50mhz = '1')) then
        conteo<=conteo+1;
        if (conteo=1) then
            conteo<=0;
            clklow<=not (clklow);
        end if;
    end if;
end process;

```

Figura 12.3. Divisor de frecuencia del sistema emulador de contadores en un monitor

La figura 12.4 muestra la declaración de constantes y la máquina de estados requerida para la obtención del contador de cuatro bits.

```

subtype state is std_logic_vector (3 downto 0);
signal present_state, next_state: state;
constant state0: state:= "0000";
constant state1: state:= "0001";
constant state2: state:= "0010";
constant state3: state:= "0011";
constant state4: state:= "0100";
constant state5: state:= "0101";
constant state6: state:= "0110";
constant state7: state:= "0111";
constant state8: state:= "1000";
constant state9: state:= "1001";
constant state10: state:= "1010";
constant state11: state:= "1011";
constant state12: state:= "1100";
constant state13: state:= "1101";
constant state14: state:= "1110";
constant state15: state:= "1111";
begin

```

Figura 12.4. Máquina de estados para contador de 4 bits

El proceso del divisor de frecuencia se muestra en la figura 12.3, donde puede observarse la entrada de reloj de 50 mHz y a su salida la señal de reloj "clklow" a muy baja frecuencia.

```
process (clk50mhz)
begin
    if (clk50mhz'event and (clk50mhz = '1')) then
        conteo<=conteo+1;
        if (conteo=1) then
            conteo<=0;
            clklow<=not (clklow);
        end if;
    end if;
end process;
```

Figura 12.3. Divisor de frecuencia del sistema emulador de contadores en un monitor

La figura 12.4 muestra la declaración de constantes y la máquina de estados requerida para la obtención del contador de cuatro bits.

```
subtype state is std_logic_vector (3 downto 0);
signal present_state, next_state: state;
constant state0: state:= "0000";
constant state1: state:= "0001";
constant state2: state:= "0010";
constant state3: state:= "0011";
constant state4: state:= "0100";
constant state5: state:= "0101";
constant state6: state:= "0110";
constant state7: state:= "0111";
constant state8: state:= "1000";
constant state9: state:= "1001";
constant state10: state:= "1010";
constant state11: state:= "1011";
constant state12: state:= "1100";
constant state13: state:= "1101";
constant state14: state:= "1110";
constant state15: state:= "1111";
begin
```

Figura 12.4. Máquina de estados para contador de 4 bits

ra 12.3, donde puede observarse la
"clklow" a muy baja frecuencia

nen

tadores en un monitor

e estados requerida para

```

contal: process(clklow)
begin
    if rising_edge(clklow) then
        if (reset='1') then
            present_state <= state0;
        else
            present_state <= next_state;
        end if;
    end if;
end process;

conta2: process(present_state)
begin
    case present_state is
        when state0=>
            next_state <= state1;
        when state1=>
            next_state <= state2;
        when state2=>
            next_state <= state3;
        when state3=>
            next_state <= state4;
        when state4=>
            next_state <= state5;
        when state5=>
            next_state <= state6;
        when state6=>
            next_state <= state7;
        when state7=>
            next_state <= state8;
        when state8=>
            next_state <= state9;
        when state9=>
            next_state <= state10;
        when state10=>
            next_state <= state11;
        when state11=>
            next_state <= state12;
        when state12=>
            next_state <= state13;
        when state13=>
            next_state <= state14;
        when state14=>
            next_state <= state15;
        when state15=>
            next_state <= state0;
        when others=>
            next_state <= state0;
    end case;
    count <= present_state;
end process;

```

Figura 12.4. (continuación) Máquina de estados para contador de 4 bits

ACTIVIDAD COMPLEMENTARIA:

Diseñar un contador binario descendente con visualización en un monitor VGA. Cuando el contador llegue a su límite de cuenta, éste deberá reiniciarse.

Práctica 13.

CAPTURE DE

OBJETIVOS:

El alumno aplica

visualización VC

comis la seña

INTRODUCC

La cámara

cuenta con

ideo, po

NGBS

un e

ni