

UNIVERSIDAD CARLOS III DE MADRID



Ejercicios de VHDL

Circuitos Integrados y Microelectrónica

Luis Entrena

Celia López

Mario García

Enrique San Millán

Marta Portela

Almudena Lindoso

Problema 1

Se pretende generar una señal digital S que tenga las siguientes características:

- Nivel en reposo: BAJO
- En el estado de reposo, si se activa la señal START durante al menos 1 ciclo de reloj, se generará un pulso de nivel ALTO de 10 μ s de duración. Al finalizar el pulso, se volverá al nivel BAJO y se esperará a que se active la señal G.
- Si se activa la señal G durante al menos un ciclo de reloj, se generará un pulso de nivel alto de 500 ns de duración. Al finalizar el pulso se volverá al estado de reposo.
- Las señales START y G sólo se tendrán en cuenta si se producen en el momento adecuado: la señal START sólo se atenderá en el estado de reposo y la señal G sólo se atenderá cuando se haya producido el pulso de START.

La frecuencia de reloj del circuito es de 20 MHz.

Entradas

CLK: reloj del circuito, activo por flanco de bajada

RESET: señal asíncrona que inicializa el circuito, activa por nivel bajo

START: Activa por nivel alto

G: Activa por nivel alto

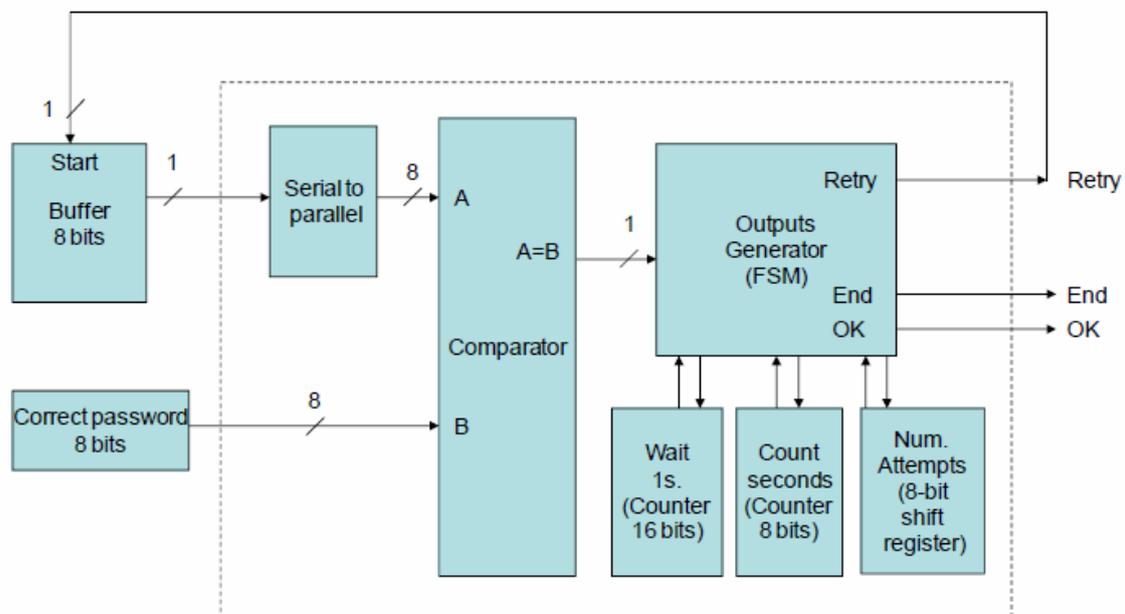
Salidas

S: Señal generada por el circuito

Realizar una descripción de la entidad y la arquitectura del circuito en VHDL

Problema 2

Se quiere diseñar un circuito digital que compruebe si una clave introducida a través de una entrada serie coincide con una clave previamente almacenada en un registro (buffer).



Entradas del circuito:

CLK: reloj del circuito, activo por flanco de subida. Frecuencia: 50 kHz

RESET: señal asíncrona que inicializa el circuito, activa a nivel bajo.

SERIE_IN: señal por la que se reciben bits en serie correspondientes al intento de clave.

CLAVE: entrada de 8 bits por la que se recibe la clave correcta

Salida del circuito:

END: señal que se activa (a nivel alto) cuando se termina el proceso.

OK: señal que se activa (a nivel alto) cuando la clave introducida es correcta.

RETRY: Señal que se activa (a nivel alto) cada vez que se quiere introducir una nueva clave.

El circuito deberá funcionar de la siguiente manera:

El circuito recibe un intento de clave de 8 bits en serie a través de la entrada SERIE_IN (la clave se encuentra almacenada previamente en un buffer externo al circuito).

Por otra parte se recibe la clave correcta de 8 bits (entrada CLAVE).

Con el propósito de comparar la clave correcta con el intento recibido, el circuito convierte el intento de clave recibido en un dato de 8 bits a través de un conversor serie a paralelo. Una vez que se tiene este dato se compara a través de un comparador con la clave correcta.

Con el resultado de la comparación se generan 3 salidas:

La salida END se activa si se ha terminado el proceso, bien porque la clave es correcta, o bien porque se ha alcanzado el número máximo de intentos.

La salida OK se activa si el resultado de la comparación indica que las claves son iguales.

Problema 3

A continuación se muestra la arquitectura de un circuito diseñado en VHDL, y el banco de pruebas de dicho circuito. Suponiendo que todas los puertos y señales son del tipo STD_LOGIC, se pide:

- a) Complete el diseño, en los siguientes aspectos:
 1. Describa la entidad correspondiente a dicho circuito
 2. Complete las declaraciones en el banco de pruebas
 3. Instancie el circuito en el banco de pruebas
 4. Complete las listas de sensibilidad de los procesos
- b) Realice un esquema del circuito utilizando componentes genéricos (biestables, multiplexores, sumadores, puertas lógicas, etc...)
- c) Dibuje un cronograma que se corresponda con la simulación de las entradas y salidas del circuito entre 0 y 180 ns.

```

-- Arquitectura del circuito
architecture a of p1 is
    signal q : std_logic_vector (3 downto 0);
begin
    process
    begin
        if a = '0' then
            q <= (others => '0');
        elsif t'event and t = '1' then
            if b = '1' then
                q <= d;
            elsif c = '1' then
                q <= d(0) & q(3 downto 1);
            else
                q <= q(2 downto 0) & d(0);
            end if;
        end if;
    end process;
    s <= q;

    process
    begin
        case q is
            when "0000" => z <= "00";
            when "1000" | "0100" | "0010" | "0001" => z <= "01";
            when "1100" | "0110" | "0011" => z <= "10";
            when others => z <= "11";
        end case;
    end process;

end a;

```

```

-- Banco de pruebas
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity tb_p1 is
end tb_p1;

architecture a of tb_p1 is
    signal a, t: std_logic := '0';
    -- COMPLETAR DECLARACIONES
begin
    -- INSTANCIAR COMPONENTE
    a <= '0', '1' after 20 ns;
    t <= not t after 10 ns;
    b <= '1', '0' after 60 ns;
    d <= conv_std_logic_vector(3, 4);
    process
    begin
        c <= '0';
        wait for 40 ns;
        c <= '1';
        wait for 60 ns;
    end process;
end a;

```

Cuestión 1

En el ejemplo siguiente existen varios errores. Se pide:

- a) Identifique los errores y clasifíquelos en errores sintácticos y errores de síntesis (no hay error sintáctico, pero no se puede sintetizar correctamente)
- b) Proponga una posible solución corregida del ejemplo

```

architecture a of c2 is
    signal y: integer range 0 to 9;
begin
    if x = '1' then
        y <= 3;
    else
        y <= 4;
    end if;

    process(reset, x)
    begin
        if reset = '1' then
            q <= "0000";
            a <= '0';
        elsif s = '1' then
            a <= x;
        end if;
    end process;

    process(clk)
    begin
        if clk'event and clk = '1' then
            q <= y;
        end if;
    end process;
end a;

```

Question 2

A continuación se muestra la arquitectura de un circuito diseñado en VHDL. Suponiendo que todas las señales que aparecen son puertos del tipo STD_LOGIC, se pide:

- Describa la entidad correspondiente a dicho circuito
- Rellene las listas de sensibilidad de los procesos
- Realice un esquema del circuito utilizando biestables D con entradas asíncronas, multiplexores y puertas lógicas genéricas (AND, OR, etc...)

ARCHITECTURE a OF circuito IS

BEGIN

PROCESS ()

BEGIN

IF a = '0' THEN

p <= "000";

ELSIF b = '1' THEN

p <= "101";

ELSIF clk'EVENT AND clk = '0' THEN

IF q = "111" THEN

p <= k;

END IF;

END IF;

END PROCESS;

```

PROCESS(          )
BEGIN
  IF k(0) = '1' THEN
    x <= '1';
  ELSIF k(1) = '1' OR k(2) = '1' THEN
    x <= '0';
  ELSE
    x <= 'Z';
  END IF;
END PROCESS;
END a;

```

Question 3

A continuación se muestra la arquitectura de un circuito diseñado en VHDL. Suponiendo que todas las señales que aparecen son puertos del tipo STD_LOGIC, se pide:

- Describa la entidad correspondiente a dicho circuito
- Rellene la lista de sensibilidad del proceso
- Realice un esquema del circuito utilizando componentes genéricos (biestables, multiplexores, sumadores, puertas lógicas, etc...)

```

ARCHITECTURE a OF circuito IS
BEGIN
  PROCESS (          )
  BEGIN
    IF a = '0' THEN
      p <= "01";
      r <= '1';
    ELSIF clk'EVENT AND clk = '0' THEN
      IF b = '1' THEN
        p <= p + 1;
      ELSIF c = '1' THEN
        r <= '0';
        p <= "00";
      END IF;
    END IF;
  END PROCESS;
END a;

```


La salida RETRY se activa para indicar que el sistema está preparado para recibir una nueva clave.

Esta salida tiene una sincronización especial. Por motivos de seguridad, cada vez que se introduce una clave incorrecta se espera un determinado tiempo hasta que se permite un nuevo intento. Este retardo irá incrementándose con cada intento fallido. La primera vez que se introduce una clave incorrecta se esperará 1s. La segunda vez se esperará el doble de tiempo (2s), la tercera vez el doble (4s) y así sucesivamente, hasta un número máximo de 8 intentos (con una espera en este caso de 128s). En el caso de alcanzar el número máximo de intentos se activará la salida END.

- a) Describa en VHDL la entidad del circuito completo.
- b) Diseñe en VHDL una entidad y una arquitectura que describan un registro de desplazamiento de 8 bits. Además del reset y del reloj, tendrá una entrada SERIE_IN, una entrada R/L que indique el sentido del desplazamiento (a la derecha si R/L=0 y a la izquierda si R/L=1) y la salida será el valor del registro (8 bits).
- c) Diseñe un proceso que describa un comparador de dos datos de 8 bits. Tendrá una salida A=B (vale 1 si A es igual a B y 0 si son distintos)
- d) Diseñe un proceso que describa un contador de 16 bits. Este contador tendrá como entradas el reloj, el reset, una entrada de enable, y tendrá una salida que se activará cuando el contador haya llegado al valor de cuenta necesario para contabilizar un segundo.
- e) Diseñe mediante un proceso un contador de 8 bits con entradas de reloj, reset y enable y salida de 8 bits con el valor de cuenta del contador.
- f) Utilizando los elementos diseñados en los bloques anteriores realice el diseño de la arquitectura del circuito completo. Puede tener en cuenta las siguientes sugerencias:
 - a. Para el conversor serie a paralelo puede utilizar una instancia del registro de desplazamiento diseñado en el apartado b) y utilizar desplazamientos a la derecha.
 - b. Para el bloque generador de salidas puede utilizar el contador de 16 bits, el contador de 8 bits, y una instancia del registro de desplazamiento del apartado b) para realizar las multiplicaciones x2 (multiplicar por dos es lo mismo que desplazar un bit a la izquierda y completar el bit de menor peso con un cero).

NOTA IMPORTANTE: El circuito debe de ser totalmente síncrono, a excepción de la inicialización mediante reset asíncrono. Comente suficientemente el código o realice diagramas aclaratorios para facilitar la corrección