

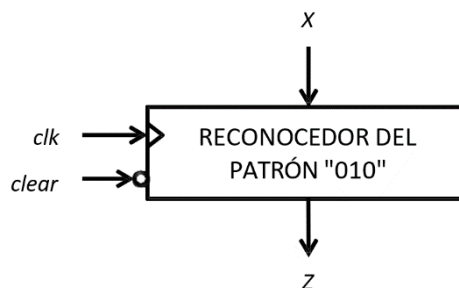


LABORATORIO DE FUNDAMENTOS DE COMPUTADORES 1

PRÁCTICA 3: DISEÑO CON VHDL Y SIMULACIÓN MEDIANTE MODELSIM DEL RECONOCEDOR DE PATRÓN 010

El objetivo de esta práctica es diseñar con VHDL y simular mediante ModelSim un reconocedor solapado del patrón “010”, modelando su comportamiento como una máquina de Moore. El circuito tendrá los puertos mostrados en la figura:

- Una entrada binaria, *X*, por la que reciben los bits en serie.
- Una salida binaria, *Z*, que tomará el valor lógico “1” cuando los últimos tres bits de entrada formen la secuencia “010”, y “0” en cualquier otro caso.
- Una entrada, *clk*, de reloj.
- Una entrada, *clear*, de inicialización asíncrona activa a baja.



A continuación, se explica cómo es el biestable D con reset asíncrono necesario para la implementación. El resto de las puertas (and, or, not) ya se vieron prácticas anteriores.

Biestable D con reset asíncrono

La descripción y simulación de los circuitos secuenciales es ligeramente diferente a la de los circuitos combinacionales que hemos visto hasta el momento. El motivo es que tiene que implementar un elemento de memoria. El código que aparece a continuación describe un biestable D sincronizado mediante flanco de subida de la señal de reloj y reset asíncrono:

```
entity ffd is
  port (
    clk: in bit;
```

```

        rst_n: in bit;
        d: in bit;
        q: out bit;
        qn:out );
end ffd;

architecture comportamiento of ffd is
begin
    p_registro: process (clk,rst_n)
    begin
        if rst_n='0' then
            q <= '0';
            qn<='1';
        elsif rising_edge (clk) then
            q <= d;
            qn<= not d;
        end if;
    end process;
end comportamiento;

```

Vamos a ver a continuación el significado de cada una de las sentencias:

- *if rising_edge (clk) then* describe la sincronización mediante flanco de subida de la señal de reloj.
- *If rst_n='0' then q<='0'*, es el reset asíncrono. Nótese que el reset está antes de la sentencia *if rising_edge (clk) then* y esto indica que es un reset asíncrono porque se activa independientemente de cuando se produzca un flanco de reloj positivo. Nótese también que el reset es negativo, es decir cuando nclear='0' es cuando se carga un cero en el registro. Por último, como el cambio de rst_n no depende de clk debe aparecer en la lista de sensibilidad del proceso.

Simulación del circuito

En esta ocasión vamos a utilizar un archivo en el que se describen las señales de entrada al circuito para realizar la simulación. Este archivo lo hemos llamado *input_signals.do* y se puede descargar del campus virtual.

Vamos a explicar a continuación como es este archivo. El reconocedor tiene tres entradas: clk, rst_n y x. La señal de reloj se describe mediante la siguiente línea de código:

```
force clk -repeat 10 ns 0, 1 5 ns
```

en la que se indica que la señal se debe repetir cada 10 ns, que inicialmente su valor es 0 y que a los 5 ns su valor pasa a ser 1.

La señal de reset negativo debe empezar en 0, para hacer el reset y después subir a 1. En la línea de código que aparece a continuación la subida sucede a los 40ns:

```
force rst_n 0, 1 40 ns
```

Por último, la señal de entrada x es la siguiente:

```
force x 1, 0 65, 1 95,0 125,1 135,0 145
```

en donde la señal se describe mediante pares de datos: el primero representa el valor que toma la señal y el segundo el instante de tiempo de simulación en la que se toma ese valor.

Por ejemplo, el par 1 135 indica que en el instante 135 de simulación la entrada tomará el valor 1.

Para utilizar este fichero arranca la simulación, en la ventana *Transcript* ejecuta *do input_signals.do*, y a continuación ejecuta la simulación.

Desarrollo de la práctica

Completar el fichero *reconocedor_alumnos.vhd*