

Electrónica Digital

Práctica 3

Diseño de Circuitos Secuenciales

Curso 2023-2024



Departamento de
**Ingeniería
Electrónica**

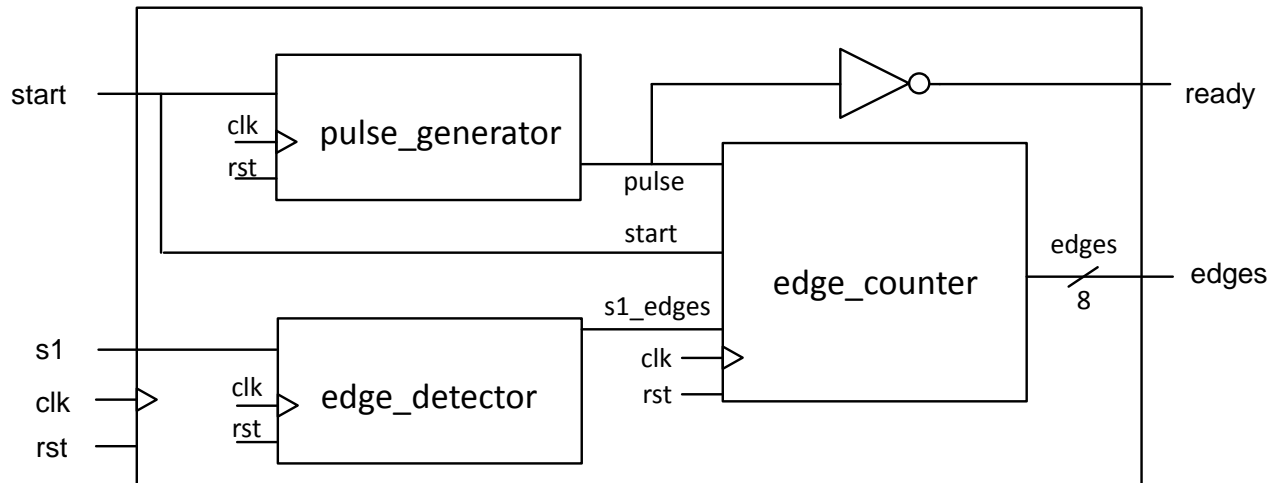
Universidad Politécnica de Madrid



Copyright © 2023, Álvaro Araujo, Pablo Ituero, Octavio Nieto-Taladriz, José Manuel Pardo,
Miguel Ángel Sánchez

En esta práctica se va a implementar un “contador de actividad”, es decir un circuito que va a monitorizar la actividad de una señal (número de cambios de valor) durante 250 nanosegundos.

El esquema de alto nivel del circuito es el siguiente:



El circuito está compuesto de tres bloques:

- El bloque **pulse_generator** genera un pulso de 250 nanosegundos cada vez que el usuario activa la señal **start**.
- El bloque **edge_detector** proporciona a su salida un pulso de un ciclo de reloj de duración por cada cambio de valor detectado en la señal **s1**.
- El bloque **edge_counter** utiliza la información proporcionada por los otros dos bloques para contar el número de cambios de valor producidos durante los 250 nanosegundos posteriores a la activación de **start**.

El sistema recibe, aparte de las señales **s1** y **start**, un reloj, **clk**, y una señal de reset, **rst**, que va a tener un funcionamiento síncrono en todos los flip-flops. Hay dos salidas: **edges**, de 8 bits, que informa del número de cambios de valor y **ready** que avisa cuando hay un valor válido en **edges**.

El concepto de “cambio de valor” depende de la última cifra del código de prácticas del alumno:

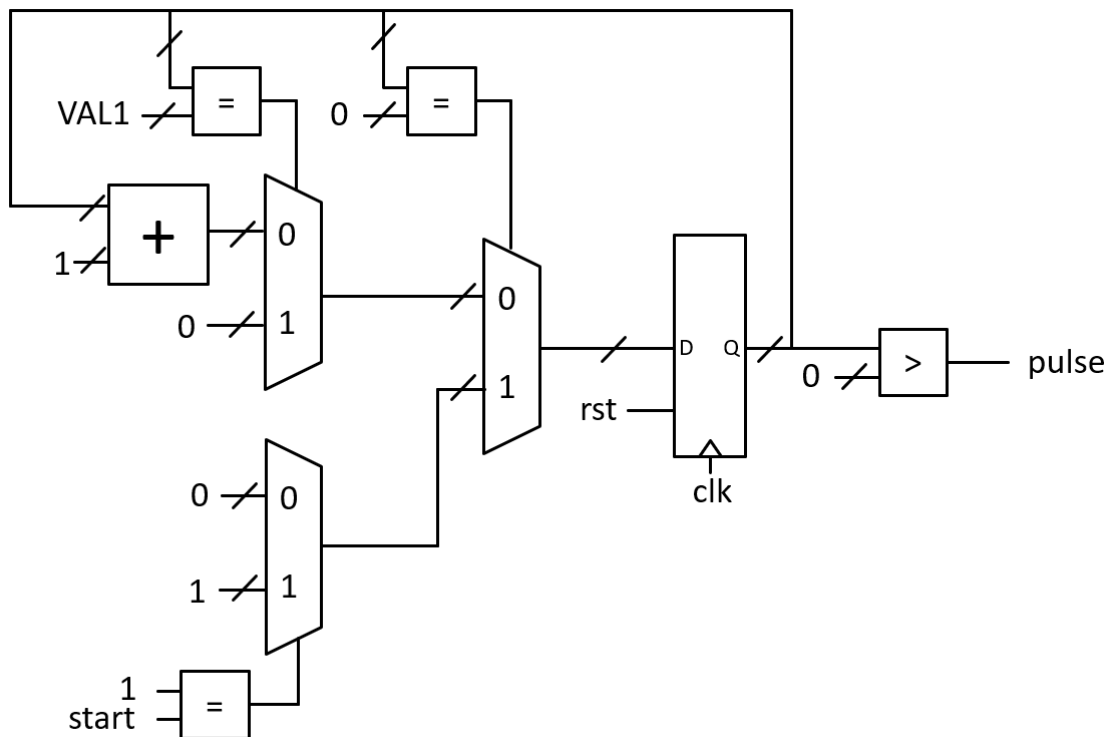
Última cifra	Cambio de valor
1,4,7	Flanco positivo (de 0 a 1)
2,5,8	Flanco negativo (de 1 a 0)
0,3,6,9	Ambos flancos (de 0 a 1 y de 1 a 0)

El período del reloj del sistema también depende la última cifra del código de prácticas:

Última cifra	Periodo del reloj
1,4,7	2 ns
2,5,8	5 ns
0,3,6,9	10 ns

A continuación, se proporcionan los esquemáticos con la estructura interna de cada bloque.

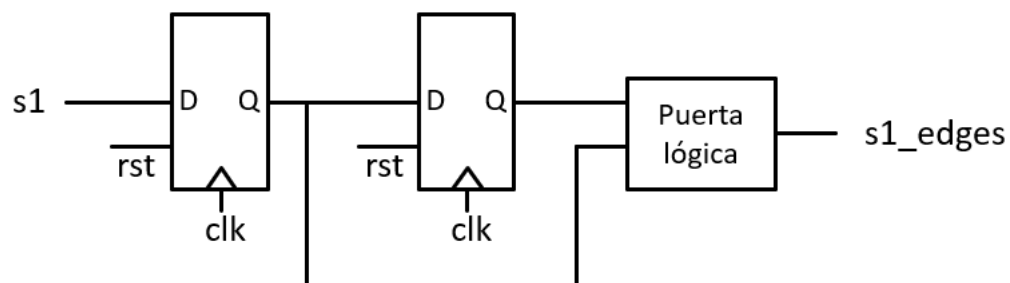
Pulse generator



El valor de **VAL1** lo tiene que determinar cada estudiante para proporcionar un pulso de duración 250 nanosegundos con la frecuencia de reloj establecida por su código.

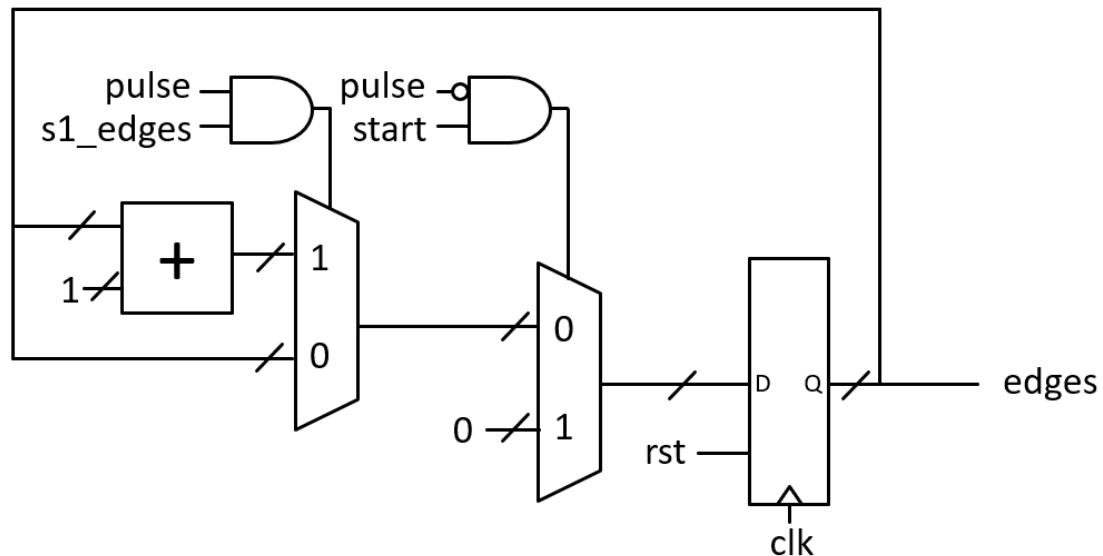
Las interconexiones marcadas con una línea oblicua representan buses de varios bits. La palabra más grande que va a almacenar el flip-flop determina la cantidad de bits necesarios en las líneas de buses del circuito.

Edge detector



La puerta lógica va a depender del tipo de cambio de valor que tiene que detectar cada alumno (i.e flanco positivo, flanco negativo o ambos).

Edge counter



Las interconexiones marcadas con una línea oblicua representan buses de 8 bits.

Especificación de la entrega 3

1. Escribe un fichero VHDL (pulse_generator.vhd) que contenga la descripción del bloque **pulse_generator** con la siguiente declaración de entidad :

```
entity pulse_generator is
  Port ( start : in STD_LOGIC;
        clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        pulse : out STD_LOGIC);
end pulse_generator;
```

2. Escribe un fichero VHDL (edge_detector.vhd) que contenga la descripción del bloque **edge_detector** con la siguiente declaración de entidad:

```
entity edge_detector is
  Port ( s1 : in STD_LOGIC;
        clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        s1_edges : out STD_LOGIC);
end edge_detector;
```

3. Escribe un fichero VHDL (edge_counter.vhd) que contenga la descripción del bloque **edge_counter** con la siguiente declaración de entidad:

```
entity edge_counter is
  Port ( s1_edges : in STD_LOGIC;
        start : in STD_LOGIC;
        pulse : in STD_LOGIC;
        clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        edges : out STD_LOGIC_VECTOR (7 downto 0));
end edge_counter;
```

4. Escribe un fichero VHDL (act_sensor.vhd) con la **descripción estructural** del circuito completo mostrado en la primera figura, utilizando los 3 módulos descritos anteriormente. Este módulo deberá tener la siguiente declaración de entidad:

```
entity act_sensor is
  Port ( s1 : in STD_LOGIC;
        start : in STD_LOGIC;
        clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        edges : out STD_LOGIC_VECTOR(7 downto 0);
        ready : out STD_LOGIC);
end act_sensor;
```

5. Comprime los 4 ficheros **.vhd** de los 4 apartados anteriores en un archivo **.ZIP** con tu código de entregas como nombre de archivo (por ejemplo, para el alumno que tenga como código de entregas el 523, será **523.zip**). No incluyas más ficheros en el archivo .zip (**OJO los que usen WinZip en entorno de Mac**). No se admitirá otro tipo de fichero comprimido (rar o similar).

IMPORTANTE

No se evaluarán:

- Entregas con nombres de ficheros equivocados.
- Módulos con declaraciones de entidad (*entity*) distintas a las descritas para cada módulo.
- Ficheros ZIP que contengan más de los 5 ficheros pedidos o carpetas adicionales.

Realiza la entrega del archivo **.ZIP** a través del buzón de entregas que se habilitará en la plataforma MOODLE de tu grupo de clase.