

# Práctica 2. FSM

Grado en Ingeniería Informática  
Grado en Ingeniería de Computadores

Óscar Garnica  
Facultad de Informática  
Universidad Complutense de Madrid

# Objetivo

1. Diseñar FSM usando VHDL
2. Asentar el manejo de Vivado

# Especificación

- Diseñar el sistema de control de un cerrojo como una FSM tipo Moore.
- Especificaciones detalladas:
  1. Un cerrojo está controlado por una clave de 8 bits que se almacena un registro.
  2. Inicialmente el cerrojo se encontrará desbloqueado.
  3. Al presionar un botón se almacenará la clave y el cerrojo pasa a estar bloqueado.
  4. A partir de entonces, se disponen de tres intentos (pulsando nuevamente el botón) para descubrir la clave, quedando para siempre bloqueado el cerrojo si se supera el número máximo de intentos. Hasta que no se descubra la clave el cerrojo estará cerrado.
  5. Si se acierta la clave en cualquier de los tres intentos, el cerrojo estará desbloqueado y vuelve a permitir insertar nueva clave.
  6. Se debe mostrar el número de intentos restantes en el display 7 segmentos.



# Interfaces

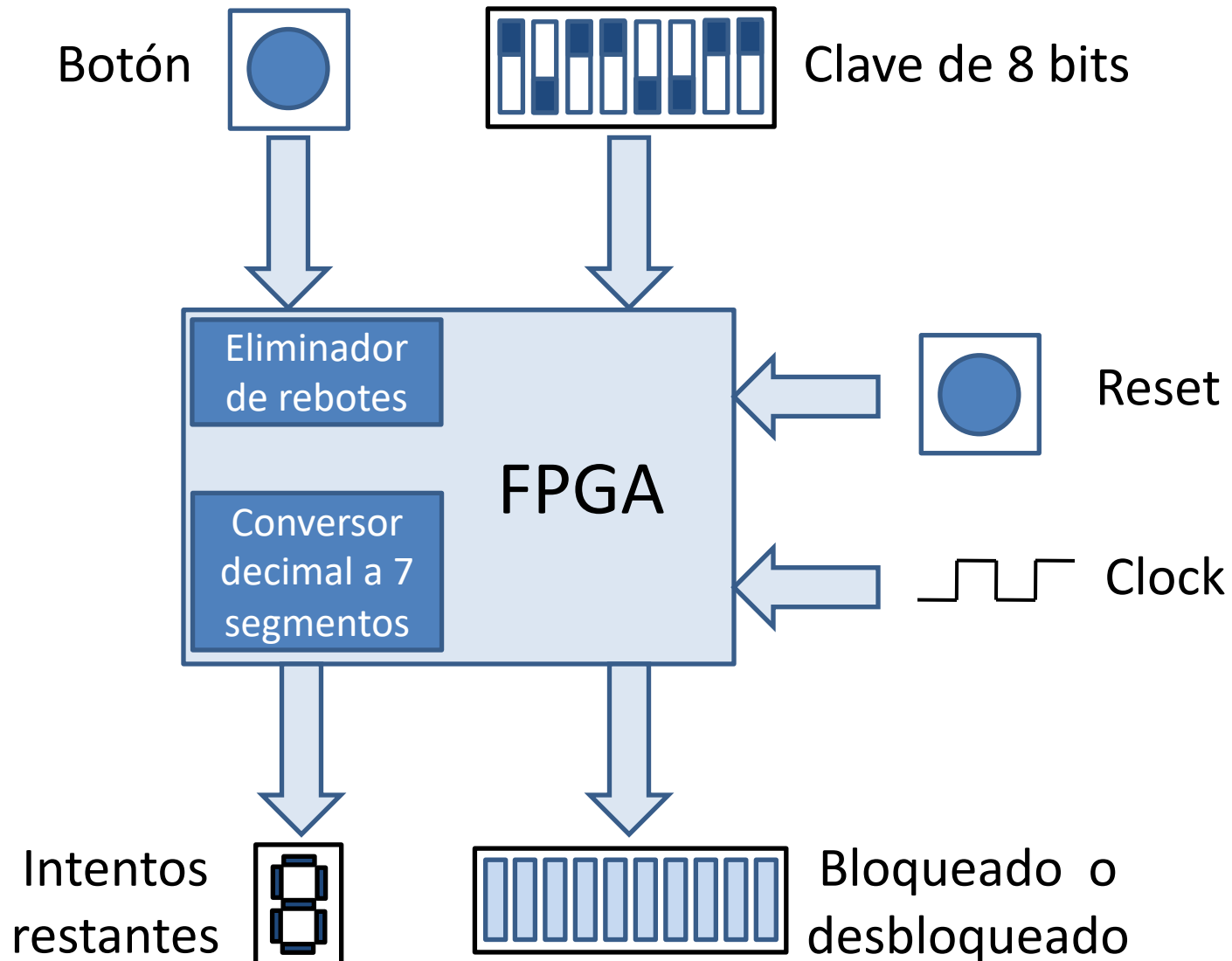
- Reloj de 100 MHz.
- Reset asíncrono activo a nivel alto.

```
entity cerrojo is
  port (clk: in std_logic;
        rst: in std_logic;
        boton: in std_logic;
        clave: in std_logic_vector(7 downto 0);
        bloqueado: out std_logic_vector(9 downto 0);
        ldisplay: out std_logic_vector(6 downto 0);
end entity;
```

# Implementación

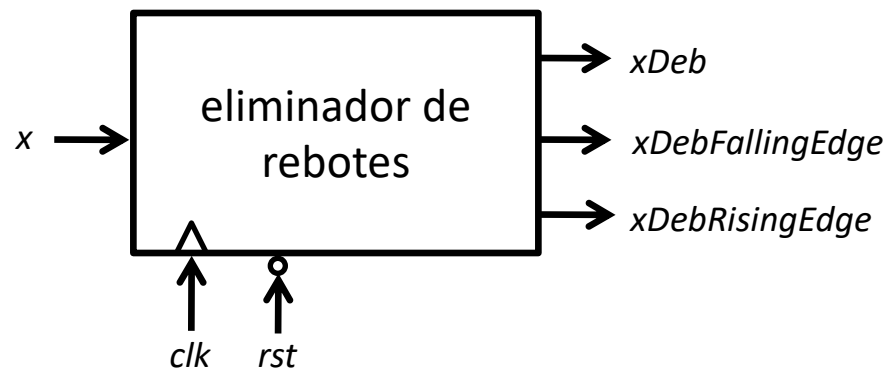
- Para introducir la **clave de 8 bits** utilizaremos el **banco de switches**.
- El **botón para almacenar o introducir la clave** será un **pulsador**.
- Para mostrar que el **cerrojo** se encuentra **abierto** se **encenderán todos los leds** del banco de leds, y por el contrario, cuando el cerrojo esté **bloqueado** el banco de leds estará **apagado**.
- El **número de intentos** restantes se mostrará un **display de 7 segmentos**.
- El **reset** es asíncrono y activo a nivel alto.

# Implementación

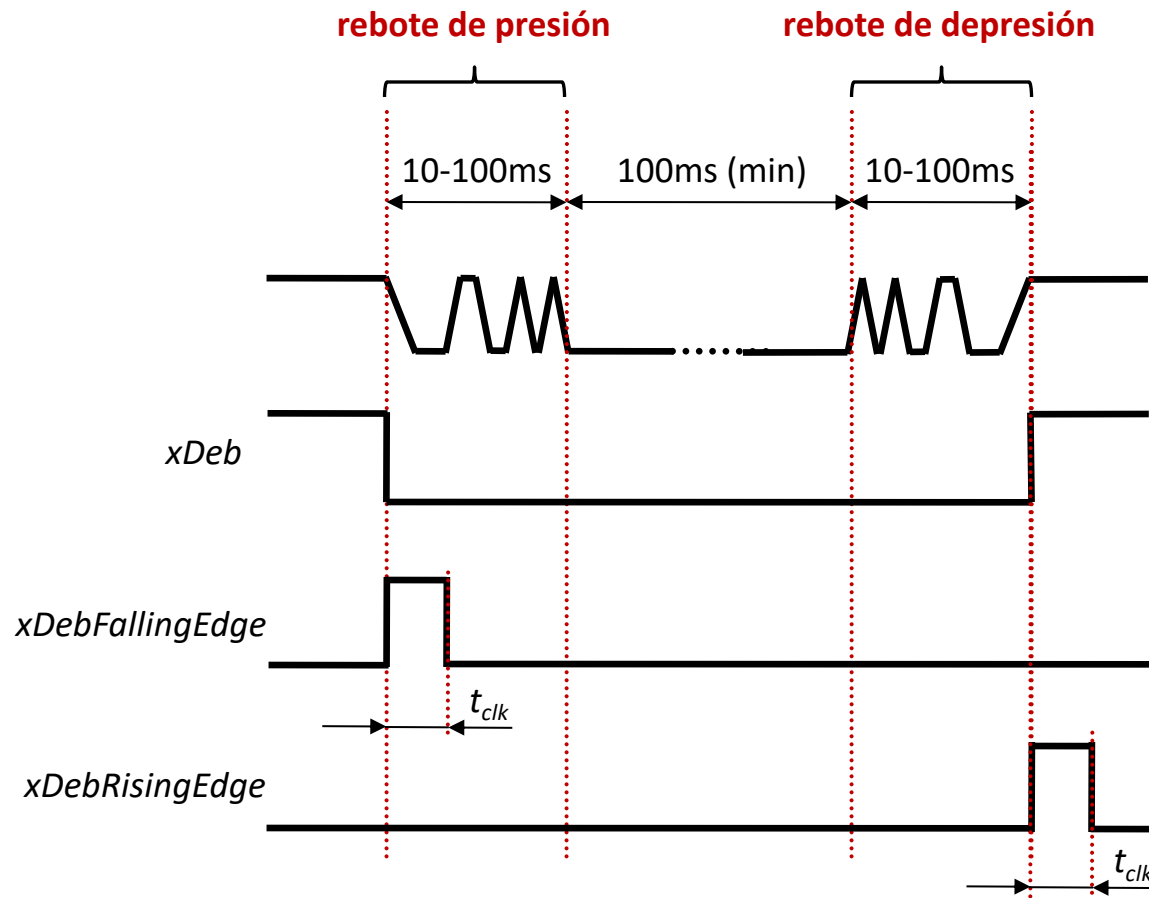


# Eliminador de rebotes

- Toda señal proveniente de un contacto mecánico (p.ej. pulsadores) presenta un **vaivén transitorio** tras cada cambio de estado
- Un **eliminador de rebotes** es un circuito que filtra las transiciones que siguen a todo cambio de estado
- Dicho eliminador de rebotes está definido en '**debouncer.vhd**'



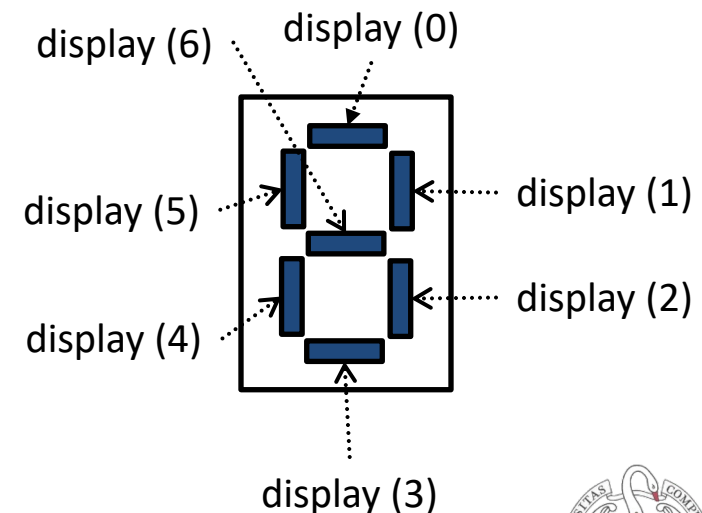
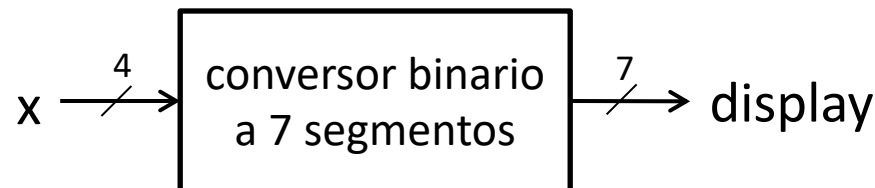
# Eliminador de rebotes





# Conversor binario a 7 segmentos

- Un conversor decimal a 7 segmentos, recibe como entrada un valor en binario sin signo de 0 a 9 y produce el encendido correspondiente de los leds del display para mostrar dicho valor.
- Está definido en '`conv_7seg.vhd`'



# Definir reloj

- **NO usar** el divisor de frecuencia.
- Definir señal de reloj externo

```
## Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
```

Descomentar estas tres líneas borrando los #’s

Asegurarse que la señal de reloj es exactamente la que aparece aquí (clk)

Esta línea declara que el puerto clk es un reloj con un periodo 10.00 ns (el oscilador conectado al pin W5 de la FPGA genera esta señal). Vivado ubicará y enrutará la señal clk para optimizar el Static Timing Analysis (Tema 2)

# Jerarquía del diseño

- El archivo con del diseño de todo el sistema se llamará “sistema.vhd” y la entidad se llamará “sistema”.
- La arquitectura de la entidad “sistema” contiene 3 componentes:
  - La FSM del cerrojo
  - El eliminador de rebotes
  - El conversor a 7 segmentos

# Tareas

- Diseñar la FSM
- Describir la FSM usando tres procesos concurrentes.
- Crear un testbench, simular y verificar el funcionamiento de la FSM.
  - El testbench (en VHDL) proporciona las entradas a la FSM y debéis comprobar que la evolución de los estados es correcta para dichas entradas.
- Sintetizar y verificar los recursos hardware empleados.
  - **NO usar** el divisor de frecuencia.
- Implementar sobre la FPGA.
  - Conectar las entradas y salidas a pines de la FPGA.
  - Conectar el puerto clk de la FSM al pin de reloj de la FPGA.
  - Validar el funcionamiento en placa proporcionando entradas mediante los switches y comprobando que las salidas son correctas.

