

## Práctica 3. Diseño de máquinas de estados en VHDL.

### 1. Introducción.

Una vez conocido el entorno de diseño y la metodología a seguir en descripciones basadas en VHDL, esta sesión de laboratorio pretende profundizar en el uso del lenguaje VHDL. En este caso, analizando las descripciones de máquinas de estados.

### 2. Objetivo de la práctica.

Diseñar un sistema de control de accesos. Se desea controlar el acceso a una zona de alta seguridad, por lo que se requiere:

1. Conocer el número de veces que se ha entrado en el recinto.
2. Conocer quién ha entrado en el recinto (de entre 4 usuarios posibles).
3. Permitir el acceso al recinto si la clave introducida es la correcta.

Para ello, se tiene un sistema de teclado hexadecimal que introduce el número de tecla pulsada (del 1h al Fh).

Las entradas y salidas de que dispone el sistema son:

- a) Entrada de 4 bits proveniente del teclado hexadecimal para indicar el código de la tecla pulsada.
- b) Entrada de reloj (50 MHz).
- c) Entrada de reset.
- d) Entrada de tecla pulsada.
- e) Salida de Activación de apertura de puerta (se enciende un led de la placa DIGILENT).
- f) Salida de ERROR en el código, denegación de acceso (se encienden todos los led de la placa DIGILENT).
- g) Salida de indicación del número de veces que el usuario A ha accedido al sistema.
- h) Salida de indicación del número de veces que el usuario B ha accedido al sistema.
- i) Salida de indicación del número de veces que el usuario C ha accedido al sistema.
- j) Salida de indicación del número de veces que el usuario D ha accedido al sistema.

```
entity acceso is
    Port ( digito : in std_logic_vector(3 downto 0);
          reloj : in std_logic;
          reset : in std_logic;
          tecla : in std_logic;
          A_puerta : out std_logic;
          Error : out std_logic (6 downto 0);
          sal_A : out std_logic_vector(3 downto 0);
          sal_B : out std_logic_vector(3 downto 0);
          sal_C : out std_logic_vector(3 downto 0);
          sal_D : out std_logic_vector(3 downto 0));
end acceso;
```

**ATENCIÓN:** Las salidas de indicación del número de veces que cada usuario ha accedido son de 4 bits porque se visualizan en los display 7 segmentos de la *placa de periféricos* (un usuario en cada display). Para que esto pueda hacerse, es necesario añadir algunas señales más a la *entity*, decide qué señales son y realiza el código apropiado para la correcta visualización.

Se deben leer tres pulsaciones de tecla seguidas para detectar el número de clave. Las claves asignadas a cada persona son:

- Usuario A: 22Ah
- Usuario B: 11Bh
- Usuario C: 00Ch
- Usuario D: FFDh

El indicador de número de veces que cada usuario ha accedido se incrementa indefinidamente hasta que se produce un reset del sistema (o se apaga la alimentación del circuito). Para fijar un límite, digamos que el

Profesores: Alfredo Rosado. Manuel Bataller.

número máximo que se puede almacenar es de 16 (el máximo que nos permite visualizar el display 7 segmentos).

El procedimiento que se debe seguir para acceder es el siguiente:

- a) El sistema espera a recibir una indicación de tecla pulsada (esta indicación es una señal proveniente de un pulsador de la placa, con una duración estimada de 10 milisegundos).
- b) Si hay tecla pulsada, se recoge el valor de la tecla pulsada, no admitiendo nuevas pulsaciones de tecla hasta transcurridos 100 milisegundos (esto hace que se eviten los posibles rebotes de pulsación), esperando posteriormente a que haya una nueva pulsación de tecla. Repitiendo el proceso hasta que haya tres pulsaciones de tecla.
- c) Si el tiempo entre pulsaciones de tecla es mayor de 5 segundos, se anula la captación de tecla y la pulsación se interpretará como la primera tecla de una nueva secuencia.
- d) Una vez recogidas las tres pulsaciones, se compara con las claves de usuario anteriormente descritas, si no coincide se activa la salida de ERROR, y si coincide con algún código, se incrementa el valor del contador de accesos del usuario correspondiente. En cualquiera de los dos casos, la salida estará activada un tiempo de 3 segundos para dar tiempo a visualizar la salida y en ese tiempo, no es posible volver a admitir un nuevo código.

### 3. Tareas a realizar.

1. Diseñar el sistema, de forma que sea capaz de funcionar en la placa disponible en el laboratorio (placa DIGILENT y de periféricos).
2. Decidir de qué modo se asignan las señales de entrada y salida para que toda la funcionalidad descrita se pueda accionar con los recursos de entradas/salidas que dispone la placa (display, pulsadores, interruptores, led, etc.). **NOTA:** NO emplear el teclado hexadecimal de la placa de periféricos.
3. Describir el diseño. Para ello, además de la memoria, se entregarán los ficheros del proyecto Xilinx.
4. Implementar el código en VHDL.
5. Simular el sistema y comprobar su correcto funcionamiento. Generar para ello un banco de pruebas que se incluirá dentro del proyecto a entregar.
6. Probar el sistema en la placa FPGA.
7. **APARTADO ADICIONAL:** Se valorará positivamente si el código se realiza de tal modo que los tiempos (tiempo de tecla muerta de 100ms, tiempo de espera entre pulsaciones de 5s, y tiempo de salida activada de 3s) sean configurables de forma sencilla con la simple variación de un valor o parámetro dentro de todo el código.