

Práctica 2

- Máquinas de estados finitas

Práctica 2 (I)

- Diseño de una máquina de estados finita (FSM)
 - Diseñar un sistema que mida los reflejos del usuario
- La coordinación visomotriz es la habilidad de los ojos y manos para trabajar juntos para la realización de una tarea.
 - Diseñar un circuito que mida cuán rápido puede la mano actuar en respuesta a un estímulo visual

Práctica 2 (II)

- En esta práctica todavía os proporcionaré el test-bench, por lo que la entity de la práctica deberá ser de la siguiente forma:

entity reflejo **is**

port(clk, rst, boton, switch: **in** std_logic;

 luces: **out** std_logic_vector(3 **downto** 0));

end reflejo;

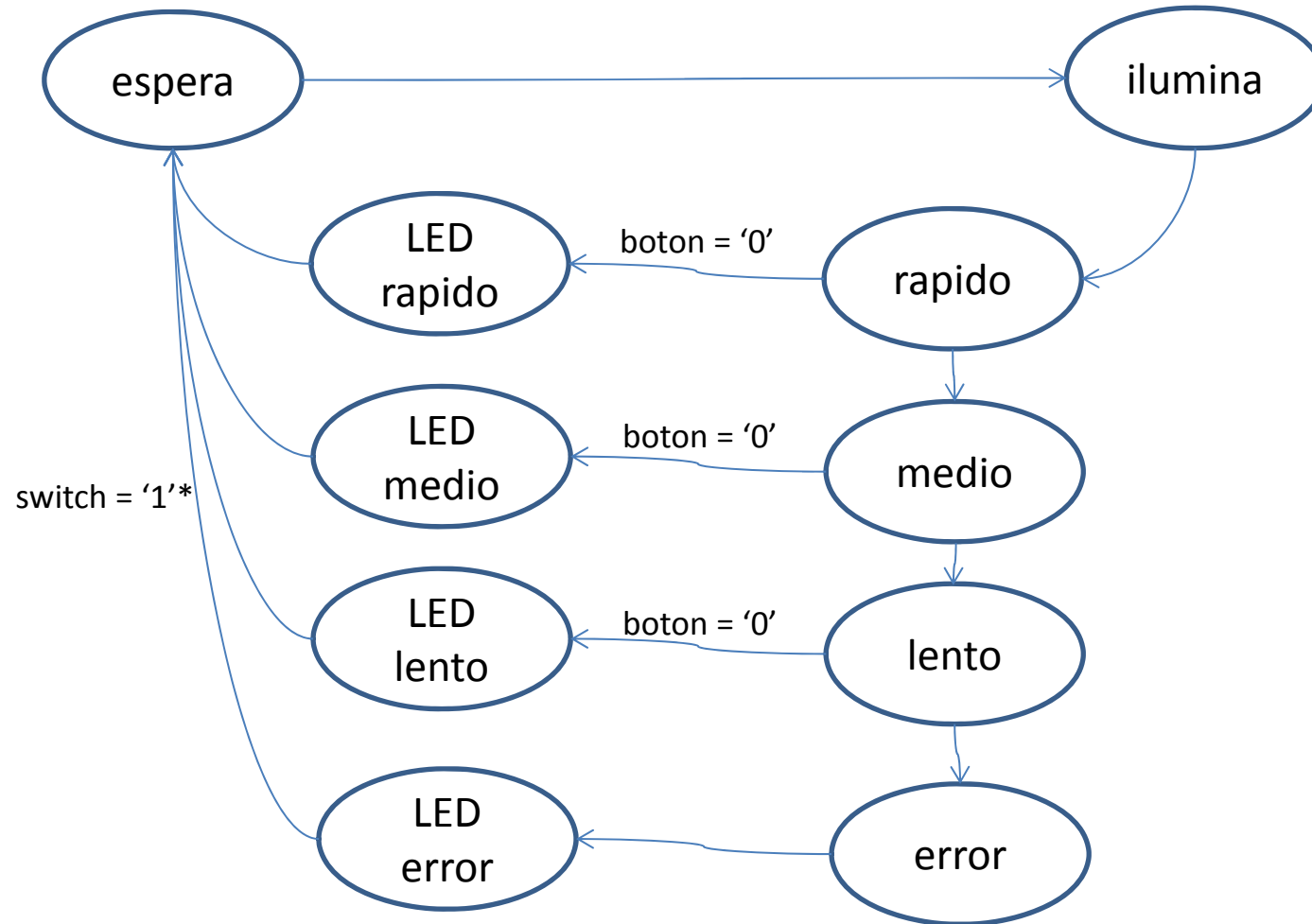
Práctica 2.a (III)

- El sistema tiene como entrada un botón
 - Este botón ofrece, en reposo, un valor de señal igual a '1' y cuando se pulsa ofrece un valor igual a '0'
- Tiene otra entrada que es switch y que servirá para poner el circuito en marcha
 - La primera vez no será necesario tocar esa entrada
 - Se pondrá a '1' después de que se mida la coordinación para volver a empezar
- Tiene 4 LEDs (luces) como salida
 - Un LED que se enciende durante el estado “ilumina”. Una vez se apague el usuario deberá pulsar el botón para medir la coordinación y deberá mantenerlo pulsado hasta que los otros 3 leds muestren la medida tomada
 - 3 LEDs que indican la medida tomada
 - 011 Error
 - 100 Respuesta lenta
 - 101 Respuesta media
 - 110 Respuesta rápida

Práctica 2.a (IV)

- El sistema comienza en un estado de espera. Siempre que se active el reset el sistema vuelve a ese estado de espera
- Desde ese estado se pasa automáticamente a un estado que enciende el LED
- Desde el estado que enciende el LED el sistema pasa secuencialmente por 4 estados. Se saldrá de esta secuencia si se pulsa el botón o si se llega al último estado
 - 1er estado: respuesta rápida
 - 2º estado: respuesta media
 - 3er estado: respuesta lenta
 - 4º estado error
- Cada uno de estos cuatro estados, tiene asociado un estado para indicar la iluminación de los LEDs correspondientes
- El sistema permanecerá en estos estados de iluminación hasta que la entrada switch pase a 1, desde donde pasará al estado espera y se podrá realizar una nueva medición

Diagrama de estados



* switch tiene que ser igual a '1' para todas las transiciones a espera

Práctica 2.a (III)

- Diseñar el sistema como una FSM tipo Moore
- Descargar el fichero Testbench simu_2a.vhd del Campus Virtual
- Simular y comprobar su correcto funcionamiento
 - El reloj para la simulación es el de **100MHz (sin divisor de frecuencias)**. Cuando se implemente en la FPGA se utilizará un reloj de **1Hz (con divisor de frecuencias)** y habrá que mantener pulsado el botón desde que se enciende el primer led hasta que se enciendan los LEDs de medición
- Implementar sobre la FPGA

Calificación (I)

- El estudiante posee el enunciado de la práctica con al menos 6 días de antelación. Debe acudir al laboratorio con la práctica estudiada desde casa
- El estudiante debe responder a una serie de preguntas tipo test sobre la parte 2.a el día anterior a la realización de la práctica
 - Para responder a las preguntas el estudiante cargará el fichero de simulación `simu_2a.vhd` y responderá a las preguntas de Moodle sobre ese apartado
- La práctica 2 presenta una parte avanzada
- La práctica 2 no se recupera

Calificación (II)

- En el laboratorio mostrar la simulación e implementación de la práctica 2.a
 - Recordar configurar correctamente el fichero pines.ucf con todos los puertos de entrada y salida de la entity
- Mostrar simulación e implementación de la parte avanzada
 - Para simular esta parte, descargarse el fichero simu_2b.vhd