

## SISTEMAS DIGITALES

### EJERCICIOS DE ENTRENAMIENTO – SECUENCIALES – (17-18)

#### Mayo 2012

Un sistema digital de seguridad que controla la apertura de una puerta con clave secreta consta de dos pulsadores I (izquierdo) y D (derecho). Se desea que la secuencia de pulsado para apertura de puerta sea D D I, lo que significa que para abrir la puerta se debe pulsar de la siguiente manera: D - no pulsar - D - no pulsar - I. El sistema debe ser síncrono y los botones NO ESTÁN REGISTRADOS, es decir, el sistema se debe diseñar sin detector de flanco de botón, previendo la posibilidad de que cada botón pueda estar pulsado durante un número indefinido de periodos de reloj.

a) Proponer el diagrama de estados de la máquina de Moore que active un relé de apertura mediante nivel alto cuando se pulse la secuencia indicada. El sistema debe incluir las siguientes características:

- La situación de no pulsar debe ser considerada como una posible señal a tener en cuenta.
- Si se pulsan los dos botones a la vez, el sistema lo interpreta siempre negativamente y esto lleva a su reinicio.
- El nivel alto que alimenta el relé se activa al pulsar la última tecla "I" de la secuencia, y permanece activado solo mientras dure la pulsación de la tecla I.

b) Escribir el código VHDL del sistema (entidad y arquitectura) que permitiría codificar el diagrama de estados propuesto en forma de máquina síncrona (incluir señal de RESET y reloj), con dos entradas externas I y D y una salida Z que activa a nivel alto el relé de apertura.

-----

a) Un sistema digital recibe por su única entrada un tren indefinido de bits, y debe generar un nivel alto cuando se detecte la secuencia 01 o 10 (donde el 2º bit de cada secuencia puede constituir el 1º de la otra). Proponer el diagrama de estados de la máquina de Moore que realiza dicha función.

b) Sintetizar un circuito correspondiente al diagrama de estados obtenido, utilizando biestables tipo D y las puertas lógicas necesarias.

#### Junio 2012

Un sistema calefactor está dotado de un motor y dos sensores de temperatura T1 y T2 (donde  $T1 < T2$ ), que informan con un nivel alto si la temperatura T es mayor que la de referencia del sensor. El sistema debe funcionar de la siguiente forma:

- Si T es mayor que cualquiera de las dos temperaturas de referencia de los sensores, entonces el motor del calefactor está apagado.
- Si desde esta situación la temperatura desciende a un valor  $T1 < T < T2$ , el calefactor continuará apagado.
- Si la T sigue descendiendo, cuando  $T < T1$ , el motor del calefactor se deberá activar.
- Así el calefactor previsiblemente aumentará la temperatura de forma que cuando ésta ascienda y esté en el rango  $T1 < T < T2$ , el calefactor continuará activado.
- Desde la anterior situación, solo cuando  $T > T2$  entonces el motor del calefactor se apagará.

Se pide proponer un diagrama de estados de la máquina de Moore que describe este sistema.

Implementar el circuito con biestables JK, y con el criterio de minimizar coste, es decir, utilizando el menor número posible de biestables y puertas. Para ello aprovechar los posibles indiferentes en la tabla de transiciones y utilizarlos en la simplificación de funciones lógicas.

#### Mayo 2013

Una grúa posee dos luces de advertencia de movimiento, que indican si se mueve a la izquierda o a la derecha. Estas luces se activan desde cabina con una palanca horizontal, que tiene una posición central de reposo. De forma general, si se mueve la palanca a la izquierda, se enciende la luz de advertencia de movimiento a la izquierda, y si se mueve la palanca a la derecha, se enciende la luz de advertencia de movimiento a la derecha. La señal de la palanca posee detector de flanco para sus dos posiciones extremas, de manera que al desplazarse a cualquier extremo envía niveles altos solamente durante un periodo de reloj.

El encendido de luces está sujeto a las siguientes características:

- Si las luces están apagadas y se mueve la palanca hacia un lado, la luz correspondiente de ese lado permanecerá encendida, tanto mientras se sujete la palanca en ese estado, como también cuando se devuelve la palanca a su posición de reposo.
  - Si queremos apagar esa luz, deberemos mover la palanca hacia ese lado otra vez. En ese instante, se apagará esa luz, y permanecerá apagada cuando se devuelve la palanca a su estado de reposo.
  - Si la luz de un lado (p. ej, izquierdo) está encendida con la palanca en ese lado, y movemos la palanca hacia el lado contrario (el derecho), entonces pasa a encenderse la luz de este lado (derecho) y se apaga la anterior (izquierdo). En esta situación, la luz (derecha) permanecerá encendida conforme a las condiciones de los dos anteriores puntos.
  - Obviamente, nunca se encenderán las dos luces a la vez y no se puede mover la palanca a derecha e izquierda simultáneamente.
- a) Diseñar el diagrama de estados de la máquina de Moore que verifica este comportamiento.
  - b) Escribir el código VHDL del sistema (entidad y arquitectura) que permitiría codificar el diagrama de estados propuesto en forma de máquina síncrona (incluir señal de RESET y reloj).

### **Junio 2013**

1.- (2.5 puntos) Una máquina secuencial síncrona de Mealy (o en su defecto, de Moore) posee una entrada IN (por la que entra un tren de bits) y una salida Z, que opera de la siguiente forma: cuando se detecta la secuencia 11, Z se pone a 1, manteniendo este valor hasta detectar la secuencia 00, en cuyo caso Z pasa a tomar valor 0 hasta que llegue una nueva secuencia 11.

Se pide:

- a) Dibujar el diagrama de estados de la máquina de Mealy (o en su defecto, de Moore) del sistema.
- b) Implementar la tabla de transiciones, excitaciones y salidas.
- c) Ofrecer la expresión simplificada de las señales de excitación a la entrada de los biestables tipo D empleados.

(1.5 puntos) Escribir el código VHDL del sistema completo (entidad y arquitectura) del ejercicio anterior (incluir entrada, salida, y señal de RESET y reloj). Nota: Es una entrada sincronizada; no es un pulsador, luego no necesita detector de flanco de botón.

### **Junio 2014**

Por un bus de 2 bits (D1 y D0) se envía información sincronizada con una señal de reloj CLK. Se ha convenido que la información sea correcta siempre que un par de bits 00 NO suceda a un par 11. En ese caso, un circuito digital tipo máquina de Mealy (o en su defecto, de Moore) debe advertir del error con un nivel alto por su salida Zout durante un periodo de reloj.

- a) Diseñar el diagrama de estados de la máquina de Mealy (o en su defecto, de Moore) de entradas X e Y y salida Zout que garantiza este comportamiento. (1 pto)
- b) Sin llegar a pintar el circuito, obtener las funciones lógicas combinacionales que dan lugar a la salida Zout, y a las entradas de los biestables JK necesarios para implementarlo. Se pide tabla de transiciones, excitaciones y salidas del sistema, y tablas de Karnaugh simplificadas para Zout, J, K (2 ptos).
- c) Escribir el código VHDL del sistema (entidad y arquitectura) que permitiría codificar el diagrama de estados propuesto en forma de máquina síncrona (incluir señales de RESET y reloj CLK). Para el bus de entrada, considerar una señal de tipo "std\_logic\_vector" de 2 bits. (1 pto)

### **Mayo 2015**

El motor de una grúa permite tres acciones: parar (P), subir (S) o bajar (B). Las señales P, S y B son distintas e independientes, y gobiernan el accionamiento del motor (obviamente no podrán estar activas simultáneamente). El movimiento del motor se maneja mediante dos botones, A y B, que envían un nivel alto al ser pulsados. La señal de estos pulsadores se presupone registrada, es decir, pasa por un detector de flanco antes de acceder a la lógica secuencial de control, que opera con un reloj de periodo muy rápido. Esto significa que al pulsar un botón sólo se envía un nivel alto durante un periodo de reloj, aunque se mantenga pulsado. Hay que despulsar y volver a pulsar para enviar de nuevo un nivel alto durante un periodo de reloj. La lógica de control deberá funcionar de la siguiente manera:

- i) Partiendo de una situación inicial de motor parado sin pulsar los botones, ocurrirá que si pulsamos A, la grúa sube. Situación que se mantiene al despulsar A, hasta que se vuelva a pulsar A, en cuyo caso la grúa se detiene. Si volvemos a pulsar A se reanuda el ascenso y un nuevo pulsado detiene la grúa. Y así sucesivamente.
- ii) Si desde una situación de parado pulsamos B, el motor desciende sujeto a las mismas condiciones de pulsado que operaban con el ascenso.
- iii) Si la grúa está descendiendo, y se pulsa A, inmediatamente se pone a subir. Este ascenso está sujeto a las condiciones del punto i. Y viceversa: si sube y se pulsa B, inmediatamente se pone a descender (descenso igualmente sujeto a las condiciones del punto ii).
- iv) Dado que las señales de los pulsadores están registradas, la situación en que ambos botones se presionan y envían un nivel alto a la vez, es muy rara. En caso de darse, el motor siempre se debe parar.

(No preocuparse por la situación en que la grúa llega al final de su recorrido permitido, ya que se activaría una señal adicional que para el motor y es ajena a la lógica secuencial).

Se pide:

- a) Diseñar el diagrama de estados de la máquina de Moore que regula la lógica de control en función de las señales registradas de los pulsadores. (1 pto)
- b) Sin llegar a pintar el circuito, obtener las funciones lógicas combinacionales que dan lugar a las tres salidas, y a las entradas de los biestables T necesarios para implementarlo. Se pide tabla de transiciones, excitaciones y salidas del sistema, y tablas de Karnaugh simplificadas para P, S y B, así como para las entradas de los biestables T (2 ptos).
- c) Escribir el código VHDL (incluir entidad y arquitectura) que permite codificar solamente el diagrama de estados propuesto en el apartado a) en forma de máquina síncrona que opera con las señales registradas de los pulsadores (incluir señales de RESET y reloj CLK). (1 pto). (No incluir los procesos de registro de botón)

### **Mayo 2016**

(2.5 puntos). Las luces intermitentes de giro de un vehículo se controlan mediante una palanca que puede situarse en tres posiciones: izquierda, central, derecha. El sistema de activación de los intermitentes debe obedecer las siguientes reglas:

- Desde una situación en que las luces están apagadas y la palanca en posición central (reposo), entonces: si la palanca se desplaza a la derecha se encenderán los intermitentes derechos, y si lo hace a la izquierda se encenderán los intermitentes izquierdos. Y así se mantendrán mientras la palanca esté desplazada. Una vez encendidos los intermitentes correspondientes, cuando la palanca retorna a su posición central dichos intermitentes permanecen encendidos.
- No pueden estar los intermitentes izquierdos y derechos luciendo a la vez.
- Desde una situación en que unos intermitentes están encendidos y la palanca en posición central (reposo), entonces: si la palanca se desplaza en el sentido de los intermitentes encendidos, estos se apagan y así quedarán cuando la palanca retorne a su posición de reposo. Pero si la palanca se desplaza en el sentido opuesto a los intermitentes encendidos, estos se apagarán y pasarán a encenderse aquellos en cuya dirección se ha desplazado la palanca, y permanecerán encendidos

mientras la palanca se mantenga así desplazada y también cuando la palanca retorne a su posición de reposo.

a) Diseñar una máquina de estados de Moore que resuelva este sistema (plantear el diagrama de estados). Para ello, considerar que el movimiento de la palanca está codificado con DOS bits. Además, incluir obligatoriamente que el desplazamiento de la palanca está registrado, es decir, cuando la desplazamos, por ejemplo, en sentido derecho, la señal del bit derecho envía un solo '1' durante un periodo de reloj aunque la palanca se mantenga desplazada a la derecha.

b) Sintetizar el circuito correspondiente mediante biestables T, donde se han de obtener las funciones lógicas simplificadas de las entradas de los biestables, y de las salidas (intermitentes izquierdo y derecho) del sistema. Se valorará utilizar el menor nº de biestables. NO ES NECESARIO PINTAR EL CIRCUITO.

-----

(2.5 puntos). Se debe codificar mediante VHDL el sistema descrito por el diagrama de estados propuesto en el anterior ejercicio. En este caso, se ha de respetar que:

- Las entradas del sistema se adaptan al movimiento de la palanca (para lo cual se utilizan DOS bits).  
- Las salidas del sistema son DOS leds, uno para indicar el giro a la derecha, y otro para indicar el giro a la izquierda. EN ESTE EJERCICIO NO ES NECESARIO INCLUIR EL EFECTO DE ENCENDIDO Y APAGADO DEL INTERMITENTE.

- El sistema está gobernado por una señal de reloj (CLK) muy rápida, y debe incluir un RESET.

Codificar obligatoriamente el registro del movimiento de la palanca, es decir, cuando la desplazamos, por ejemplo, en sentido izquierdo, la señal del bit izquierdo deberá enviar un solo '1' durante un periodo de reloj aunque la palanca se mantenga desplazada a la izquierda.

### **Mayo 2017**

El sistema que mueve una pala agitadora del líquido de una cubeta está dotado de un motor que puede mover dicha pala de derecha a izquierda o de izquierda a derecha, entre dos topes. El movimiento de la pala, por ejemplo de izquierda a derecha, se mantiene hasta que la pala entre en contacto con el tope derecho, en cuyo caso un sensor debe enviar una señal (SD) al motor para que éste cambie el sentido del desplazamiento de la pala, de derecha a izquierda. Todo el tiempo transitorio que la pala esté en contacto con el tope derecho, el sensor enviará la señal SD = '1' (no hay detector de flanco aquí). Cuando el motor cambie el sentido del movimiento, éste se prolongará hasta que la pala entre en contacto con el tope izquierdo, en cuyo caso el sensor de este tope enviará la correspondiente señal (SI) al motor para que cambie el sentido del desplazamiento. Y así sucesivamente. Todo el tiempo que la pala esté en contacto con el tope izquierdo, el sensor enviará la señal SI = '1' (tampoco hay detector de flanco aquí) Asumir que el motor se mueve continuamente, ya sea en un sentido o en otro, es decir, nunca está parado.

Se pide: a) Proponer un diagrama de estados de una máquina que describa este sistema a partir de las señales de los sensores de los topes (SD y SI), y de una única señal de accionamiento del motor (a nivel alto para desplazar la pala hacia la derecha y a nivel bajo para desplazar la pala a la izquierda). b) Implementar el circuito con biestables D, y con el criterio de minimizar coste, es decir, utilizando el menor número posible de biestables y puertas. Para ello aprovechar las posibles situaciones indiferentes en la tabla de transiciones (en caso que las haya) y utilizarlas en la simplificación de funciones lógicas.

-----

Un sistema de apertura de puerta con seguridad posee dos botones (pulsadores P y Q) y una salida (Z). Los pulsadores ya vienen dotados de un sistema de detector de flanco o registro de botón, por el cual al pulsarlos (y mientras se mantengan pulsados) solo se envía un '1' durante un periodo de reloj.

Codificar mediante VHDL el circuito secuencial que active una señal a nivel alto cuando se pulse consecutivamente la secuencia Q – P – P – Q, y mantenga dicha señal a '1' hasta que se pulse una nueva tecla, en cuyo caso el sistema se reinicia.

a) Dibujar el diagrama de estados del sistema.

b) Para el código VHDL, suponer que:

- Las señales P y Q se consideran ya registradas de antemano en otro proyecto y no hace falta codificar su detector de flanco.
- Considerar la situación de no pulsar (señal NP) como una entrada más del sistema.
- El sistema está gobernado por un CLK muy rápido.
- Si se produjera la remota situación de pulsar exactamente dos teclas a la vez, el sistema reaccionaría considerándolo una secuencia equivocada y situándose en el estado inicial.
- Se debe incluir un RESET tal que al pulsarlo el sistema se mantiene en el estado inicial.

### Junio 2017

(2 puntos) Por la entrada única de un sistema se reciben bits de forma síncrona. Diseñar el circuito que genera un '1' a la salida cuando se detecta un cambio en el valor del bit de entrada, o sea, cuando hay un paso de '0' a '1' o de '1' a '0'.

Se pide: a) Proponer un diagrama de estados de una máquina que describa este sistema. b) Implementar el circuito con biestables T, y con el criterio de minimizar coste, es decir, utilizando el menor número posible de biestables y puertas. Para ello aprovechar las posibles situaciones indiferentes en la tabla de transiciones (en caso que las haya) y utilizarlas en la simplificación de funciones lógicas. (Nota: no hace falta dibujar el circuito, solamente expresar las funciones lógicas necesarias)

-----

Un sintetizador de sonido electrónico con 2 teclas (pulsadores) genera, respectivamente, señales de dos periodos distintos en la banda de audio, que denominaremos Sig\_1 y Sig\_2. Las entradas que activan estas señales son los 2 pulsadores BTN1 y BTN2. La salida será única y obedece a los siguientes principios:

- Si no pulsamos ninguna tecla, el sistema debe permanecer en silencio.
  - Si pulsamos solamente una tecla, BTN<sub>x</sub>, deberá escucharse la señal Sig<sub>X</sub> mientras la mantengamos pulsada.
  - Cuando se pulsan dos teclas, no se superpondrán las señales, sino que sonará el sonido de la última tecla pulsada (mientras ambas lo estén). Si nos encontramos en la situación de no pulsar, y diera la casualidad (enormemente improbable) de que pulsáramos las dos teclas exactamente a la vez, entonces, convenimos que NO SONARÍA ningún sonido a la salida.
- a) Escribir el diagrama de estados del sistema (se acepta de Moore o de Mealy).  
 b) Escribir, en coherencia con el anterior diagrama, el código VHDL completo (entidad y arquitectura) del sistema, incluyendo las señales de reloj (CLK) y RESET. (Las señales Sig<sub>1</sub> y Sig<sub>2</sub> basta con declararlas internamente donde corresponda sin necesidad de generar sus procesos).

### **Problema especial:**

“Querido amigo: Al poco tiempo de comprar esta vieja mansión tuve la desagradable sorpresa de comprobar que está hechizada con dos sonidos de ultratumba que la hacen prácticamente inhabitable: un canto picaresco y una risa sardónica. Aún conservo, sin embargo, cierta esperanza, pues la experiencia me ha demostrado que su comportamiento obedece a ciertas leyes, oscuras pero infalibles, y que puede modificarse tocando el órgano y quemando incienso. En cada minuto, cada sonido está presente o ausente. Lo que cada uno de ellos hará en el minuto siguiente depende de los que pasa en el minuto actual, de la siguiente manera: El canto conservará el mismo valor (presente o ausente) salvo si durante el minuto actual no se oye la risa y toco el órgano, en cuyo caso el canto toma el valor opuesto. En cuanto a la risa, si no quemo incienso, se oirá o no según que el canto esté presente o ausente (de modo que la risa imita al canto con un minuto de retardo). Ahora bien, si quemo incienso la risa hará justamente lo contrario de lo que hacía el canto. En el momento en que te escribo estoy oyendo a la vez la risa y el canto. Te quedaré muy agradecido si me dices que manipulaciones de órgano e incienso debo seguir para establecer definitivamente la calma.”

Construir un autómata síncrono con flip-flops JK que simule el comportamiento de la mansión, proporcionando una salida activa durante la situación de calma (sin risa ni canto). ¿Qué tendría que hacer mi amigo para restablecer la calma y mantenerla?

## ANEXO: Plantilla VHDL

### Declaración de entidad

```
entity <entity_name> is
  port (
    <port_name_1> : <mode> <type>;
    ...
    <port_name_n> : <mode> <type>
  );
end <entity_name>;
```

### Declaración de arquitectura

```
architecture <arch_name> of <entity_name> is
  -- zona declarativa
begin
  -- zona de sentencias
end <arch_name>;
```

### Para declarar nuevos tipos enumerados:

```
type <type_name> is (<nombre1>, <nombre2>, ...);
```

### Algunas sentencias condicionales (en procesos):

#### Setencia if

```
if <condition> then
  <statement>
elsif <condition> then
  <statement>
else
  <statement>
end if;
```

#### Sentencia CASE

```
case (<2-bit select>) is
  when "00" =>
    <statement>;
  when "01" =>
    <statement>;
  when "10" =>
    <statement>;
  when "11" =>
    <statement>;
  when others =>
    <statement>;
end case;
```

### Alguna sentencia condicional concurrente:

```
<name> <= <expression> when <condition> else
  <expression> when <condition> else
  <expression>;
```