

UNIVERSIDAD CARLOS III DE MADRID

Electrónica Digital

Manual de Prácticas



Autores:

Celia López Ongil

Marta Portela García

Susana Patón Álvarez

Almudena Lindoso Muñoz

DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA

ESCUELA POLITÉCNICA SUPERIOR



Índice

1.	Introducción a las Prácticas de Fundamentos de Electrónica Digital	2
2.	Descripción del Sistema a diseñar	4
3.	Práctica 1. Diseño de Circuitos Combinacionales (I). Diseño con componentes discretos	5
4.	Práctica 2. Diseño de Circuitos Combinacionales (II).....	10
5.	Práctica 3. Diseño de Circuitos Secuenciales (I)	12
6.	Práctica 4. Diseño de Circuitos Secuenciales (II).....	15
7.	Práctica 5. Implementación del Sistema Completo	18
	Lista de Material	21



1. INTRODUCCIÓN A LAS PRÁCTICAS DE FUNDAMENTOS DE ELECTRÓNICA DIGITAL

Le aconsejamos que lea este manual con atención. Las prácticas de laboratorio son, junto con su trabajo personal, una de las mejores herramientas para la comprensión de la asignatura. Están diseñadas para *ayudarle* a consolidar sus conocimientos y desarrollarlos en la práctica. Para sacar mayor provecho de las prácticas que le proponemos es necesario dedicarles algo de tiempo antes de acudir al laboratorio. Si tiene dudas mientras prepara las prácticas no dude en consultar a sus profesoras de laboratorio.

El contenido de las prácticas aumenta de complejidad progresivamente, acorde a los contenidos teóricos de la asignatura. La primera práctica la dedicaremos a la implementación de un sistema combinatorial sencillo empleando componentes discretos. Con anterioridad a la sesión de prácticas cada grupo deberá preparar el diseño del circuito y una memoria técnica. La sesión de laboratorio se dedicará a la implementación del circuito diseñado y su verificación. Al finalizar la sesión, un profesor de laboratorio evaluará su trabajo.

En los enunciados de prácticas aparecen listados los puntos mínimos que se deben incluir en la memoria, bajo los epígrafes “Se pide...”. Cuide la presentación de estas memorias, de forma que refleje su trabajo y el esfuerzo dedicado a la asignatura. No copie los contenidos de sus compañeros. La extensión de cada memoria no debería superar las 10 páginas. Sea breve y conciso, pero justifique adecuadamente las decisiones de diseño que tome, incluyendo breves comentarios.

Las prácticas 2, 3 y 4 proponen diseños combinatoriales y secuenciales de propósito general que forman parte de una aplicación completa que se integrará en la práctica 5. La aplicación que se realizará este curso es un juego de dados para dos personas. La implementación de las prácticas 2 a 5 se hará mediante el uso de dispositivos programables, en lugar de usar componentes discretos. Para el diseño se usará la herramienta de ayuda al diseño electrónico (CAD/EDA) Quartus™ de ALTERA (www.altera.com) u otra herramienta alternativa de otro proveedor.

El objetivo es que el alumno se familiarice con el uso de este tipo de herramientas para la captura de esquemas electrónicos digitales y la simulación de los mismos, al tiempo que pone en práctica las técnicas básicas de diseño en electrónica digital. Finalmente, el circuito descrito se implementará físicamente en un dispositivo lógico programable del fabricante ALTERA (u otro proveedor alternativo) y se realizarán las pruebas pertinentes para comprobar que el funcionamiento de los circuitos desarrollados es el especificado. El proceso de diseño que debe seguir se muestra en la siguiente figura.

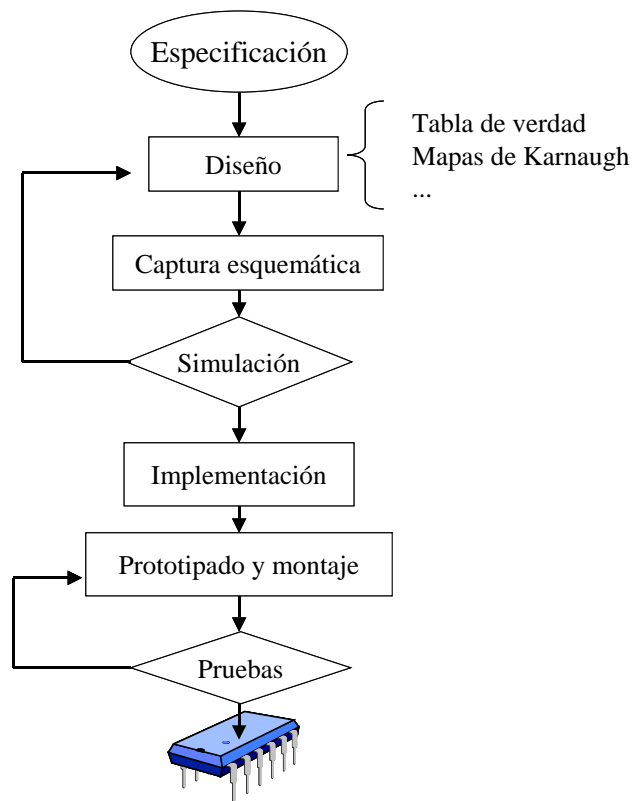


Figura 4.1. Ciclo de diseño de circuitos digitales mediante esquemáticos.

El material fungible que necesite para las prácticas NO SE SUMINISTRA EN EL LABORATORIO. Consulte la lista de material que hemos preparado a fin de orientarle en sus compras. Los componentes que comprará le serán, probablemente, de utilidad para otras prácticas debido a que son de propósito general. En cualquier caso, especialmente para la práctica 1, el material a comprar dependerá del diseño que haya adoptado. Téngalo en cuenta con suficiente antelación.

2. DESCRIPCIÓN DEL SISTEMA A DISEÑAR

El circuito que se va a diseñar en las sesiones prácticas de la asignatura Fundamentos de Electrónica Digital consiste en un juego electrónico de dados para dos personas. En el sistema se incluyen interfaces con componentes externos como pulsadores, displays de 7 segmentos y dispositivos LED. Así mismo, el circuito contiene elementos secuenciales de memoria y de control y elementos combinacionales aritméticos y de decodificación que permitirán realizar todas las tareas involucradas en el juego.

El sistema digital a diseñar es un juego de dados para dos participantes, cada uno de los cuales puede realizar dos tiradas con un dado electrónico. El jugador cuya puntuación acumulada sea mayor ganará la partida.

En la figura adjunta se muestra la arquitectura del circuito a diseñar. En los recuadros incluidos en la figura se indica la práctica en la que se realizará cada bloque.





3. PRÁCTICA 1. DISEÑO DE CIRCUITOS COMBINACIONALES (I). DISEÑO CON COMPONENTES DISCRETOS

En esta práctica se pretende que el alumno se familiarice con la lógica combinacional. Para ello se propone el diseño y la implementación mediante puertas lógicas de dos sistemas combinacionales:

- **Convertidor de código hexadecimal a pantallas de 7 segmentos**
- **Decodificador de mensajes para juego de dados**

La implementación del sistema debe ser óptima empleando el **menor número de componentes discretos posible**. Tenga en cuenta que en un mismo integrado hay varias puertas lógicas, pero siempre del mismo tipo.

La metodología de trabajo que adoptaremos en esta práctica será la siguiente:

- 1.- Realizar el diseño del bloque propuesto empleando mapas de Karnaugh para la simplificación de las funciones lógicas.
- 2.- Decidir la tecnología sobre la que implementaremos nuestro proyecto. Todas las puertas deben ser de la misma tecnología para evitar incompatibilidades. Así si las puertas poseen la nomenclatura 74LSXX corresponderán a la tecnología TTL y si poseen la nomenclatura 74HCXX corresponderán a la tecnología CMOS.
- 3.- Realizar el mapeado tecnológico y rutado de las pistas para esa tecnología, teniendo en cuenta que el diseño va a realizarse sobre los entrenadores del laboratorio. Este proceso debe hacerse manualmente, ya que no disponemos de herramientas que nos asistan en este paso.
- 4.- Realizar el montaje del circuito en el laboratorio, a partir de los planos de mapeado y rutado obtenidos en el punto anterior.
- 5.- Proceder a la verificación del funcionamiento del circuito montado, utilizando los entrenadores lógicos del laboratorio.

Los pasos 1 a 3 se incluirán en la memoria que debe entregarse al inicio de la sesión de prácticas, mientras que los pasos 4 y 5 se realizarán durante la sesión.

Para confeccionar el plano de mapeado tecnológico y rutado de componentes es necesario conocer la estructura de las placas de inserción del laboratorio.

Los entrenadores tienen dos o tres placas de inserción, consistentes en una matriz de agujeros donde se introducen los pines de los componentes. No se necesita ningún ajuste mecánico adicional, ya que los pines ajustan en los agujeros. Además, la placa provee una conexión eléctrica entre agujeros que están en la misma columna. En la figura 5.1 se indica el aspecto de un bloque.

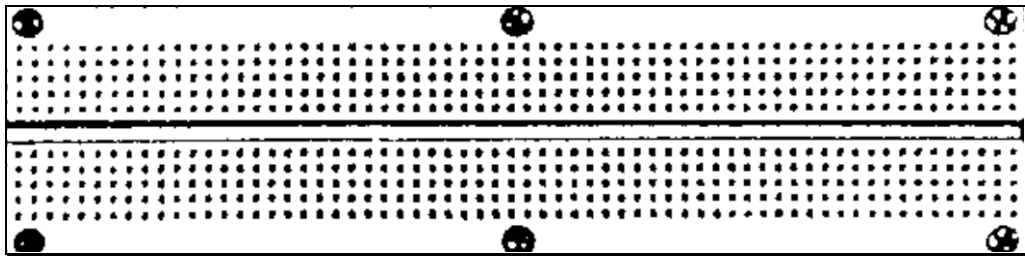


Figura 6.1. Placa de inserción

Las placas del entrenador tienen tres bloques como los de la figura. Se observa que las columnas de agujeros están divididas por la mitad, dejando cinco agujeros en cada parte. Esto significa que la conexión eléctrica se interrumpe ahí. Están conectados los cinco de arriba de cada columna entre sí y los cinco de abajo de cada columna entre sí, por lo que el rutado deberá realizarlo mediante cables.

No olvide incluir la alimentación de los circuitos integrados. El pin 1 es el que se encuentra a la izquierda de la muesca del chip, o bien viene indicada por un punto. El pin 14 (o el de mayor numeración) es el que se encuentra a la derecha de la muesca.

Ejemplo de mapeado y rutado:

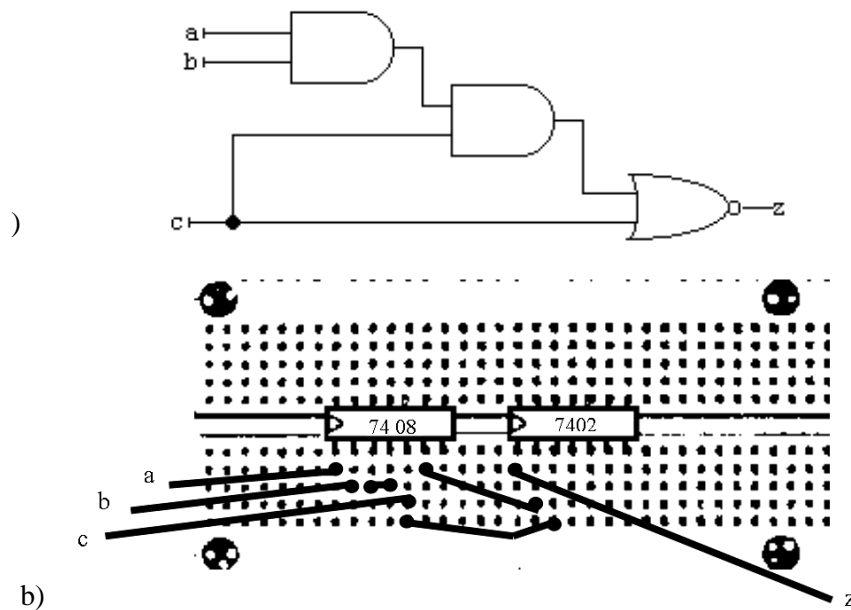


Figura 6.2. a) Esquema del circuito b) Circuito mapeado sobre tecnología TTL 74 XX y rutado sobre uno de los bloques de las placas del laboratorio (por simplicidad se ha omitido la alimentación de los chips):

3.1 Convertidor de código hexadecimal a pantalla de 7 segmentos.

Este bloque combinacional permite la visualización de un dato en código hexadecimal en un *display* de 7 segmentos.

NOTA: Los *displays* de 7 segmentos son dispositivos formados por 7 diodos LED configurados según la figura, de forma que si todos están iluminados muestran un '8' y si se iluminan de forma adecuada pueden mostrar todos los números del 0 al 9 y letras de la A a la F.

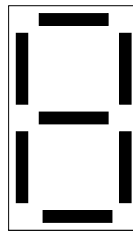


Figura 6.3. Display de 7 segmentos

Estos dispositivos constan de 7 diodos LED conectados bien por el cátodo, en cuyo caso se denominan por cátodo común, o bien por el ánodo denominándose por ánodo común. El dispositivo no incluye ninguna alimentación ni polarización, por lo que ésta se debe suministrar externamente. Dependiendo del terminal común y su conexión externa las patillas de cada segmento serán activas por '1's o por '0's.¹

A continuación se indican las entradas y salidas del circuito a diseñar en este apartado:



Figura 6.4. Decodificador BCD a 7 segmentos

Entradas:

- **A**: Dato hexadecimal (4 bits) (Valores posibles: 0,1 2, 3, 4, 5, 6, 7, 8, 9, A, b, C, d, E y F).
- **Enable**: 1 bit. Si vale '0' en el display no se muestra nada, en caso contrario el circuito realiza la decodificación del dato A a 7 segmentos.

Salidas:

- **A, b, c, d, e, f y g**: 1 bit cada una. Activas por '1's o por '0's según los displays de 7 segmentos adquiridos por el alumno.

¹ Ver hoja de catálogo adjunta

**SE PIDE:**

- Tabla de verdad
- Mapas de Karnaugh y minimización de funciones lógicas
- Funciones lógicas resultantes
- Esquema resultante con puertas lógicas de forma que el resultado sea el más óptimo en tamaño final y velocidad.

3.2 Decodificador de mensajes para juego de dados

Este bloque se encarga de convertir un dato hexadecimal (4 bits) a un código válido para el juego de dados. Así, los códigos admitidos en este juego son:

<i>Código hexadecimal</i>	<i>Significado</i>	<i>Mostrado en pantalla</i>
0	Juego sin comenzar	0
1	Tirada de dado → 1	1
2	Tirada de dado → 2	2
3	Tirada de dado → 3	3
4	Tirada de dado → 4	4
5	Tirada de dado → 5	5
6	Tirada de dado → 6	6
7	Código No Valido	9
8	Dado en movimiento	8
9	Código No Valido	9
A	Gana A	A
B	Gana B	B
C	Código No Valido	9
D	Código No Valido	9
E	Empate	E
F	Código No Valido	9

Por tanto, este bloque recibirá un código hexadecimal de 4 bits de entrada y generará otro código hexadecimal de 4 bits tal como indica la tabla anterior.

SE PIDE:

- Tabla de verdad
- Mapas de Karnaugh y minimización de funciones lógicas
- Funciones lógicas resultantes
- Esquema resultante con puertas lógicas de forma que el resultado sea el más óptimo en tamaño final y velocidad.



3.3 Prueba en el laboratorio: representación de resultados en el juego de los dados

Una vez diseñados y probados en el laboratorio los bloques anteriores, se procederá a su conexión para poder representar números entre 1 y 6 (resultados del dado), las letras A y B (jugadores), la letra E (empate), el número 0 (juego no comenzado), el número 9 (error) y el número 8 (representación de tirada de dado). El montaje se realizará con un *display* de 7 segmentos. Para proporcionar las entradas a este circuito, se indicarán los interruptores del entrenador de microelectrónica. Las salidas se mapearán en el *display* de 7 segmentos que el alumno habrá traído al laboratorio.

4. PRÁCTICA 2. DISEÑO DE CIRCUITOS COMBINACIONALES (II)

En esta práctica el alumno continuará adquiriendo destreza en el diseño y desarrollo de circuitos combinacionales sencillos. El prototipado se realizará en un dispositivo programable (FPGA) que permite la implementación de sistemas de mayor complejidad con menor esfuerzo de montaje.

4.1 Comparador

Este bloque combinacional realiza una comparación entre dos datos de entrada (A y B) e indica en sus salidas si A es igual que B, si A es mayor que B o si A es menor que B.

A continuación se indican las entradas y salidas del componente:



Figura 7.1. Comparador

Entradas:

- **A**: Dato de 4 bits codificado en binario natural.
- **B**: Dato de 4 bits codificado en binario natural.
- **Enable**: 1 bit. Si vale '0' todas las salidas son '0', en caso contrario el circuito compara el dato A con el dato B.

Salidas:

- **Igual**: 1 bit. Si A es igual que B vale '1', en otro caso vale '0'.
- **Mayor**: 1 bit. Si A es mayor que B vale '1', en otro caso vale '0'.
- **Menor**: 1 bit. Si A es menor que B vale '1', en otro caso vale '0'.

SE PIDE:

- Tabla de verdad
- Mapas de Karnaugh y minimización de funciones lógicas
- Funciones lógicas resultantes
- Esquema resultante con puertas XOR, AND, OR y NOT de forma que el resultado sea el más óptimo en tamaño final, consumo y velocidad.
- Simulación del circuito. Justificación breve y razonada de los valores asignados a las entradas para la simulación
- En el laboratorio, simulación y pruebas en la FPGA EPM7128SLC84-10

4.2 Sumador completo de cuatro bits.

El sistema a diseñar es un sumador de cuatro bits como el que se muestra en la figura adjunta y consta de las entradas y salidas detalladas a continuación.

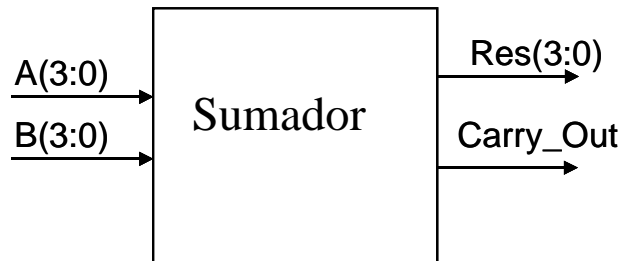


Figura 7.2. Sumador de cuatro bits

Entradas:

- **A, B**: Operandos codificados en binario natural, de cuatro bits cada uno.

Salidas:

- **S**: cuatro bits correspondientes al resultado de la suma representada en binario natural.
- **Carry Out**: 1 bit correspondiente al acarreo de la suma.

Para la implementación de este sistema podrán usarse los interruptores y LEDs disponibles en el entrenador de su puesto de laboratorio.

SE PIDE:

- a) Diseño de la arquitectura del circuito. Diagrama de bloques y diseño de cada bloque (tabla de verdad, mapa de Karnaugh y funciones lógicas resultantes)
- b) Captura esquemática del circuito en Quartus II mediante diseño jerárquico de bloques
- c) Simulación del circuito. Justificación breve y razonada de los valores asignados a las entradas para la simulación
- d) En el laboratorio, simulación y pruebas en la FPGA EPM7128SLC84-10

5. PRÁCTICA 3. DISEÑO DE CIRCUITOS SECUENCIALES (I)

En esta práctica se pretende que el alumno adquiera capacidad suficiente para el diseño de circuitos secuenciales síncronos con funcionalidad específica. La práctica consiste en el diseño de componentes sencillos ampliamente utilizados en sistemas digitales aritméticos:

- Registro de datos
- Acumulador (que permite almacenar la puntuación de cada jugador)

Siguiendo el ciclo de diseño expuesto en la introducción a las prácticas, el alumno deberá plantear el diseño de acuerdo a la especificación propuesta y obtener la solución más sencilla, barata y eficiente posible.

5.1 Registro de datos

Este componente es síncrono (reloj *clock*), con inicialización asíncrona (*reset*), tiene una señal que habilita la carga (*Load*) de un dato (*Data*), en cada flanco activo del reloj. Además, el bloque cuenta con una señal de inicialización síncrona (*Clear*) que permite el borrado del contenido del registro en el flanco activo del reloj.

A continuación se indican las entradas y salidas del componente:

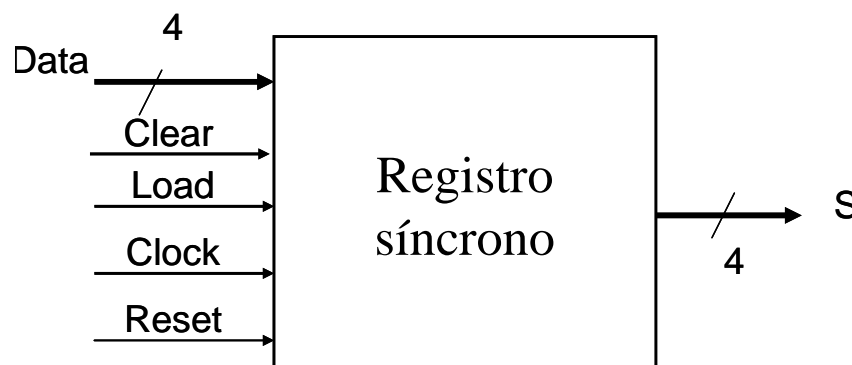


Figura 8.1. Registro síncrono

Entradas:

- **Data**: Dato binario (4 bits).
- **Clear**: 1 bit. Si vale '1' se borra el contenido almacenado.
- **Load**: 1 bit. Si vale '1' se carga el dato de entrada (*Data*) en el registro, en caso contrario se mantiene el dato anterior.
- **Clock**: 1 bit. Reloj de sincronismo que permite la actualización del contenido del registro en cada flanco de subida del mismo.
- **Reset**: 1 bit. Señal de inicialización asíncrona. Activa por nivel bajo, si vale '0' el bloque actualiza su valor inmediatamente y se carga a "000".

Salidas:

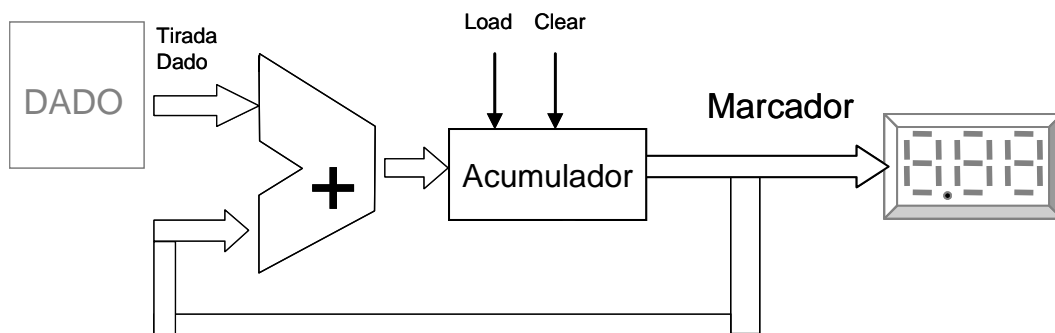
- **S**: 4 bits. Contenido del registro en todo momento.

SE PIDE:

- Diagrama de bloques. Diseño de los bloques que no estén diseñados en los apartados anteriores.
- Captura esquemática del circuito en Quartus II mediante diseño jerárquico de bloques
- Simulación del circuito. Justificación breve y razonada de los valores asignados a las entradas para la simulación

5.2 Marcador para cada jugador (Acumulador)

Este bloque acumula la puntuación obtenida por un jugador a lo largo de sus dos tiradas de dado. El acumulador es un registro de datos, diseñado en el apartado anterior. Además, el bloque contiene el sumador diseñado en la práctica 2. En la figura se detallan los componentes mencionados.

**Figura 8.2. Registro síncrono****Entradas:**

- **Tirada Dado:** Dato binario (3 bits).
- **Load:** 1 bit. Si vale '0' se mantiene almacenado el valor anterior.
- **Clear:** 1 bit. Si vale '1' se borra el contenido almacenado.
- **Clock:** 1 bit. Reloj de sincronismo que permite la actualización del contenido del registro en cada flanco de subida del mismo.
- **Reset:** 1 bit. Señal de inicialización asíncrona. Activa por nivel bajo, si vale '0' el bloque actualiza su valor inmediatamente y se carga a "000".

Salidas:

- **Marcador:** 4 bits. Contenido del registro en todo momento que indica la puntuación obtenida por el jugador



SE PIDE:

- a) Diagrama de bloques. Diseño de los bloques que no estén diseñados en los apartados anteriores.
- b) Captura esquemática del circuito en Quartus II mediante diseño jerárquico de bloques
- c) Simulación del circuito. Justificación breve y razonada de los valores asignados a las entradas para la simulación
- d) En el laboratorio, simulación y pruebas en la FPGA EPM7128SLC84-10

6. PRÁCTICA 4. DISEÑO DE CIRCUITOS SECUENCIALES (II)

El objetivo de esta práctica es que el alumno diseñe dos máquinas de estados, aplicación clásica dentro de la electrónica digital. Las máquinas de estados son ampliamente usadas en la industria en multitud de sistemas de control. En este caso las máquinas a implementar son:

- Contador síncrono ascendente. Generador aleatorio de números entre 1 y 6.
- Detector de flancos (conformador de pulsos)

Siguiendo el ciclo de diseño expuesto en la introducción a las prácticas, el alumno deberá plantear el diseño de acuerdo a la especificación propuesta y obtener la solución más sencilla, barata y eficiente posible.

6.1 Contador síncrono. Generador aleatorio de números entre 1 y 6

Este componente es síncrono (reloj *clock*), con inicialización asíncrona (*reset*) y tiene una señal que habilita la cuenta (*Enable*). El contador actualiza su valor en cada flanco activo del reloj. Este circuito cuenta entre los valores 1 y 6 de forma continua, siempre que la señal *Enable* lo permita. Este contador permitirá emular el comportamiento aleatorio de un dado, si la frecuencia de funcionamiento es suficientemente rápida (> 1 MHz).

A continuación se indican las entradas y salidas del componente:

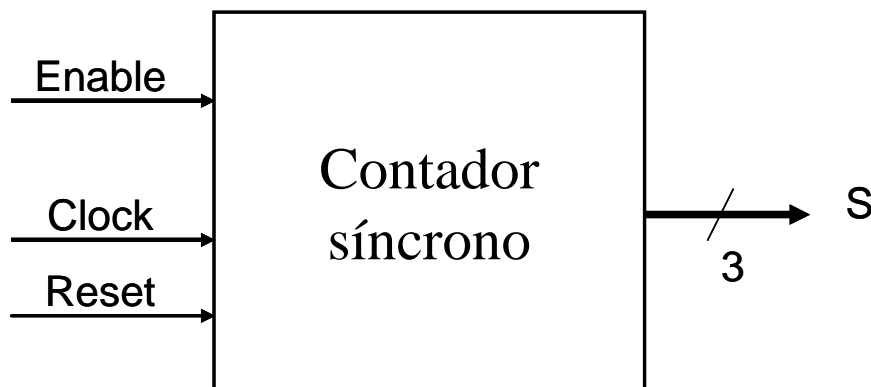


Figura 9.1. Contador síncrono

Entradas:

- **Enable:** 1 bit. Habilita la cuenta, si vale '0' el contador mantiene su valor, si vale '1' el contador incrementa/decrementa su valor.
- **Clock:** 1 bit. Reloj de sincronismo que permite la actualización del contenido del registro en cada flanco de subida del mismo.
- **Reset:** 1 bit. Señal de inicialización asíncrona. Activa por nivel bajo, si vale '0' el bloque actualiza su valor inmediatamente y se carga a "000".

Salidas:

- S : 3 bits. Contenido del contador en todo momento (valores entre 1 y 6)

El circuito se diseñará como una máquina de estados de Moore.

SE PIDE:

- Diagrama de estados indicando los valores de las salidas en cada estado, y los valores de las entradas en las transiciones y en los estados.
- Número de biestables necesario y asignación de estados. Utilice biestables tipo D.
- Tabla de verdad
- Mapas de Karnaugh y minimización de funciones lógicas (utilizando biestables D)
- Funciones lógicas resultantes
- Esquema resultante con puertas lógicas y biestables de forma que el resultado sea el más óptimo en tamaño final, consumo y velocidad.
- Simulación del circuito. Justificación breve y razonada de los valores asignados a las entradas para la simulación
- En el laboratorio, simulación y pruebas en la FPGA EPM7128SLC84-10

6.2 Detector de flancos

Este componente se encarga de preprocesar la información procedente de la interfaz con el usuario. Esta interfaz está compuesta en este caso por botones, cuando el usuario pulse un botón se activa la entrada correspondiente. La dificultad radica en la duración y la calidad de los pulsos recibidos a la entrada (rebotes, *glitches*, etc.). Dependiendo de la frecuencia de reloj utilizada en el circuito, una única pulsación del botón puede provocar varias operaciones.

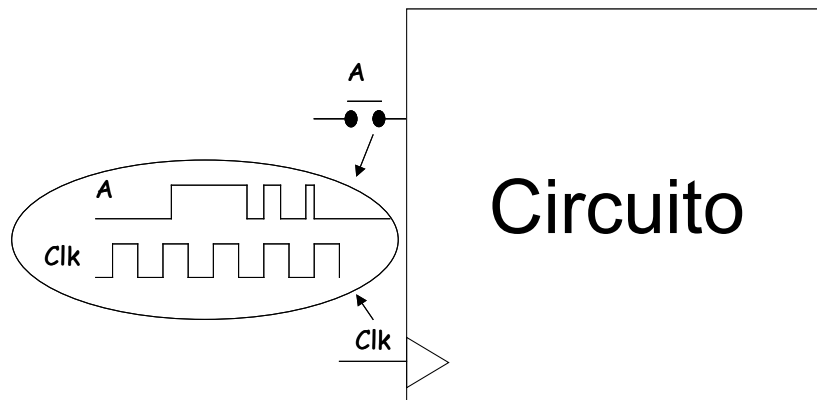


Figura 9.2. Entradas al bloque detector de flancos

Para resolver esta situación se utilizan botones antirrebotes (ya disponibles en el entrenador del laboratorio) y un circuito detector de flancos que permite ajustar la duración del pulso a la frecuencia del reloj interno. Este circuito es el que se va a diseñar en este apartado como un circuito síncrono y concretamente como una máquina de estados.

El componente pedido es síncrono (reloj *clock*), con inicialización asíncrona (*reset*), tiene una entrada (*Botón*) y una salida (*Pulso*). Cuando *Botón* se activa (se pone a '1') el circuito debe activar la salida *Pulso*, durante un único ciclo de reloj (*Clock*), posteriormente *Pulso* se pone a cero y no se vuelve a activar hasta que *Botón* no haya bajado y vuelto a subir. Si el *Botón*

se activa durante un tiempo más corto que el período del reloj del circuito, *glitch*, la salida se mantiene inactiva ('0').

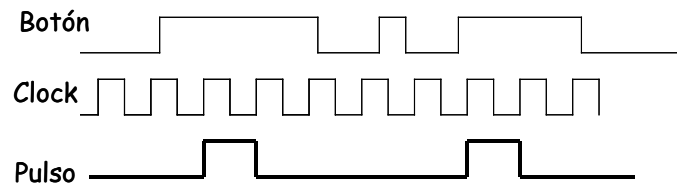


Figura 9.3. Cronograma de las señales del detector de flancos

A continuación se indican las entradas y salidas del componente:

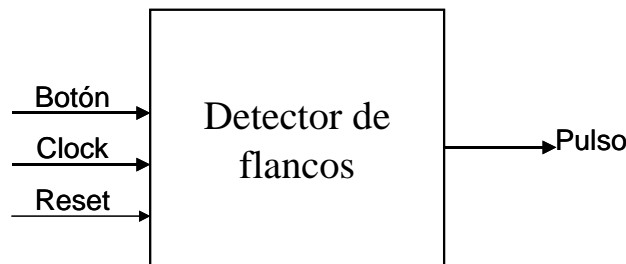


Figura 9.4. Detector de flancos

Entradas:

- **Botón:** 1 bit. Activo por nivel '1'.
- **Clock:** 1 bit. Reloj de sincronismo que permite la actualización del contenido del registro en cada flanco de subida del mismo.
- **Reset:** 1 bit. Señal de inicialización asíncrona. Activa por nivel bajo, si vale '0' el bloque actualiza su valor inmediatamente y se carga a "00".

Salidas:

- **Pulso:** 1 bit. Activa por nivel '1', sólo dura 1 ciclo de reloj *Clock*.

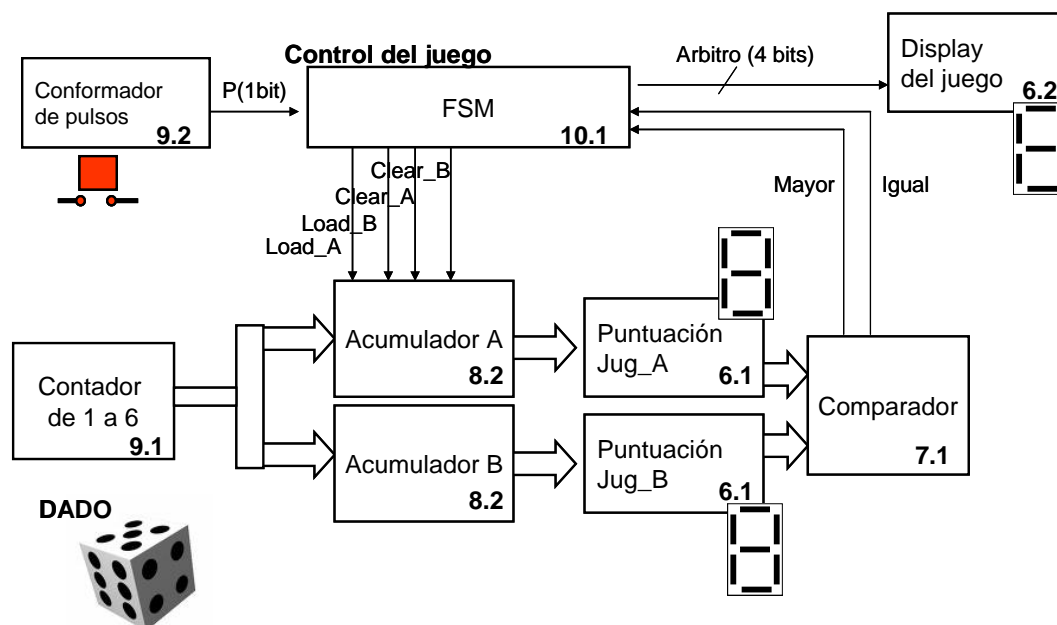
SE PIDE:

- a) Diagrama de estados indicando los valores de las salidas en cada estado, y los valores de las entradas en las transiciones y en los estados.
- b) Número de biestables necesario y asignación de estados. Utilice biestables tipo D.
- c) Tabla de verdad
- d) Mapas de Karnaugh y minimización de funciones lógicas (utilizando biestables D)
- e) Funciones lógicas resultantes
- f) Esquema resultante con puertas lógicas y biestables de forma que el resultado sea el más óptimo en tamaño final, consumo y velocidad.
- g) Simulación del circuito. Justificación breve y razonada de los valores asignados a las entradas para la simulación
- h) En el laboratorio, simulación y pruebas en la FPGA EPM7128SLC84-10

7. PRÁCTICA 5. IMPLEMENTACIÓN DEL SISTEMA COMPLETO

El objetivo de esta práctica es que el alumno implemente un sistema digital de complejidad media, utilizando componentes anteriormente diseñados. Como ya se dijo, el sistema completo es un juego de dados para dos participantes, cada uno de los cuales puede realizar dos tiradas con un dado electrónico. El jugador cuya puntuación acumulada sea mayor ganará la partida.

El esquema del sistema total se muestra en la figura adjunta. En dicha figura se indican los bloques que forman el sistema y en qué práctica se han desarrollado (en qué apartado de este manual figuran). En esta práctica, además del circuito completo, se realizará la máquina de estados que controla el juego (10.1).



7.1 Control del juego (Máquina de estados)

Este bloque controla el juego de dados, dando el turno a cada uno de los jugadores y decidiendo el ganador. En concreto, este bloque realiza las siguientes tareas

1. Genera el código a mostrar en el display del juego (7 segmentos)
2. Habilita el borrado de los acumuladores de A y B antes de empezar una nueva partida
3. Permite la carga de cada tirada de los jugadores, actualizando sus marcadores
4. Decide cuándo termina el juego y asigna un ganador o indica un empate

Esta máquina de estados tiene las siguientes entradas y salidas:

Entradas:

- **P:** 1 bit. Activo por nivel '1' (1 ciclo de reloj). Activa el juego o indica tirada de dado por un jugador.



- **Mayor:** 1 bit. Activa por nivel '1'. Indica que la puntuación del jugador A es mayor que la de B
- **Igual:** 1 bit. Activa por nivel '1'. Indica que la puntuación del jugador A es igual que la de B
- **Clock:** 1 bit. Reloj de sincronismo que permite la actualización del contenido del registro en cada flanco de subida del mismo.
- **Reset:** 1 bit. Señal de inicialización asíncrona. Activa por nivel bajo, si vale '0' el bloque actualiza su valor inmediatamente y se carga a "00".

Salidas:

- **Load A:** 1 bit. Activa por nivel '1', sólo dura 1 ciclo de reloj *Clock*. Permite la carga del acumulador para el jugador A
- **Clear A:** 1 bit. Activa por nivel '1', sólo dura 1 ciclo de reloj *Clock*. Permite el borrado síncrono del acumulador del jugador A
- **Load B:** 1 bit. Activa por nivel '1', sólo dura 1 ciclo de reloj *Clock*. Permite la carga del acumulador para el jugador B
- **Clear B:** 1 bit. Activa por nivel '1', sólo dura 1 ciclo de reloj *Clock*. Permite el borrado síncrono del acumulador del jugador B
- **Arbitro:** 4 bits. Código de situación del juego:

<i>Código</i>	<i>Significado</i>
0	Reposo. El juego no ha comenzado
8	Dado en movimiento
9	Error
A	Gana A
B	Gana B
E	Empate



SE PIDE:

1. Diagrama de estados indicando los valores de las salidas en cada estado, y los valores de las entradas en las transiciones y en los estados.
2. Número de biestables necesario y asignación de estados. Utilice biestables tipo D.
3. Tabla de verdad
4. Mapas de Karnaugh y minimización de funciones lógicas (utilizando biestables D)
5. Funciones lógicas resultantes
6. Esquema resultante con puertas lógicas y biestables de forma que el resultado sea el más óptimo en tamaño final, consumo y velocidad.
7. Simulación del circuito. Justificación breve y razonada de los valores asignados a las entradas para la simulación
8. En el laboratorio, simulación y pruebas en la FPGA EPM7128SLC84-10

7.2 Sistema completo

Este es el último paso para el montaje del sistema completo. Si todos los bloques han sido simulados y montados en la FPGA, este paso no deberá suponer ningún problema a los alumnos. Se recomienda asegurarse del funcionamiento del conjunto mediante simulación antes de realizar el prototipado en la FPGA.

SE PIDE:

1. Simulación del circuito. Justificación breve y razonada de los valores asignados a las entradas para la simulación
2. En el laboratorio, simulación y pruebas en la FPGA EPM7128SLC84-10



LISTA DE MATERIAL

En el laboratorio dispondrá de un puesto equipado con un entrenador digital, un ordenador, un polímetro y varios equipos electrónicos. Para las prácticas 2 a 5 dispondrá de una placa de circuito impreso que contiene un dispositivo programable sobre el que grabará las implementaciones de sus diseños.

El resto de material debe aportarlo el alumno. A continuación se ofrece una lista orientativa:

- Alicates de corte o pelacables
- Cablecillo de conexión (se recomienda tener varios colores)
- 1 Display de 7 segmentos
- 7 Resistencias de 220Ω (1/4 W)
- Circuitos integrados 74XX a determinar en el diseño de la práctica 1