

---

# Práctica 4: Proyecto final

---

## *Máquinas de Estados (FSM). Montaje con EEPROMs*

### INTRODUCCIÓN

En esta última práctica se pretende desarrollar un proyecto que incluye dos ejemplos de sistemas de control digital. Por una parte, se realizará el diseño y simulación de un sistema de control sencillo por medio de una máquina de estados finitos (FSM) y de manera complementaria, se resolverá el mismo problema usando una memoria EEPROM para sustituir a la lógica combinacional, tanto para las ecuaciones de excitación como para las ecuaciones de salida.

Los únicos elementos secuenciales que se podrán utilizar serán FF tipo D. El elemento de la librería de Xilinx para este elemento se denomina "fd". Si fuera necesario un FF con entrada de clear, puede utilizar el "fdc".

Para implementar los FF's tipo D en el montaje, se podrá utilizar el componente 74HC175 o el contador 74HC163, el cual, al realizar una carga ("load"), la salida del contador en su primer ciclo de reloj se comporta de forma análoga a 4 FF's tipo D.

En el ejercicio de montaje se demostrará, de una manera aplicada, la utilidad de las memorias ROMs para construir funciones lógicas complejas, en particular un sistema de control.

La duración prevista de esta sesión práctica es de 4 semanas en total, incluyendo diseño y montaje. Los estudiantes se repartirán el tiempo teniendo en cuenta que en la última semana de curso se realizará un control.

## PARTE 1: DISEÑO CON XILINX ISE

### Ejercicio 1 (3 puntos)

Hemos sido contratados por la empresa internacional M'estresso que nos ofrece la oportunidad de diseñar por medio de una FSM de Moore el funcionamiento de una máquina de café de capsulas de última generación. Nuestra máquina deberá tener 4 fases diferentes de funcionamiento: INICIO, CALENTAR, READY y CAFE.

Pulsando un interruptor RESET, la máquina se conecta y entra automáticamente en la fase de calentamiento, activando la resistencia que calienta la cafetera (CALENTAR = '1'). El agua necesita 3 ciclos enteros de reloj en calentarse, una vez pasados estos 3 ciclos, el agua se mantiene caliente sin necesidad de tener activa la resistencia y en el siguiente ciclo, la máquina ya se encuentra preparada para funcionar (READY=1) y permanece en ese estado hasta que pulsamos el interruptor "MARCHA".

Al pulsar el interruptor "MARCHA", la máquina activa la bomba de agua caliente para sacar café (CAFE=1), siempre y cuando se haya puesto una cápsula en su interior. (Nota: si no hay capsula, la maquina saca agua caliente, perfecto para infusiones).

La bomba funciona de manera continua hasta que apagamos el interruptor "marcha". En ese momento, la máquina vuelve al estado "READY".

La FSM a diseñar tendrá, por tanto:

- 3 entradas (RESET, CLK y MARCHA). Considere RESET como una entrada asíncrona de inicialización, por lo que ni esta señal ni el reloj CLK, deben ser consideradas como variables en el diseño de la FSM.

- 3 salidas en forma de bit de estado: CALENTAR, READY, CAFE

Diseñar con la herramienta Xilinx ISE y simular con ModelSim la FSM de tipo Moore. Los únicos elementos secuenciales que se podrán utilizar serán FF tipo D. El elemento de la librería de Xilinx para este elemento se denomina "fd". Si fuera necesario un FF con entrada de clear, puede utilizar el "fdc".

Para los ejercicios posteriores, se recomienda crear un símbolo con el mismo nombre que el ejercicio hecho "p4ej1.sym".

Para probar el correcto funcionamiento del diseño se proporciona un fichero de testbench "p4ej1\_tb.vhd".

Para que el banco de pruebas funcione correctamente se debe cumplir lo siguiente:

Nombre del componente: p4ej1

Nombre de las entradas: CLK, RESET, MARCHA

Nombre de las salidas: CALENTAR, READY, CAFE

Fichero a entregar: p4ej1.sch

## Ejercicio 2 (3 puntos)

El departamento de marketing ha recibido numerosas quejas de clientes por la incomodidad que supone estar pendiente del llenado. Por tanto, se nos solicita que la máquina tenga prestaciones similares a las ya existentes en el mercado, debemos diseñar un modelo automático y así evitar estar pendiente de la maquina mientras prepara el café.

El reto consiste en, utilizando las mismas entradas, realizar el diseño de un sistema automático. El interruptor de MARCHA se convierte ahora en un pulsador (genera un solo pulso) con el mismo nombre. Al pulsar "marcha" desde el estado ready, la cafetera deberá proporcionar café durante 8 ciclos. Para el diseño se recomienda el uso del contador "cb4cled" y los Flip Flops tipo D "fdc" disponibles en Xilinx.

Para probar el correcto funcionamiento de este diseño se proporciona un fichero de testbench "p4ej2\_tb.vhd". Para que el banco de pruebas funcione correctamente se debe cumplir lo siguiente:

Nombre del componente: p4ej2

Nombre de las entradas: RESET, CLK y MARCHA

Nombre de las salidas: CALENTAR, READY, CAFE

Fichero a entregar: p4ej2.sch

## Ejercicio 3 (4 puntos):

Por motivos de industrialización, es necesario simplificar las tarjetas electrónicas para reducir el coste de fabricación. Para lograrlo, se nos solicita diseñar y simular el problema del ejercicio 2 utilizando FF's tipo D y un bloque de memoria de n bits de ancho, donde n es el número de funciones combinacionales diferentes que se necesitan para las salidas (Calentar, Ready, Café)) y para excitar a los biestables (Di). Los bits de estado (Qi) y la entrada del sistema (Marcha), sirven para direccionar la memoria.

Para el diseño de este esquema, se sugiere utilizar  $n$  memorias ROM de 32 palabras de 1 bit (ROM32x1), donde  $n$  indica el número de las funciones combinacionales que el diseño precisa.

Para grabar el contenido en una memoria de este tipo, pulsar sobre el dispositivo con el botón derecho del ratón y seleccionar las propiedades de objeto. En la fila llamada INIT, escribir en hexadecimal los 16 o 32 bits de la función combinacional deseada, en donde el bit más a la izquierda corresponde a la dirección más alta de la memoria ("11111") y el mas a la derecha a la dirección más baja ("00000"). Así por ejemplo si INIT = F000, las 12 primeras direcciones de la memoria valen 0 y las cuatro últimas 1.

Como una opción de diseño, se sugiere combinar las  $n$  memorias ROM de 1 bit en un símbolo para generar una única memoria de 16 o 32 palabras de  $n$  bits de ancho.

Para el diseño del sistema, se utilizará una señal de entrada de reloj CLK para gobernar una máquina de estados.

Para probar el correcto funcionamiento de este diseño se proporciona un fichero de testbench "p4ej3\_tb.vhd". Para que el banco de pruebas funcione correctamente se debe cumplir lo siguiente:

Nombre del componente: p4ej3

Nombre de las entradas: CLK, RESET, MARCHA

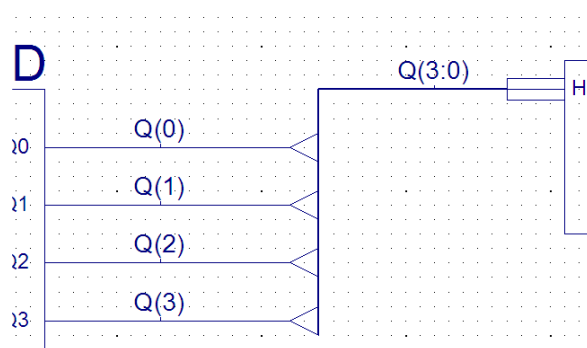
Nombre de las salidas: CALENTAR, READY, CAFE

Fichero a entregar: p4ej3.sch

### Nota sobre creación de buses

La figura ilustra como dibujar un bus de señales. Los pasos son los siguientes:

- Crear una línea de señal con Add Wire
- Anadir nombre XXX(n:0)
- Anadir bus taps con Add bus taps en cada punto en donde se quiera conectar una señal individual del bus
- Conectar las señales individuales al bus tap y llamarla XXX(i)



## Entrega

El día anterior a la tercera sesión presencial de la práctica (14 y 15 de diciembre), cada pareja entregará una memoria resumen que contendrá el siguiente material:

1. Descripción concisa del diseño para cada los tres ejercicios propuestos.
2. Descripción de los pasos intermedios: tablas de verdad, mapas de Karnaugh, simplificación de expresiones lógicas, etc.
3. Esquemáticos de todos los circuitos (capturas de pantalla con los diseños realizados con ISE).
4. Resultados de las simulaciones (normalmente, capturas de pantalla de ModelSim, con una breve explicación del resultado, que sean legibles y donde se puedan distinguir perfectamente las entradas y salidas).
5. Esquema (plano de montaje) lo más detallado posible del montaje hardware a realizar en la Parte 2 de la práctica (Ejercicio 3).

Se deberán subir a Moodle 3 ficheros por uno de los miembros de la pareja:

- Fichero comprimido con todos los esquemáticos (Max 1MB).
- Memoria PDF con la descripción del diseño, tablas y simplificaciones, resultados de simulación y su explicación (Max 1MB).
- Fichero PDF con el plano de montaje (Max 1MB).

La memoria será subida a Moodle por uno de los miembros de la pareja, como máximo una semana después de su diseño en el laboratorio, que en este caso supone **antes del martes 13 de diciembre a las 23:55h para los grupos 1111 y 110x, y antes del miércoles 14 de diciembre a las 23:55h para los grupos 112x y 116x.**

## Parte 2: Montaje

**ATENCIÓN:** Con independencia de la fecha límite para la entrega de la memoria, para poder efectuar el montaje es indispensable:

1. Presentar el diseño del circuito que se va a montar (Ejercicio 3), especificando los circuitos integrados que se van a usar y las conexiones en cada uno de los pines (ver ejemplo en el guion de la práctica 0a). No se permitirá realizar la práctica de montaje a ningún estudiante que no haya entregado este diseño del circuito (plano de montaje).
2. Presentar un archivo de texto en el formato HEX de Intel con el contenido de la memoria EEPROM a implementar. La descripción de este formato se muestra como Anexo B de esta memoria.

### HOJAS DE DATOS DE LOS CIRCUITOS DISPONIBLES

- Memoria EEPROM tipo SST39SF040 de SST
- Contador: 74HC163
- FF-D: 74HC175
- Lógica HCMOS de prácticas anteriores: 74HC08, 74HC21, 74HC08, 74HC32.

### Ejercicio único. Montaje del circuito del ejercicio 3 diseñado en Xilinx utilizando una memoria EEPROM. (10 puntos)

El montaje debe realizarse utilizando el FF-D 74HC175 o el contador con carga habilitada y la memoria EEPROM. Cada posición válida de la EEPROM debe contener tanto el valor de las entradas de carga del contador, equivalente al estado siguiente, como las salidas del estado actual. Como reloj se utilizará el reloj de 1 Hz provisto por el entrenador o alternativamente, se podrá generar manualmente con un interruptor del entrenador.

Las señales de entrada se simularán con interruptores del entrenador. Para representar las salidas del circuito se usarán los LEDs del entrenador.

Todas las señales de salida deberán estar conectadas a un LED. Adicionalmente, se recomienda conectar las entradas CLK y CLEAR o salidas de los FF's (Qs) como ayuda en la depuración.

El sistema debe incluir una entrada de reset (CLR) activa a nivel alto, que se utilizará para inicializar la máquina de estados.

**EXAMEN.** Durante los 30 primeros minutos de la quinta sesión de la práctica (quinta semana) se realizará un examen acerca de los diseños realizados. El objetivo de este examen es verificar que los estudiantes entienden la práctica y el diseño que han entregado.

**EVALUACIÓN.** La nota de esta práctica se calcula de la siguiente manera:

$$\text{Nota Práctica} = 0,25 * \text{Diseño} + 0,25 * \text{Memoria} + 0,25 * \text{Montaje} + 0,25 * \text{Examen}$$

La evaluación del diseño se basará no sólo en lo hecho en clase sino también en la calidad de la memoria entregada. Esta evaluación tendrá en cuenta aspectos como la simulación realizada con Xilinx, la claridad del diseño o la adecuada simplificación de las funciones lógicas. La evaluación del montaje valorará preferentemente que el circuito funcione.

## ANEXOS PARA LA REALIZACIÓN DEL MONTAJE PROPUESTO

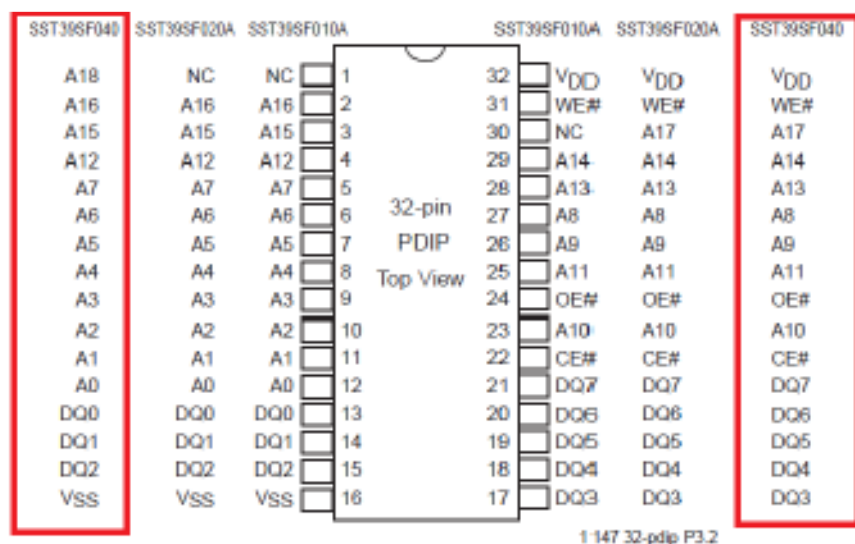
### A. USO DE LA MEMORIA EEPROM

Para realizar esta práctica se utilizará una memoria EEPROM tipo SST39SF040 de SST. Es una memoria ROM reprogramable, que se puede borrar eléctricamente. La memoria es de 1Mbit (128k x 8bits), organizada en 131.072 posiciones de 8 bits (131.072 x 8). Por lo tanto, tiene 17 líneas en el bus de direcciones (A16:A0) y 8 en el de datos (I/O7:I/O0).

Aparte de las líneas de dirección y datos, hay tres entradas de control, CE (Chip Enable), WE (WriteEnable) y OE (Output Enable), activas a nivel bajo. Para que la salida de la memoria quede habilitada, OE y CE tienen que estar a cero. Además, una vez grabada la EEPROM, la primera conexión a realizar en el montaje debe ser poner WE a nivel alto para evitar sobrescribir los datos almacenados. Ponga también a cero todas las líneas que no utilice salvo las NC, que no son necesario conectar. Lea nuevamente todo el párrafo: es vital realizar las conexiones mencionadas.

Para la preparación de la práctica es imprescindible conocer el patillaje (*pinout*) de la memoria, que está disponible en la hoja de datos que se facilita. El encapsulado del chip que se dispone en los laboratorios es DIP; no confundir con el TSOP cuyo patillaje también aparece en la hoja de datos facilitada.

Se reproduce a continuación el patillaje de la memoria SST39SF040, a efectos prácticos. El pin VSS debe conectarse a cero (GND) y el pin VDD a uno (Vcc) para la correcta alimentación de la memoria.



### B. ESCRITURA DE LOS DATOS EN LA EEPROM (formato HEX)

El primer paso en la grabación de una memoria tipo EEPROM es escribir los datos que se desean introducir en el fichero con un formato compatible con los grabadores de EEPROMs disponibles en el laboratorio. El formato que usaremos en esta práctica es el Intel HEX. Un archivo HEX típico tiene el siguiente aspecto en donde todos los dígitos numéricos están en hexadecimal:

```
:10008000AF5F67F0602703E0322CFA92007780C3FD
:1000900089001C6B7EA7CA9200FE10D2AA00477D81
:0B00A00080FA92006F3600C3A00076CB
:00000001FF
```

Los datos se dividen en campos (*records*), cada uno en una línea. Tomando como ejemplo la primera línea (:10 0080 00 AF5F67F0602703E0322CFA92007780C3 FD), el formato es:

- El primer carácter (:) indica el comienzo de un campo.
- Los dos caracteres siguientes ( $10_{16}$ ) indican la longitud en bytes (número de datos) del campo como un número hexadecimal de dos cifras. Por ejemplo, en este caso ( $10_{16}$ ) indica que el campo contiene  $16_{10}$  bytes de datos.
- Los cuatro caracteres siguientes ( $0080_{16}$ ) indica la posición de memoria donde se cargará el primero de los bytes de datos ( $128_{10}$ ).
- Los dos caracteres siguientes (00) indican el tipo de campo, 00 si se trata de datos, 01 para fin de fichero y 02 para dirección extendida (por ahora este último caso no lo usaremos).
- A continuación vienen los datos (AF.....C3), cada uno de ellos representado por dos caracteres como un número hexadecimal de un byte.
- Los dos últimos caracteres (FD) son el *checksum*. Este *checksum* se calcula de manera que la suma de todos los bytes del campo más el propio *checksum* sea 0x00 (00 en hexadecimal)

Con esta notación que utiliza 2 bytes para la dirección (16 bits), sólo se puede direccionar  $2_{16}$  bytes, es decir no podemos escribir (utilizar) toda la EEPROM. El bit de más peso (msb) de la dirección siempre vale 0 ( $A_{16}='0'$ ). No obstante, las posiciones sobre las que se puede escribir son suficientes para la realización de todas las prácticas de este curso.

Ya en el contexto de la práctica, lo primero será calcular los contenidos de la memoria, teniendo en cuenta el montaje que se vaya a realizar. Una vez calculados los datos, se pasarán a hexadecimal y se escribirá el archivo HEX en un editor de texto cualquiera (como el notepad de Windows)