

MICROELECTRÓNICA, 4º CURSO, P94

Diseño del Trazado de un Microcontrolador

Pablo Ituero y Marisa López Vallejo

Marzo 2013



Departamento de
**Ingeniería
Electrónica**

Universidad Politécnica de Madrid

Contenido

1	Introducción	3
2	Objetivo	4
3	Especificaciones	5
3.1	La ruta de datos	5
3.2	La máquina de estados finitos	7
3.3	La memoria ROM	7
4	Plan de trabajo	8
4.1	Revisión del diseño. Anteproyecto	8
4.2	Memoria final	8
5	Mejoras	11
5.1	Control en Pipeline	11
5.2	Instrucciones de Control	11
5.4	Operaciones adicionales multiciclo	12
5.5	Sustitución de la ROM de programa por una memoria RAM	12
5.6	Implementación real de estrategias de test	12
5.7	Estrategias para tolerancia a errores	12
5.8	Estrategias para bajo consumo	12
5.9	Estilos de lógica alternativos	12
5.10	Circuitería de entrada/salida customizada	12
6	Entrega del Proyecto	13
7	Críticas	14

1 Introducción

La gran variedad de materias que se cubren en este curso tiene como objetivo proporcionarle todo lo que necesita para diseñar y llevar a la práctica un circuito integrado CMOS. La razón del proyecto final es, justamente, la de ayudarlo a profundizar en estas materias y darle un mejor conocimiento de cómo es el proceso de diseño.

Su trabajo consistirá en diseñar un sencillo sistema CMOS hasta el punto de obtener un plano de base detallado. Partiendo de las especificaciones que se le entregan, y trabajando **en grupos de dos**, debe llevar a cabo el diseño y entregar una memoria de gran calidad. Procure finalizar el diseño, puesto que un diseño a medias será calificado pero y es menos instructivo que uno completo, a la vez que resultará mucho más difícil de escribir la memoria final.

Se proponen unas especificaciones mínimas que deberán cumplir todos los proyectos entregados y una serie de mejoras opcionales.

En concreto, para las especificaciones mínimas, el proyecto plantea la implementación de un microcontrolador, es decir, de una ruta de datos controlada con una máquina de estados finitos. El diseño, aunque sencillo, va a cubrir la mayor parte del temario de la asignatura, permitiéndole poner en práctica los conocimientos que ha adquirido, enfrentándole al reto de la gestión de la complejidad desde el nivel de transistor hasta el nivel de sistema.

Las mejoras cubren aspectos más avanzados del diseño e implican un pequeño avance del alumno respecto a los contenidos cubiertos por la asignatura. Al final de este documento hay una lista con mejoras propuestas, aunque esta lista puede abrirse a propuestas de los alumnos, siempre que cuenten con el visto bueno de los profesores de la asignatura.

La calificación del proyecto estará basada fundamentalmente en la calidad de su diseño - su simplicidad, corrección y grado de finalización-. Conforme a esto, debe buscar como meta principal un diseño correcto, con un trazado (layout) regular y bien estructurado. La caracterización del diseño y cada uno de los bloques que lo constituye respecto de tamaño, consumo y velocidad también es un factor que se tendrá muy en cuenta.

El resto de este documento establece los objetivos y los requisitos del proyecto, continúa con las especificaciones de la arquitectura del sistema. Posteriormente, se expone el plan de trabajo que se ha de seguir. Y, finalmente, se describen las mejoras propuestas.

Lea atentamente las instrucciones detalladas que se dan a continuación.

2 Objetivo

El objetivo del proyecto es diseñar un circuito integrado CMOS que cumpla las especificaciones descritas en este documento. Para ello debe implementar su arquitectura, adaptándola a nuestro estilo de diseño:

- Utilice CMOS de cualquiera de los tipos estudiados (dinámico, estático, dominó, etc.).
- Emplee dos fases de reloj, suponga, si lo desea, que $\Phi 1$ y $\Phi 2$ le vienen dados desde fuera de la pastilla. Por contra, también puede partir de una única fase externa y generar internamente las dos fases. Además, siga la disciplina de temporización (con extensiones) que se estudia en clase.

Puede realizar las suposiciones que considere necesarias, señalándolas y justificándolas convenientemente, se pueden obviar tareas repetitivas tales como la interconexión de un elevado número de cables, etc. Sus metas son un diseño regular, correcto, que pueda ser comprobado (test) y con mínimo cableado. Es decir, un diseño que pueda de hecho realizar y depurar sobre un ordenador.

El proyecto se realizará en parejas utilizando la herramienta de diseño full-custom ELECTRIC™, presentada en clase. Esta es una herramienta especialmente diseñada para el entorno académico, por lo que el trabajo que se haga con ella se verá facilitado en gran medida.

Para consultar cualquier duda que surja respecto al proyecto debe ponerse en contacto con el profesor Pablo Ituero Herrero (correo electrónico: pituero@die.upm.es, despacho C-226). Solicite siempre cita previa vía correo electrónico.

3 Especificaciones

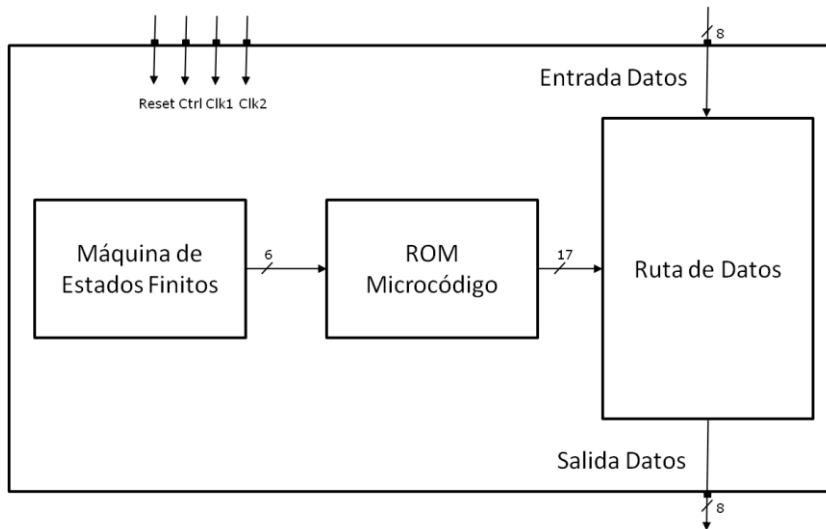


Figura 1: Esquema general de la arquitectura del controlador.

El esquema general de la arquitectura del microcontrolador se muestra en la figura 11. El sistema se compone de una pequeña máquina de estados finitos que genera direcciones para una memoria ROM. Esta memoria contiene el microcódigo, es decir las palabras de control que van a dirigir el funcionamiento de la ruta de datos. El chip tiene 20 pines, 14 de entrada y 8 de salida; en concreto hay un pin de control, un pin de reset, dos pines de reloj, ocho pines para la entrada de datos y ocho pines para la salida de datos.

El funcionamiento del microcontrolador es sencillo: el sistema permanece inactivo hasta que se recibe el código de 6 bits “100101” en la señal `Ctrl`. Una vez recibido este código, se registran los tres siguientes bits que entren por la señal de `Ctrl` que forman una palabra de tres bits llamada `número_de_repeticiones`. En el ciclo siguiente, la máquina de estados finitos comienza a generar direcciones para la memoria ROM empezando desde la dirección más baja y avanzando una dirección en cada ciclo. Cuando termina de recorrer todas las direcciones de la ROM, se vuelve a la primera dirección tantas veces como marque el valor almacenado en `número_de_repeticiones`. Cuando se llega al final de todas las repeticiones, el sistema se queda inactivo, pendiente de la señal de `Ctrl` para comenzar de nuevo el proceso. Mientras tanto, la ruta de datos, con la ayuda de la secuencia de señales de control que le ha proporcionado la ROM ha ido adquiriendo los datos externos, procesándolos y finalmente ha producido los resultados necesarios. Si en algún momento se recibe la un pulso activo en la señal de `Reset`, el sistema se queda inactivo.

3.1 La ruta de datos

En la figura 2 tenemos el esquema de la ruta de datos con el ancho de palabra de cada una de las señales. Numerados del 0 al 16 tenemos los bits de control de la ruta, que van a coincidir con los bits del microcódigo almacenado en la memoria ROM. En la figura 3 se especifica el formato del microcódigo. La figura 4 resume las operaciones que realiza la ALU. Es obligatorio implementar la ALU con una estructura en bit-slice.

¹ El diseño del microcontrolador está basado en las prácticas de laboratorio de la asignatura Digital Design with HDL IL2217 que imparte el profesor Johnny Öberg en el Real Instituto de Tecnología de Suecia

Diseño del Trazado de un Microcontrolador

La operación 3-Bit Check comprueba si son iguales el bit actual y los dos bits anteriores de los operandos A y B. Más formalmente, si $Z = 3\text{-Bit Check}(A,B) \rightarrow z_i = 1$ si $a_i = a_{i-1} = a_{i-2}$ o $b_i = b_{i-1} = b_{i-2}$. Para los dos bits menos significativos el valor de esta operación es indiferente. Ejemplos:

3-Bit Check(10101010, 10010010)=000000XX

3-Bit Check(10101010, 00000000)=111111XX

3-Bit Check(10001101, 01101110)=010010XX

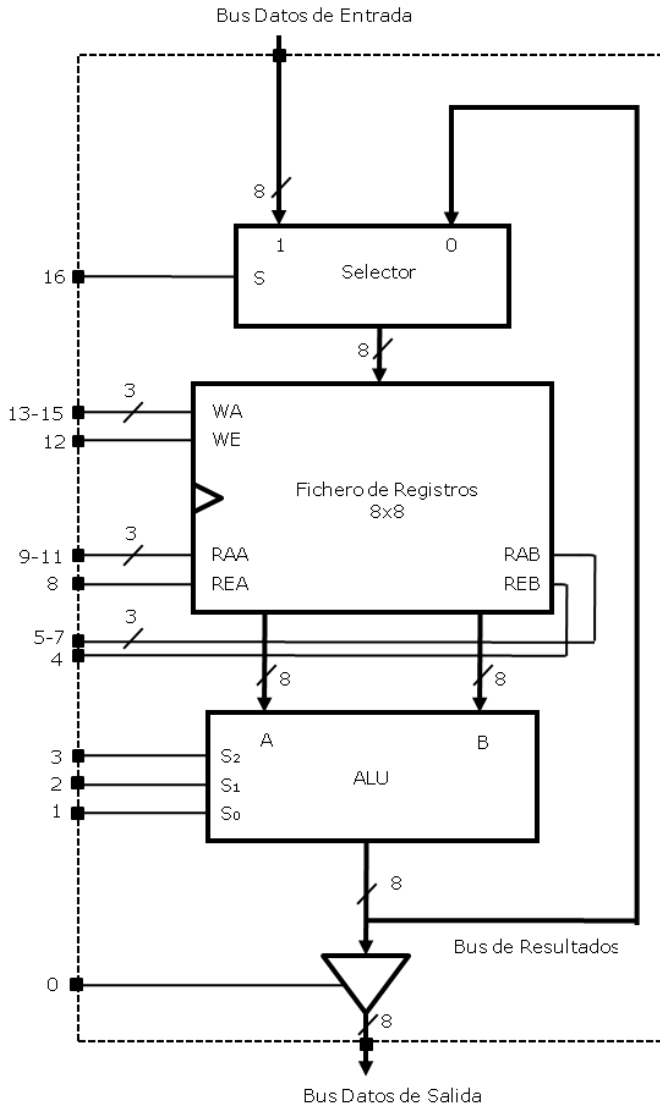


Figura 2: Esquema de la ruta de datos.

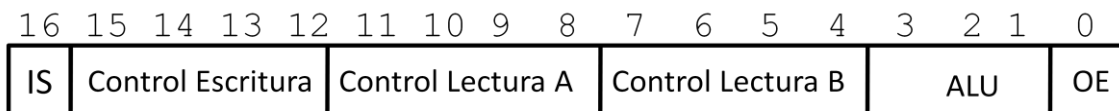


Figura 3: Formato de la palabra de control.

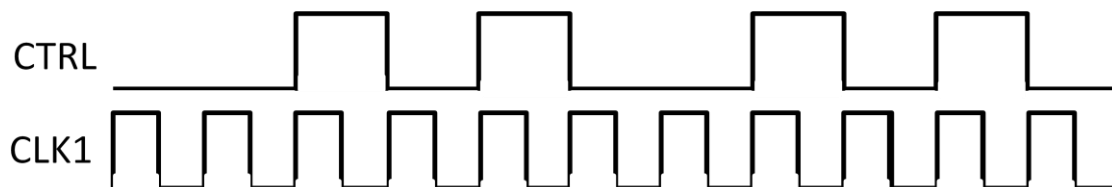
3	2	1	Operaciones ALU
0	0	0	AND
0	0	1	OR
0	1	0	XOR
0	1	1	A invertido
1	0	0	Carga literal
1	0	1	Suma
1	1	0	3 x A
1	1	1	3-Bit Check

Figura 4: Tabla de operaciones de la ALU.

3.2 La máquina de estados finitos

El programador que emplee el sistema tiene dos señales para controlarlo, la señal de `Ctrl` y la señal `Reset`. El sistema permanecerá inactivo hasta que el programador escriba en la señal `Ctrl` en serie la palabra "100101", empezado con el bit que está más a la derecha, el menos significativo. Una vez enviados estos 6 bits, el programador puede seleccionar el número de veces que se va a ejecutar el código almacenado en la ROM con siguientes 3 bits. "000" significa una sola repetición, "001" dos repeticiones, etc. Se envía primero el bit menos significativo.

Por ejemplo, si el programador quiere que haya tres repeticiones después de la primera ejecución (es decir, tres ejecuciones en total), tendrá que producir el siguiente cronograma:



Todo esto se traduce en el siguiente funcionamiento para la máquina de estados finitos:

- El sistema estará inactivo hasta la palabra "100101" en la señal de `Ctrl`.
- Se registran los tres siguientes bits de la señal de `Ctrl`.
- Seguidamente, se activa un contador que empezará a contar desde la dirección "000000" de la ROM, incrementándose en uno en cada ciclo.
- Cuando la cuenta llega a "111111", el contador deja de contar y se comprueba si quedan repeticiones pendientes. Si ese fuera el caso, se volvería a comenzar la cuenta que recorre todas las direcciones de la ROM. En caso contrario, se volvería al estado de inactividad.
- Si en algún momento se recibe un pulso activo en la de `Reset`, se pasaría inmediatamente al estado de inactividad.

3.3 La memoria ROM

La memoria ROM va a almacenar el microcódigo según se especifica en la figura 3. La memoria va a contener 64 palabras, es decir va a tener 6 bits de direccionamiento.

4 Plan de trabajo

En un principio, este proyecto debe dejar libertad completa al alumno para aplicar los conocimientos adquiridos en la asignatura. A pesar de ello, el diseño se ha planteado para realizarse en las siguientes etapas, que van a coincidir con los ejercicios propuestos durante el curso:

- Implementación de los módulos combinacionales que componen la ALU y el selector. Caracterización en términos de área, potencia y retardo de éstos módulos.
- Implementación y caracterización de la máquina de estados finitos y el contador de programa.
- Implementación y caracterización de la memoria ROM y el fichero de registros.
- Redacción del anteproyecto.
- Diseño y caracterización del plano de base completo, incluyendo el posicionamiento de los distintos bloques y los pads, interconexión (tanto de las señales de alimentación, masa y relojes, como del resto de señales) y estrategias de test.
- Finalización del proyecto y Redacción de la memoria final.

Este diseño goza de una gran libertad, por lo que cualquier simplificación para evitar tareas repetitivas que se quiera realizar está permitida siempre que esté justificada y no afecte al funcionamiento básico del microcontrolador. En caso de hacer simplificaciones se recomienda comentarlas antes con algún profesor de la asignatura.

Para realizar el seguimiento y la evaluación del proyecto, aparte de las entregas individuales de los ejercicios propuestos, los equipos de trabajo deben entregar dos documentos: el anteproyecto y la memoria final.

4.1 Revisión del diseño. Anteproyecto

La primera etapa es una breve memoria que se utilizará para revisar su diseño. Debe convencer con este anteproyecto de la viabilidad del plan previsto para el proyecto. Esta memoria debe contener:

1. Una idea general de su diseño, incluyendo un plano de base general y un plano general de la temporización del sistema.
2. Una breve descripción de cada bloque funcional, con su plano de color y esquemáticos.
3. Finalmente, un plano de conexión mostrado como diagrama de barras, no es necesario que esté a escala, pero que incluya cómo se haría el encaminamiento de las señales de alimentación y masa.

Con excepción de los diagramas, esta memoria **no debe tener más de 4 páginas**.

4.2 Memoria final

La memoria final debe describir el diseño completamente. Es decir, el camino seguido desde las especificaciones hasta el plano de base detallado. Sugerimos el siguiente índice:

4. Resumen
5. Introducción y visión global. Formulación del problema, idea general de la solución. Esquema de bloques general. Esquema de temporización.
6. Descripción detallada de cada célula. Algoritmos, diagrama de bloques, esquemáticos, temporización, caracterización de consumo, retardo y área para cada célula.
7. Descripción de las mejoras. Algoritmos, diagrama de bloques, esquemáticos, temporización, caracterización de consumo, retardo y área para cada mejora.
8. Descripción del sistema completo. Layout completo, plano de base, plano de conexión, plano de temporización, plano de distribución de las señales de masa y alimentación, plano de distribución de la señal de reloj, caracterización de consumo, retardo y área para el sistema completo.
9. Conclusiones.

10. Apéndice A: Trazados (trazados y esquemáticos de cada célula y trazado global).
11. Apéndice B: Resultados (tabla que comprenda la caracterización de cada célula y la caracterización global en términos de área, retardo y consumo).

Además de este índice, la memoria debe ser **sólo** lo suficientemente larga como para describir lo que Ud. ha hecho. Sea conciso. Lleve todo lo que sean detalles a los apéndices, e intente hacer una memoria atractiva de leer.

A continuación se incluye más información sobre lo que se espera de su memoria.

4.2.1 Resumen

Con no más de 150 palabras, sintetice el trabajo realizado en el proyecto. Incluya una breve descripción de las mejoras realizadas, si es que las hubiera.

4.2.2 Introducción y visión global

Describa el diseño en términos generales. Dado que se le han entregado las especificaciones básicas, no necesita discutir las con ningún detalle, aunque puede incluirlas como un apéndice. Sin embargo, sí debe explicar cualquier mejora, diferencia respecto a las restricciones de diseño de ellas, o cualquier aspecto que juegue un papel importante en su solución. Utilice figuras para representar la idea global. Piense que el objetivo es centrar a un lector sobre el objeto y contenido del trabajo que sigue. Recuerde: “una imagen vale más que mil palabras”.

Después de la idea general, continúe con una descripción más detallada de los algoritmos involucrados y de los diagramas que muestren cómo fluye la información. Discuta la arquitectura que haya elegido, dando un diagrama de bloques físicos para ella, y explique las decisiones tomadas durante el diseño para llegar a dicha arquitectura.

Explique la temporización del sistema. Para ello, incluya un plano detallado de la temporización utilizando el criterio de dos colores explicado en clase. Identifique claramente con qué color corresponde cada elemento del sistema y si ha utilizado alguna extensión del modelo inicial.

4.2.3 Descripción detallada de cada célula

Para cada una de las células que componen el sistema explique cómo ha realizado el diseño, si lo ha hecho partiendo de cero, o si ha reutilizado algún diseño existente. Si el diseño lo ha realizado a mano, incluya un esquemático o un diagrama lógico que explique el funcionamiento.

Caracterice el área, el consumo y el retardo. Este es un aspecto muy importante de la evaluación del proyecto, por lo tanto incluya todos los detalles que pueda sobre el modo de realización de la caracterización. En concreto para la caracterización del retardo, explique exactamente qué transistores y condensadores están involucrados en el camino crítico de cada módulo combinacional. Para la caracterización del consumo explique detalladamente qué vectores de prueba se han empleado, la frecuencia de conmutación de los mismos y cuál era la capacidad en cada salida del módulo.

Según aumente la complejidad de las células, la complejidad de las caracterizaciones de consumo y retardo se harán igualmente mayores. Para células pequeñas le será posible realizar simulaciones empleando trazados, para células medias empleará simulaciones en esquemáticos y para macrocélulas y el sistema completo tendrá que extrapolar datos de medidas simplificadas haciendo las suposiciones que considere necesarias. De cualquier modo, no deje ninguna célula sin caracterizar.

Explique cuidadosamente cualquier circuito con alguna particularidad o truco en la temporización que haya decidido utilizar. Para cada transistor de la célula, especifique en qué fase va a estar activo, el tipo de señales que le entran y el tipo de salidas que produce.

4.2.4 Plano de base

Explique el proceso de diseño y caracterización del plano de base completo, incluyendo el posicionamiento de los distintos bloques y los pads, interconexión (tanto de las señales de alimentación, masa y relojes, como del resto de señales). Para ello, incluya los siguientes elementos:

Layout completo.

Plano de base. Con los tamaños a escala de cada uno de los bloques y el espacio reservado para el conexionado.

Plano de conexionado. Planificación de las metalizaciones de las conexiones y su distribución.

Plano de temporización. Similar al realizado en la introducción, pero con los módulos finalmente implementados dando todo el detalle necesario para entender cualquier extensión incluida.

Plano de distribución de las señales de masa y alimentación

Plano de distribución de la señal de reloj.

Caracterización de consumo, retardo y área para el sistema completo. Especifique dónde se encuentra el camino crítico del sistema concretando sobre qué transistores pasa y qué capacidades tiene que cargar. Calcule la frecuencia máxima de funcionamiento del sistema, esta frecuencia tiene que coincidir con la frecuencia utilizada en las pruebas de consumo de todo el documento.

4.2.5 Test

Discuta cómo comprobaría su circuito. No dé una descripción extremadamente detallada del procedimiento de prueba, sino muestre cómo, bajo suposiciones razonables -por ejemplo, parte de la lógica de control es incorrecta-, proporciona un acceso adecuado al interior del chip. Para la comprobación (test) debe ser posible tanto el control como la observación de su estado; un contador de cuenta libre sin reset NO es aceptable. Describa cualquier hardware especial para test que haya incluido, tal como caminos serie para exploración, circuitos de inicialización, ...

Procure presentar su arquitectura de test de forma esquemática, de forma que resulte fácil comprender su idea.

4.2.6 Apéndices

En el Apéndice A incluya el trazado y el esquemático de todas las células individuales y el trazado del sistema global. Si quiere, puede ampliar alguna zona que le haya resultado especialmente compleja y explicar las soluciones que ha adoptado.

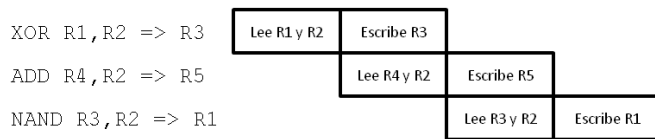
El Apéndice B será una hoja resumen con los resultados numéricos más importantes del proyecto. Aquí estarán incluidas todas las caracterizaciones de las células en términos de área, retardo y consumo (y cualquier otra métrica que considere interesante) junto con la caracterización del sistema completo. Incluya también el dato de la frecuencia máxima de funcionamiento del sistema.

5 Mejoras

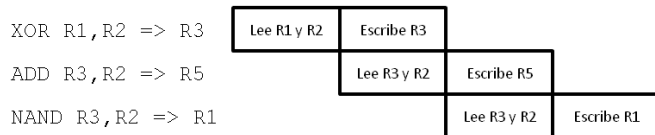
A continuación se proporciona una lista de mejoras propuestas por los profesores de la asignatura. Cada mejora será evaluada en función de su dificultad y la calidad del diseño y no tiene por qué necesariamente proporcionar los dos puntos extras. Además, se anima a los alumnos a que presenten sus propias propuestas de mejoras. Antes de realizar cualquier mejora, comuníquese a los profesores de la asignatura.

5.1 Control en Pipeline

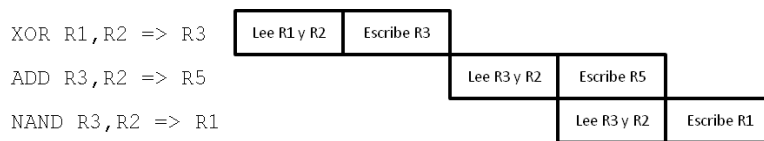
Realización de un control en pipeline que emplee un ciclo para la lectura de datos y otro para la escritura:



El sistema debe de ser capaz de detectar conflictos de datos como en el siguiente ejemplo:



E introducir un *stall* de un ciclo para evitarlos:



5.2 Instrucciones de Control

Adicción de un bit a la palabra de control para introducir la posibilidad de incluir estructuras de control en el programa. Si el bit es '0', la palabra se utiliza como se ha descrito anteriormente:

18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	IS	Control Escritura				Control Lectura A				Control Lectura B				ALU			OL	

Si el bit es '1', la palabra cambia a:

18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Control	Dirección Memoria Programa						Indiferente										

Con el siguiente control:

17 16 Operaciones de control		
0	0	CALL – Llama Subrutina
0	1	RET – Retorno de Subrutina
1	0	GOTO – Ir a dirección
1	1	BZ – Ir a dirección si el último resultado de la UF 1 es igual a 0

En este caso habría que incluir un contador de programa y una estructura de pila de programa asociada.

5.4 Operaciones adicionales multiciclo

Para esta mejora se sustituirá una de las palabras de control de la ALU por una operación externa a ésta que se podrá realizar en varios ciclos de reloj. Ejemplo de operaciones son multiplicación, logaritmo, seno, coseno, etc.

5.5 Sustitución de la ROM de programa por una memoria RAM

Empleo de una memoria RAM para la memoria de programa e inclusión de un mecanismo para que esta memoria se pueda escribir desde fuera del chip.

5.6 Implementación real de estrategias de test

Implementación de elementos de test que ayuden a la comprobación del funcionamiento del circuito.

5.7 Estrategias para tolerancia a errores

Implementación de técnicas de diseño que garanticen la protección frente a eventuales errores.

5.8 Estrategias para bajo consumo

Implementación de técnicas de diseño orientadas a la reducción del consumo del sistema.

5.9 Estilos de lógica alternativos

Implementación de las funciones combinacionales con familias lógicas distintas al CMOS convencional.

5.10 Circuitería de entrada/salida customizada

En esta mejora en lugar de emplear las células de la biblioteca para implementar la circuitería asociada a los pads, ésta se diseñará e implementará.

6 Entrega del Proyecto

La memoria se entregará en la plataforma web de la asignatura como un único archivo comprimido “.zip” o “.rar” en el que se incluirá la propia memoria en formato “.pdf” junto con todos los trazados, pruebas, esquemas, imágenes y datos que considere relevantes para una mejor comprensión del proyecto. Además se entregará una copia impresa de la memoria.

Con respecto al formato de nombre de los distintos ficheros, se establece que comiencen con grupo<número de grupo>_ y sigan con la explicación más clara posible del contenido del archivo. Por ejemplo, si el grupo 3 quisiera incluir el trazado de un sumador completo, lo nombraría grupo3_FullAdder. El archivo comprimido de la entrega se llamará grupo<número>.rar o grupo<número>.zip (e.g. grupo3.zip o grupo3.rar).

7 Críticas

Se le anima a hacer por escrito una crítica del curso, anónima si quiere al margen de la memoria del proyecto. El interés fundamental es conocer la adecuación, tanto a un nivel conceptual como concreto, de la clase (si considera que su opinión no quedó suficientemente reflejada en la encuesta final), así como sugerencias para mejorar esta asignatura.

Si tiene problemas con su compañero/a debe hacerlo saber. No todos los compañeros trabajan. No debe confundirse compañerismo con encubrimiento de los que “pasan” a costa de los demás. Si no existe información en contra, debe suponerse que todo marcha bien y, por tanto, el trabajo será evaluado como el resultado de un equipo de tres personas.