



DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Simulación del computador simple

1. Introducción y objetivos

- Objetivo general: utilizar el entorno Xilinx ISE para describir y simular los computadores simples CS1 y CS2.

- Objetivos específicos:

- Identificar y completar un programa escrito en código máquina en la memoria de programa.
- Reconocer los ciclos de fetch y execute durante la ejecución del programa.
- Realizar cambios en la unidad de control.
- Calcular tiempos de ejecución.

Durante la sesión de laboratorio debe utilizar una carpeta llamada CS1 que contiene los ficheros indicados en la tabla 1 y otra llamada CS2 (tabla 2). Estas están disponibles en el directorio correspondiente a esta práctica en la web de la asignatura.

Nombre del fichero	Contenido
ComputadorSimple1.v	Descripción a nivel estructural en verilog del CS1.
UnidadControlCS1.v	Descripción verilog de la unidad de datos del CS1.
UnidadDatosCS1.v	Descripción verilog de la unidad de control del CS1.
CS1_tb.v	Testbench para la simulación del CS1.
CS1.wcfg	Fichero de configuración de ondas para la simulación.

Tabla 1: Ficheros de la carpeta CS1 (necesarios para realizar la práctica)

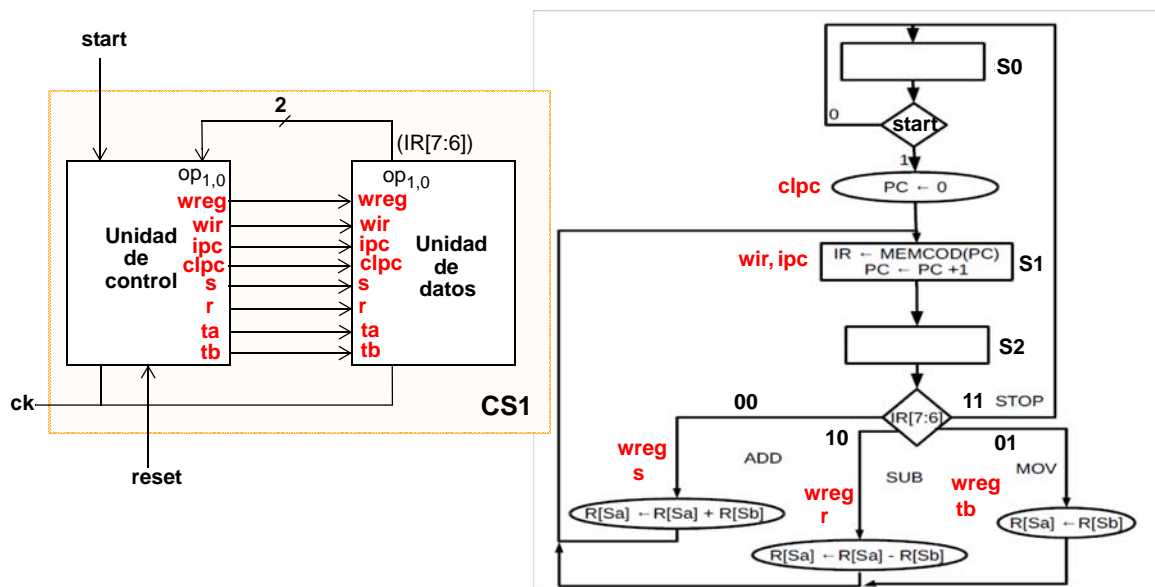
Nombre del fichero	Contenido
ComputadorSimple2.v	Descripción a nivel estructural en verilog del CS2.
UnidadControlCS2.v	Descripción verilog de la unidad de datos del CS2.
UnidadDatosCS2.v	Descripción verilog de la unidad de control del CS2.
CS2_tb.v	Testbench para la simulación del CS2.
CS2.wcfg	Fichero de configuración de ondas para la simulación.

Tabla 2: Ficheros de la carpeta CS2 (necesarios para realizar la práctica)

2. Estudio teórico (leerlo con atención es obligatorio)

Computador simple 1: CS1

En la figura se muestra el diagrama de bloques (U. Datos/U. Control) del CS1 así como la carta ASM asociada.



Puede comprobar en el fichero correspondiente (ComputadorSimple1.v) que se ha realizado un diseño jerárquico conectando estructuralmente los módulos unidad de datos y unidad de control. Dichos módulos también han sido descritos mediante Verilog.

- La descripción de la unidad de control se ha realizado por medio de su carta ASM.
- Para la unidad de datos se han descrito en un único fichero todos los registros y componentes (PC, IR, ALU y memoria ROM) y se han conectado adecuadamente. Puede comprobar que en primer lugar aparece la conexión entre componentes y posteriormente la descripción de los mismos. También se han inicializado algunos de los registros mediante un bloque *initial*. Estos registros serán los utilizados por el programa que vamos a ejecutar. Es **importante** que identifique dentro del fichero el lugar donde se ha inicializado la memoria con dicho programa.

El programa que se pretende ejecutar con el CS1 realiza la operación $R6 \leftarrow 3R4 - 2R1$, y es el siguiente:

```

MOV R6,R4
SUB R6,R1
ADD R6,R6
ADD R6,R4
STOP

```

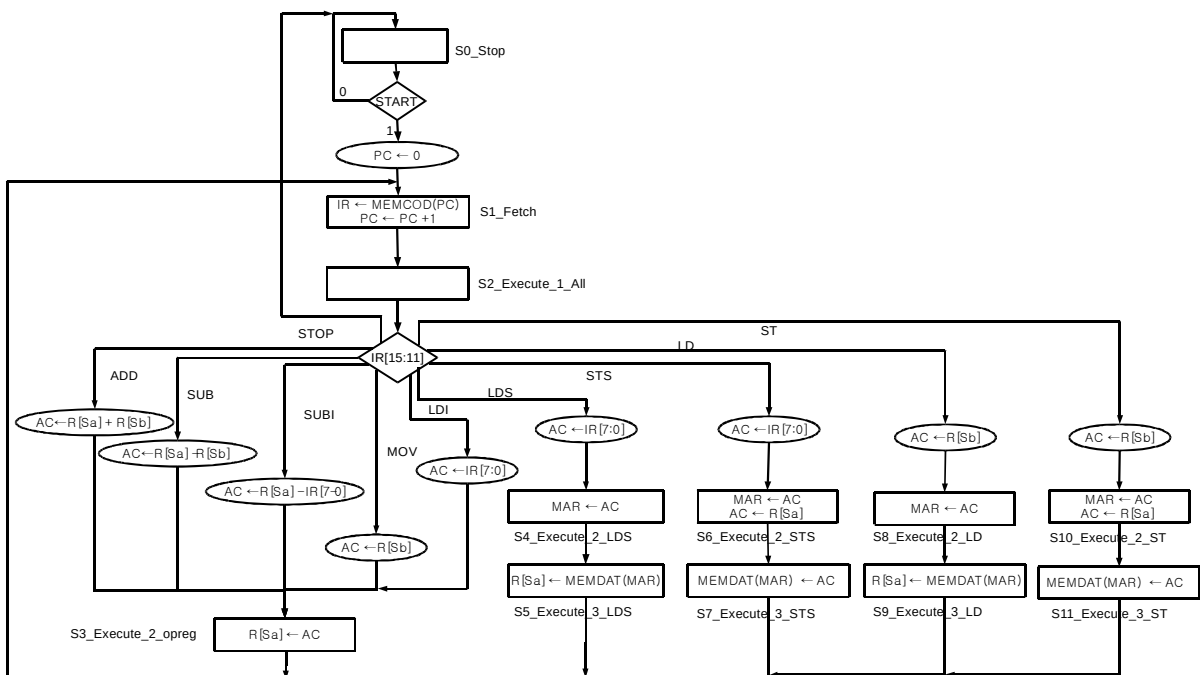
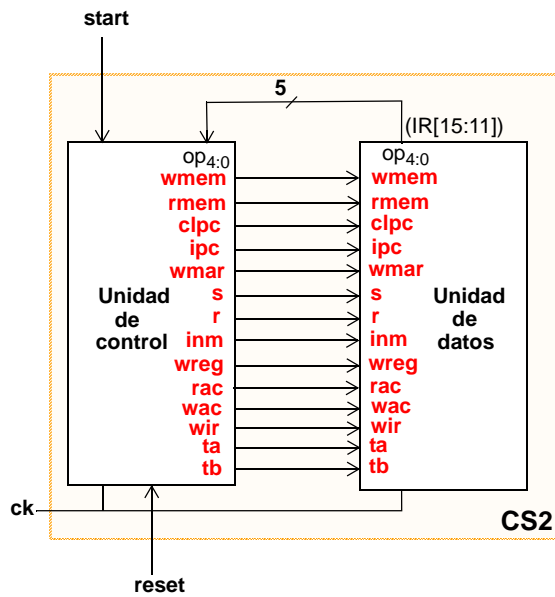
Pos	Contenido
\$00	01 110 100
\$01	10 110 001..
\$02	00 110 110
\$03	00 110 100
\$04	11 --- ---

MEMORIA DE CÓDIGO

Para obtener su código máquina y poder revisarlo o modificarlo en la memoria de código tendrá que tener en cuenta el formato de instrucción.

Computador simple 2: CS2

En la figura se muestra el diagrama de bloques (U. Datos/U. Control) del CS2 así como la carta ASM asociada.

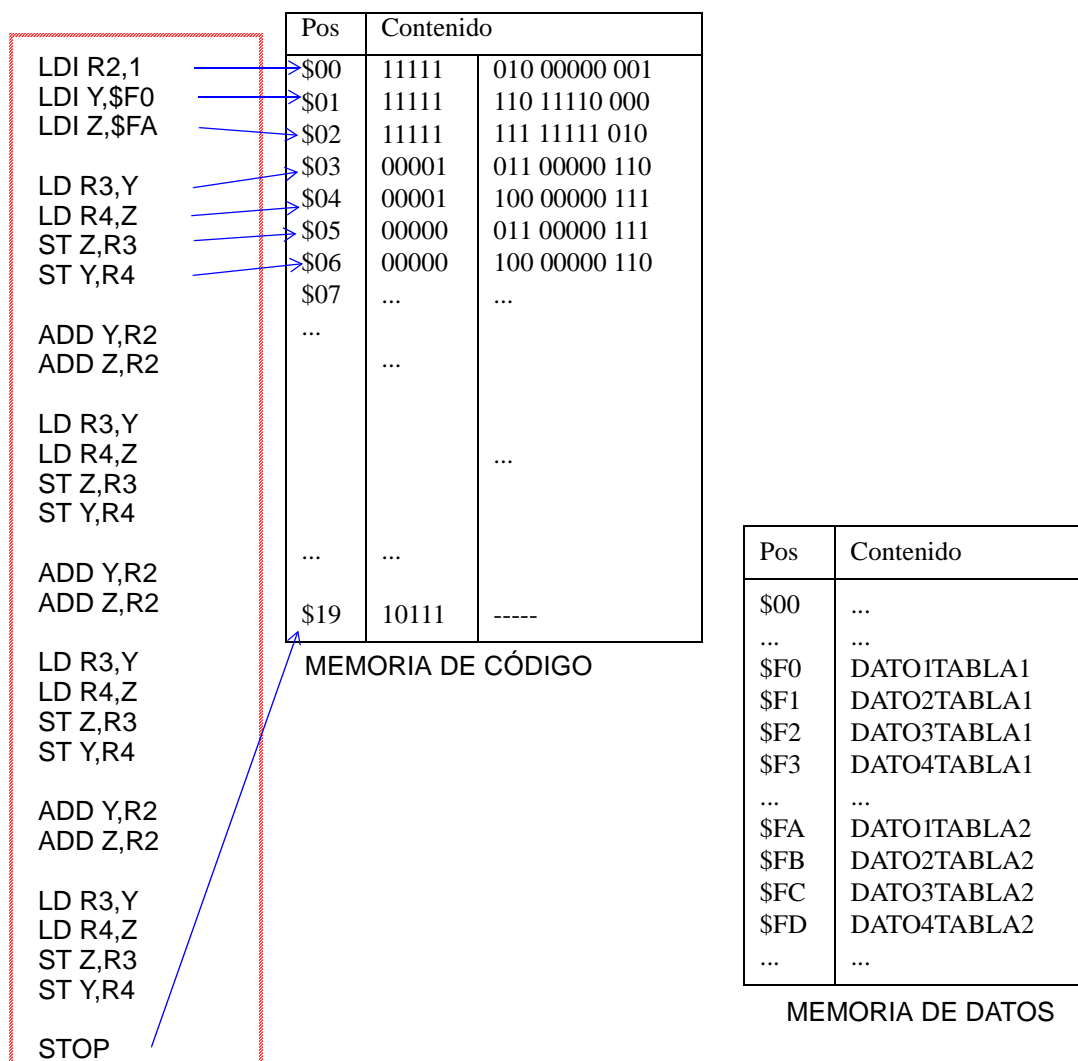


Nuevamente, como en el caso del CS1, se ha realizado un diseño jerárquico conectando estructuralmente los módulos unidad de datos y unidad de control. Dichos módulos también han sido descritos mediante Verilog.

- La descripción de la unidad de control se ha realizado por medio de su carta ASM. Se han definido los códigos de operación de las instrucciones utilizando parámetros con el fin de hacerlos más fácilmente identificables. Es conveniente que revise este fichero y vea donde se han definido las instrucciones ya que durante la práctica tendrá que realizar cambios.

- Para la unidad de datos se han descrito en un único fichero todos los registros y componentes (PC, IR, AC, MAR, MUX2:1, ALU y las memorias de dato y de código) y se han conectado adecuadamente. Puede comprobar que en primer lugar aparece la conexión entre componentes y posteriormente la descripción de los mismos. También se han inicializado algunos de los registros y de las posiciones de memoria de datos mediante un bloque *initial*. Es **importante** que identifique dentro del fichero el lugar donde se ha inicializado la memoria con el programa a ejecutar.

El programa a ejecutar en el CS2 debe intercambiar dos tablas de 4 datos que se encuentran almacenadas en la memoria y cuyas direcciones base están guardadas en los registros Y(R6) y Z(R7) respectivamente. El registro R2 tiene almacenado el valor 1 y se utiliza para incrementar las direcciones almacenadas en los registros base y acceder a los diferentes elementos de las tablas. El programa es el siguiente:

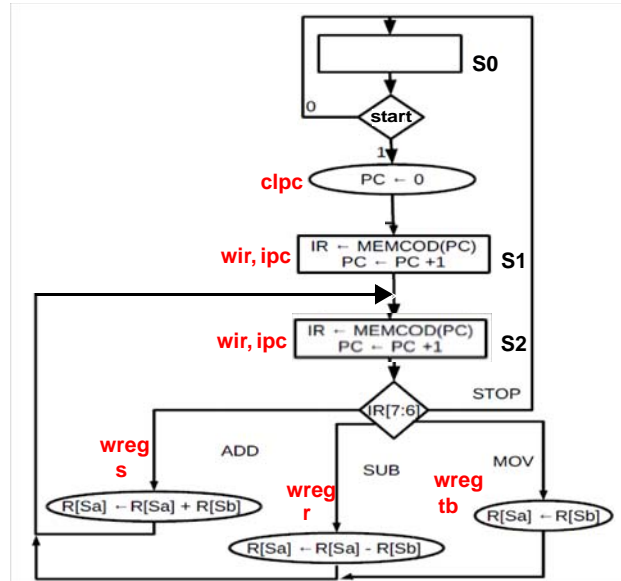


Para obtener el código máquina, se ha tenido en cuenta el formato de instrucción. Puede completar dicho código si lo desea.

3. Estudio experimental

Bloque I: Simulación del CS1

- 1- Cree un proyecto denominado ComputadorSimple1.
- 2- Siga los pasos habituales para añadir al proyecto todos los ficheros de la carpeta CS1.
- 3- Localice y corrija el error que aparece en la descripción Verilog de la **unidad de control**.
- 4- Localice el programa escrito en código máquina en la memoria de programa de la **unidad de datos**. Dicho programa aparece incompleto, complételo.
- 5- Simule el circuito del CS1 utilizando el *testbench* proporcionado: test_cs1.v. Para comprobar el buen funcionamiento del sistema le será útil cargar el fichero de ondas que se le ha suministrado, para ello siga los siguientes pasos: en el entorno de simulación (ISim) pique sobre la pestaña File -> Open. En el cuadro de diálogo seleccione el fichero CS1.wcfg. Si a continuación le aparece un nuevo cuadro de diálogo seleccione la opción "connect to existing". Una vez hecho esto, pique los iconos correspondientes para comenzar de nuevo la simulación desde el entorno ISim (sin cerrarlo).
- 6- A partir de este momento puede identificar en la simulación los hechos que se producen desde el comienzo de la operación (pulso de start):
 - observe los valores que va tomando el contador de programa
 - examine los estados por los que pasa la unidad de control
 - identifique los ciclos de búsqueda y ejecución
 - fíjese en las señales de la ALU que se van activando según la instrucción ejecutada
 - supervise cómo cada una de las instrucciones de la memoria de programa van pasando por el registro de instrucciones IR
 - puede ver también cómo los buses *sa* y *sb* van tomando los valores necesarios para seleccionar los registros fuente y destino en cada instrucción
 - no olvide comprobar si el resultado de la operación del programa es correcto, para ello puede ver los valores de los registros implicados en cada momento.
 - MIDA el tiempo que tarda el computador en ejecutar el programa y anote este valor
- 7- A continuación debe realizar una optimización de la carta ASM para conseguir una mejora en la velocidad del sistema. La optimización consistirá en permitir que en un mismo ciclo de reloj se ejecute la instrucción actual y se busque la siguiente. La modificación a realizar debe permitir que tras la búsqueda de la primera instrucción (estado S1 de la carta) no se vuelva a este estado para buscar otras instrucciones sino que las operaciones de búsqueda se incluyan en S2 y el bucle de búsqueda-ejecución se produzca solo sobre S2. En la siguiente figura se ilustra este cambio.



8- Modifique el código verilog de la unidad de control para incluir la optimización propuesta en el punto anterior.

9- Simule el nuevo sistema (utilice el mismo fichero de ondas CS1.wcfg) y observe las diferencias con la simulación anterior. Concretamente: vea la nueva secuencia de estados, MIDA el tiempo que tarda el computador optimizado en ejecutar el programa y compare con el valor anotado en el apartado 6.

Bloque II: Simulación del CS2

10- Cree un proyecto denominado ComputadorSimple2.

11- Siga los pasos habituales para añadir al proyecto todos los ficheros de la carpeta CS2.

12- Localice el programa escrito en código máquina en la memoria de programa de la **unidad de datos**. Dicho programa aparece incompleto, complételo.

13- Simule el circuito del CS2 utilizando el *testbench* proporcionado: test_cs2.v. Para comprobar el buen funcionamiento del sistema le será útil cargar el fichero de ondas que se le ha suministrado, para ello siga los siguientes pasos: en el entorno de simulación (ISim) pique sobre la pestaña File -> Open. En el cuadro de diálogo seleccione el fichero CS2.wcfg. Si a continuación le aparece un nuevo cuadro de diálogo seleccione la opción "connect to existing". Una vez hecho esto, pique los iconos correspondientes para comenzar de nuevo la simulación desde el entorno ISim (sin cerrarlo).

14- A partir de este momento puede identificar en la simulación los hechos que se producen desde el comienzo de la operación (pulso de start):

- compruebe el resultado de la operación del programa, para ello **despliegue** las líneas de MEMDAT y observe si las tablas se han intercambiado
- observe los estados por los que pasa la unidad de control (tenga en cuenta que el códigos asignados a cada estado viene determinado por su nombre):

por ejemplo:

S5_Execute_3_LDS → código 5

S8_Execute_2_ST → código 8

- identifique los ciclos de búsqueda y ejecución y observe cómo el tiempo de ejecución es distinto para distintas instrucciones
- compruebe cómo cada una de las instrucciones de la memoria de programa van pasando por el registro de instrucciones IR
- siga en la simulación cómo cambian los contenidos de los registros de datos
- puede ver también otras señales si así lo desea como las señales de control de la ALU o de los registros, para ello tendrá que incorporar dichas señales a la ventana de simulación y resimular sin salir de ISim
- MIDA el tiempo que tarda el computador en ejecutar el programa y anote este valor

15- Añada la instrucción ADDI al computador, para ello necesitará añadir algunas líneas en varios puntos del fichero UnidadControl.v. Aproveche el paralelismo de la instrucción ADDI con la ya implementada SUBI. Necesitará escoger un opcode que aún no se haya usado, darle nombre y añadir un ítem al case del estado "S2_Execute_1_All" similar al de SUBI.

16- Modifique el programa utilizando la nueva instrucción. Escríbalo primero con mnemónicos, obtenga después el código máquina y, a continuación, inclúyalo en la memoria de programa. Simule de nuevo el sistema y tras comprobar que funciona adecuadamente, MIDA el tiempo que se tarda ahora en ejecutar el programa y compare con el valor anotado en el apartado 14.