



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

ESTRUCTURA DE COMPUTADORES

GRADO EN INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA DE COMPUTADORES

Actividades prácticas

Simulación de programas en el camino de datos unicycle de MIPS

CURSO 2019-20

Jorge Ángel Colinas García

Susana Mata Fernández

Luis Rincón Córcoles

Arquitectura y Tecnología de Computadores

INTRODUCCIÓN

En esta actividad se simularán programas en bajo nivel sobre un modelo de camino de datos unicycle para MIPS realizado en VHDL. La descripción del modelo y del repertorio de instrucciones que soporta se realiza en el documento “**Modelos de caminos de datos unicycle y multiciclo para MIPS**”, disponible en el aula virtual de la asignatura.

Los pasos realizados serán los siguientes:

1. Crear un programa ensamblador de **MIPS** sobre el entorno **MARS** que utilice únicamente instrucciones del repertorio reducido, y verificar su funcionamiento.
2. Mediante un entorno de desarrollo y simulación para **VHDL**, comprobar el funcionamiento del programa anterior en un modelo **VHDL** de **MIPS** unicycle mediante análisis de cronogramas, respondiendo a las preguntas de un test en el aula virtual (el test se publicará más adelante).
3. Mediante el análisis de los cronogramas generados por **ModelSim**, anotar las informaciones producidas en la simulación en una ficha en formato **Excel (.xlsx)**.

Estas actividades se podrán realizar tantas veces como se desee, para diferentes programas de prueba.

TODAS LAS ACTIVIDADES SE REALIZARÁN DE FORMA INDIVIDUAL.

TRABAJO PREPARATORIO

Es imprescindible estudiar los contenidos teóricos relativos a lenguaje ensamblador y camino de datos unicycle al completo.

También es recomendable repasar los conceptos de **VHDL** aprendidos en la asignatura de Tecnología de Computadores.

ANTES DE COMENZAR

El material necesario para realizar la actividad pedida se encuentra en diferentes archivos:

- **Laboratorio_caminos_datos.pdf**: aquí se presenta un manual de uso de los modelos VHDL utilizados en esta actividad práctica y la siguiente, dedicada al camino de datos multiciclo.
- **Laboratorio_caminos_datos_apoyo.zip**: aquí está el modelo VHDL utilizado, junto con las plantillas de generación de cronogramas, y un ejemplo en ensamblador junto con los ficheros imagen de los segmentos de texto y datos de MIPS.
- **fichas_unicycle.xlsx**: fichero con las fichas de instrucciones que hay que rellenar.

PARTE 1 (OBLIGATORIA)

Los pasos que hay que realizar son los siguientes:

1. En primer lugar, será preciso crear un proyecto en **ModelSim** con los archivos que conforman el modelo de procesador utilizado.
2. El siguiente paso es compilar el proyecto.
3. Después se simulará el camino de datos unicycle cuyos ficheros **Memoria_Instrucciones.txt** y **Memoria_Datos.txt** se incluyen con el código de apoyo, y se responderán las preguntas del test en el aula virtual mediante el análisis de las formas de onda obtenidas mediante la simulación. ESTE PASO ES OBLIGATORIO.

En el apéndice A de este enunciado se muestra el cronograma resultante de la simulación.

PARTE 1: TEST

1. ¿Cuál es la instrucción ejecutada en el ciclo 0?
 - a) lui.
 - b) lw.
 - c) add.
 - d) addi.
 - e) Ninguna, es el reset.**
2. ¿Cuál es la instrucción ejecutada en el ciclo 1?
 - a) lui.**
 - b) lw.
 - c) add.
 - d) addi.
 - e) Ninguna, es el reset.
3. ¿Qué señal indica cuál el registro destino de la instrucción del ciclo 1?
 - a) Read_Data1.
 - b) Read_Data2.
 - c) WriteReg.
 - d) Ninguna de las restantes respuestas es cierta.**
4. ¿De qué campo de la instrucción procede el indicador del registro destino en la instrucción 1?
 - a) rs.
 - b) rt.**
 - c) rd.
 - d) Ninguna de las restantes respuestas es cierta.
5. ¿Cuál es la instrucción ejecutada en el ciclo 2?
 - a) lui.
 - b) lw.**
 - c) add.
 - d) addi.
 - e) Ninguna de las restantes respuestas es cierta.
6. ¿Cuál es el registro destino en la instrucción ejecutada en el ciclo 2?
 - a) 0.
 - b) 1.
 - c) 16.**
 - d) 17.
 - e) Ninguna de las restantes respuestas es cierta.
7. ¿En qué ciclo se ejecuta por primera vez la instrucción `beq $t0,$s0,end_bucle`?
 - a) 4.
 - b) 5.**
 - c) 8.
 - d) 10.
 - e) Ninguna de las restantes respuestas es cierta.
8. ¿Cuántas veces se ejecuta la instrucción `beq $t0,$s0,end_bucle` a lo largo del programa?
 - a) 0.
 - b) 1.
 - c) 4.
 - d) 5.**
 - e) Ninguna de las restantes respuestas es cierta.

9. ¿Hay algún ciclo en el que las señales Branch y Zero valgan 1 y causen que la ramificación sea efectiva?
- a) No, esto no sucede nunca.
 - b) Sí, en el ciclo 3.
 - c) Sí, en el ciclo 13.
 - d) Sí, en el ciclo 21.**
 - e) Ninguna de las restantes respuestas es cierta.
10. ¿En qué ciclo se ejecuta por primera vez una instrucción de salto o ramificación que es efectiva, es decir, que produce un salto que rompe la secuencia de ejecución de instrucciones?
- a) En el ciclo 5.
 - b) En el ciclo 8.**
 - c) En el ciclo 9.
 - d) En el ciclo 13.
 - e) Ninguna de las restantes respuestas es cierta.
11. ¿Cuántas veces en el programa se rompe la secuencia de ejecución de instrucciones por efecto de un salto o una ramificación efectiva? 5
12. ¿En qué ciclo se ejecuta la instrucción `addi $t0,$t0,1` por segunda vez? 10
13. ¿Qué instrucción se ejecuta en el ciclo 8?
- a) `add`
 - b) `addi`
 - c) `beq`
 - d) `j`**
 - e) `sw`
14. ¿Qué instrucción se ejecuta en el ciclo 14?
- a) `add`
 - b) `addi`**
 - c) `beq`
 - d) `j`
 - e) `sw`
15. ¿Qué instrucción se ejecuta en el ciclo 19?
- a) `add`
 - b) `addi`**
 - c) `beq`
 - d) `j`
 - e) `sw`
16. ¿Qué instrucción se ejecuta en el ciclo 23?
- a) `add`
 - b) `addi`
 - c) `beq`
 - d) `j`
 - e) `sw`**
17. ¿Cuántas veces se ejecutan instrucciones `addi` en el programa? 6
18. ¿Cuántas veces se realiza una suma en la unidad aritmético lógica a lo largo de la ejecución del programa? (contar desde el ciclo 1 inclusive, no se deben contar las instrucciones `nop`) 12?
19. ¿Cuántas de las sumas son forzadas, es decir, no dependen del campo de función de la instrucción? 6?
20. ¿Cuántas veces se ejecutan instrucciones de tipo R en el programa? (no se deben contar las instrucciones `nop`) 8
21. ¿Cuántas veces se realiza una resta en la unidad aritmético lógica a lo largo de la ejecución del programa? (contar desde el ciclo 1 inclusive) 4

22. ¿En qué ciclo se realiza una lectura sobre la memoria de datos?
- a) En todos los ciclos.
 - b) En el ciclo 2.**
 - c) En el ciclo 23.
 - d) Nunca se realiza una lectura sobre la memoria de datos.
23. ¿En qué ciclo se realiza una escritura sobre la memoria de datos?
- a) En todos los ciclos.
 - b) En el ciclo 2.
 - c) En el ciclo 23.**
 - d) Nunca se realiza una escritura sobre la memoria de datos.
24. ¿Cuál es la dirección de la última instrucción del programa (en hexadecimal)?
- a) 0040002C.
 - b) 00400030.**
 - c) 00400034.
 - d) 00400038.
 - e) Ninguna de las restantes respuestas es cierta.
25. ¿En qué ciclo se ejecuta la última instrucción del programa?
- a) 20.
 - b) 26.**
 - c) 27.
 - d) El programa no termina nunca de ejecutarse.
 - e) Ninguna de las restantes respuestas es cierta.

PARTE 2 (OPCIONAL)

Esta parte consiste en el análisis del cronograma creado con la simulación anterior y posterior cumplimentación de unas fichas incluidas en un archivo **Excel** con extensión **.xlsx**.

Adicionalmente, se podrán crear nuevos programas de prueba en **MARS**, generar los ficheros de volcado de memoria **Memoria_Instrucciones.txt** y **Memoria_Datos.txt**, tal como se indica en el apéndice A del documento “**Modelos de caminos de datos unicycle y multiciclo para MIPS**” (fichero **Laboratorio_caminos_datos.pdf**). A continuación, se simulará el nuevo caso de prueba en ModelSim y se comprobará su funcionamiento mediante el análisis de las formas de onda obtenidas, anotando las informaciones requeridas en las fichas. Para facilitar el seguimiento, se puede ir simulando al mismo tiempo el caso de prueba en **MARS** o **Simula3MS**. ESTA ACTIVIDAD ES OPCIONAL.

RELLENANDO EL ARCHIVO .xlsI

En esta parte de la práctica se analizará el comportamiento de las instrucciones soportadas por el modelo mediante la ejecución de al menos un caso de prueba. El objetivo consiste en rellenar un **archivo de fichas de ejecución** en formato **Excel** (con extensión **.xlsx**). Este archivo contiene una hoja (ficha) por cada ciclo de reloj, indicando:

- La instrucción concreta ejecutada en el ciclo.
- Los valores de las señales de control (que se expresarán en binario).
- Los valores de las señales de datos (que se expresarán en decimal o hexadecimal).

Los archivos de fichas de ejecución proporcionados constan de 6 fichas cada uno. En caso necesario, se pueden añadir más fichas sin más que duplicar las ya existentes antes de rellenarlas.

El análisis de la ejecución completa de un caso puede ser muy tedioso si el número de instrucciones ejecutadas es elevado (longitud del programa fuente, existencia de bucles, etc). Para aligerar el trabajo, **en**

el archivo de fichas de ejecución será suficiente con rellenar una por cada tipo de instrucción en el programa (una instrucción **lw**, una instrucción **add**, una instrucción **sub**, una instrucción **sw**, etc), sin necesidad de que se hayan ejecutado de forma consecutiva (para ello se indicará en cada caso el número de ciclo, indicado en la parte superior del cronograma).

Se puede analizar uno o varios casos de prueba, a elección del alumno. Entre todos los casos de prueba se deberán rellenar como mucho 12 fichas de ejecución, una por cada una de las instrucciones que figuran en el apéndice A del documento “**Modelos de caminos de datos uniciclo y multiciclo para MIPS**” (descontando **syscall** y **nop**).

Los casos de prueba pueden ser programas que tengan sentido (es decir, que realicen algún cálculo útil) o simplemente secuencias de instrucciones que no realicen ningún cálculo concreto. Se puede usar el caso de prueba proporcionado, o bien diseñarse casos de prueba nuevos.

ENTREGA DE LA ACTIVIDAD 2

Se entregarán en el campus virtual los archivos de prueba y las fichas generadas, todo ello empaquetado en un único archivo **.ZIP**. Este archivo contendrá **una subcarpeta por cada caso de prueba analizado**. Dentro de cada subcarpeta se incluirá la siguiente información:

- El archivo fuente de **MARS**.
- Los archivos de volcado **Memoria_Instrucciones.txt** y **Memoria_Datos.txt** generados con **MARS** para este caso de prueba.
- El archivo **fichas-uniciclo.xlsx** de **Excel** con las fichas de ejecución rellenas.

APÉNDICE A

CRONOGRAMA RESULTANTE DE LA SIMULACIÓN DEL EJEMPLO



