UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

ORGANIZACIÓN DE COMPUTADORES

LABORATORIO 1

Profesores: Alfonso Guzmán & Daniel Wladdimiro
Ayudantes: Pablo Ulloa & Ariel Undurraga

Santiago - Chile
9 de marzo de 2017

CAPÍTULO 1. CONTEXTO

Como lo confirma la teoría vista en cátedra, en la organización de computadores lo esencial es el camino de datos (datapath en inglés), el cual consiste en un conjunto de unidades funcionales (unidad de control, ALU, multiplexores, buses, registros, etc), donde sin la existencia de estos elementos, no sería posible ejecutar las instrucciones que se les da a un computador, por lo que en otras palabras, no existirían los computadores como los conocemos hoy en día.

Este camino de datos, como su nombre lo sugiere, representa el camino que recorren las señales eléctricas a través de las distintas unidades funcionales, con el fin de realizar una operación (una instrucción particular). Es por esto que cada instrucción básica, como las del lenguaje MIPS, enciende un conjunto acotado de unidades del camino de datos, las cuales se pueden evidenciar en las señales activas de la unidad de control. Ante esto surge la siguiente interrogante: ¿Qué unidades funcionales se activan al ejecutar un determinado conjunto de instrucciones?

CAPÍTULO 2. INSTRUCCIONES

Usted debe leer un archivo de entrada, el cual contendrá instrucciones de un programa MIPS. El programa debe ser capaz de generar dos archivos de salida, donde el primer entrega la traza de los registros de las instrucciones de entrada, y el segundo corresponde a los valores de la unidad de control para cada instrucción. Para efectos de este laboratorio, debe considerar que el camino de datos está siendo ejecutado en un procesador monociclo y utilizando los componentes que muestra en el Anexo.

Por ejemplo, en el caso que el archivo de entrada sea:

```
addi $t1, $zero, 2
addi $t2, $zero, 3
add $t0, $t1, $t2
add $t2, $t3, $t0
```

Cuando queramos leer la tercera instrucción `add $t0, $t1, $t2`, podemos mostrar la traza de cada uno de los registros y las líneas de control de la arquitectura del computador, en este caso, la arquitectura de un procesador MIPS de la siguiente manera:

Instrucción	RegDst	Jump	Branch	MemRead	MemToReg	ALUOp
`add $t0, $t1, $t2`	1	0	0	0	0	10

Instrucción	MemWrite	ALUSrc	RegWrite
`add $t0, $t1, $t2`	0	0	1

Cuadro 2.1: Salida del primer archivo

Instrucción	PC	Etapa	$t0	$t1	$t2	$t3
`add $t0, $t1, $t2`	8	IF	0	2	3	0
`add $t0, $t1, $t2`	8	ID	0	2	3	0
`add $t0, $t1, $t2`	8	EX	0	2	3	0
`add $t0, $t1, $t2`	8	MEM	5	2	3	0
`add $t0, $t1, $t2`	8	WB	5	2	3	0

Cuadro 2.2: Salida del segundo archivo

Cabe destacar que esto es una sugerencia y **no** la forma en como deben entregar el archivo de salida, el cual queda a criterio de cada persona. Y también no se han incluido **todos** los registros por motivos de espacio, lo cual si **debe** hacerse en el segundo archivo de salida.

Para efectos de la evaluación, se entregarán 3 archivos de entrada distintos: un programa secuencial, un programa con una función recursiva y un programa con ciclos, los cuales estarán disponibles en Moodle.

Junto con el programa, usted debe entregar un informe que cumpla con el formato tesis del Departamento de Ingeniería Informática de la Universidad de Santiago de Chile. Para efectos prácticos, se dejará disponible una plantilla en LaTeX en el Moodle, la cual posee las secciones a evaluar en la entrega del laboratorio.

En el desarrollo del informe se evaluará la forma en cómo usted dio solución al enunciado de este laboratorio, es decir, cómo decodificó el archivo de entrada, cómo realizó el seguimiento de las trazas, y así mismo, cómo presenta los resultados en el archivo de salida.

2.1 EXIGENCIAS

- El programa debe estar escrito en C o C++ (estándar ANSI C). En caso de usar C++ se exigirá orientación a objetos, de caso contrario no será revisado.
- El programa y el informe deben ser entregados como una carpeta comprimida en formato `zip` o `tar.gz`-
- El programa debe ser entregado en su código fuente junto a un archivo Makefile para su compilación.
- El programa debe de tener usabilidad. El usuario debe ser capaz de ingresar el nombre del archivo a leer y los nombres de los archivos de salida por una interfaz.
- El programa debe cumplir con un mínimo de calidad de software (funciones separadas y nombres tanto de funciones como de variables, de fácil comprensión).
- El informe escrito no debe exceder 10 páginas de texto escrito, sin considerar portada ni índicer, en caso contrario, por cada página extra, se descontará 5 décimas.
- El informe escrito debe ser entregado en formato PDF, por lo que puede ser desarrollado en LaTeX, Microsoft Word, OpenOffice Writer, etc.

2.2 RECOMENDACIONES

- Pueden usar estructuras de datos para almacenar la información de los registros y señales de control en caso de utilizar C, sino, debe utilizar un objeto para esto.
- Utilizar la plantilla de LaTeX disponible en el Moodle del curso.
- Consultar a los ayudantes y en Moodle del curso.

2.3 DESCUENTOS

- Por cada exigencia no cumplida, se descontarán dos décimas a la nota, a excepción las que ya mencionan su descuento.
- Por cada día de atraso, se descontará un punto a la nota.
- Por cada tres faltas ortográficas o gramaticales en el informe, se descontará una décima a la nota.
- Por cada falta de formato en el informe, se descontará una décima a la nota.

2.4 EVALUACIÓN

- La nota del laboratorio será el promedio aritmético del código fuente con el informe, y en caso de obtener una nota inferior a 4 en alguno de las dos calificaciones, se evaluará con la menor nota.
- En caso que no se entregue alguno de los dos, se evaluará con la nota mínima.

Este laboratorio debe ser entregado el día 1 de abril del año 2017, hasta las 23:59 hrs. Los descuentos por atraso corren a contar de las 00:59 hrs del día 2 de abril del año 2017

En caso de dudas o problemas en el desarrollo, comunicarse con su ayudante de laboratorio.

CAPÍTULO 3. ANEXOS

Camino de datos de ejemplo: