

ORGANIZACIÓN DE COMPUTADORES

LABORATORIO 1

Profesores: Alfonso Guzmán & Daniel
Wladdimiro

Ayudantes: Mario Álvarez, Sebastián Pinto-
Agüero & Ariel Undurraga

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 instrucción específica. 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

En el presente laboratorio, se deberán leer dos archivos de entrada, el primero contiene las instrucciones de un programa de MIPS, mientras que el segundo, contiene las líneas de control de cada una de las instrucciones del programa. Dado los dos archivos de entrada, usted debe generar un archivo de salida donde entregue la traza de la ejecución del código propuesto en el primer archivo de entrada, indicando si las líneas de control que usted obtiene son las mismas que las líneas de control del segundo archivo, y en caso de existir un error, indicar cual de los bits es el erróneo.

Por ejemplo, se tiene este primer archivo con el siguiente código:

```
addi $t1, $zero, 2
```

Y el segundo archivo de entrada con la siguiente línea de control correspondiente a la anterior instrucción:

RegDst	Jump	Branch	MemRead	MemToReg	ALUOp	MemWrite	ALUSrc	RegWrite
0	0	1	0	0	00	0	1	0

Cuadro 2.1: Entrada del segundo archivo

La salida del programa va a ser la ejecución del código, indicando si existe un error en las líneas de control del segundo archivo, como se aprecia a continuación:

Estado	RegDst	Jump	Branch	MemRead	MemToReg	ALUOp	MemWrite	ALUSrc	RegWrite
Error	-	-	1	-	-	-	-	1	-

Cuadro 2.2: Salida del archivo

Otro ejemplo, con el archivo de entrada de instrucciones correspondiente a:

```
add $t1, $zero, $zero
beq $t1, $zero, END
addi $t1, $t1, 4
END:
```

Y con segundo archivo de entrada:

En este archivo de salida, se puede observar que existen dos líneas solamente, dado que la instrucción `add` y `beq` son las únicas instrucciones que se ejecutan, debido que la condición se cumple y salta a la etiqueta `END`:

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.

RegDst	Jump	Branch	MemRead	MemToReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	0	0	0	0	10	0	0	1
1	0	0	0	0	01	0	1	0
1	1	1	0	0	00	1	0	0

Cuadro 2.3: Entrada del segundo archivo ejemplo dos

Estado	RegDst	Jump	Branch	MemRead	MemToReg	ALUOp	MemWrite	ALUSrc	RegWrite
Correcto	-	-	-	-	-	-	-	-	-
Error	-	-	0	-	-	-	-	1	-

Cuadro 2.4: Salida del archivo segundo ejemplo

Para efectos de la evaluación, se subirán archivos de prueba para que puedan probar su programa, los cuales no necesariamente van a ser utilizados para la revisión del laboratorio

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 índice, 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 08 de Septiembre de 2017, hasta las 23:59 hrs. Los descuentos por atraso corren a contar de las 00:59 hrs del día 09 de septiembre 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:

