

ORGANIZACIÓN DE COMPUTADORES

LABORATORIO 1

Profesores: Néstor González & Leonel Medina

Ayudantes: Matías Fuentes & Ricardo Álvarez

CAPÍTULO 1. CONTEXTO

En el curso de organización de computadores podemos ver que una de las cosas esenciales en estos es el camino de datos ó (*datapath*), el cual consiste en un conjunto de unidades funcionales para la ejecución de las diferentes instrucciones que se les da a un computador, es decir, sin este conjunto de unidades no existirían los computadores como los conocemos hoy en día.

En específico, el camino de datos representa el camino que recorren las diferentes señales eléctricas a través de las distintas unidades funcionales, esto, con el fin de llevar a cabo una instrucción básica. Un ejemplo de estas son las del lenguaje MIPS, las que encienden un conjunto de unidades del camino de datos y con esto llevar a cabo la instrucción deseada mas estos no siempre tienen un correcto funcionamiento.

Ante esto surge la siguiente interrogante: ¿Qué unidades funcionales se activan al ejecutar un determinado conjunto de instrucciones?

CAPÍTULO 2. INSTRUCCIONES

Usted debe leer dos archivos de entrada, el primero contendrá instrucciones de un programa MIPS y el segundo contendrá una lista de las líneas de control, indicando con un valor 0 después del nombre de la línea si esta presenta el problema stuck at 0, o con una x si no tiene problema (para efectos de este laboratorio las líneas que pueden presentar problema son Jump, Branch, Regwrite, Memread y Memwrite). El programa debe ser capaz de generar 2 archivos de salida, donde el primero entrega la traza de ejecución del programa MIPS ejecutado con el posible error, y el segundo debe entregar los valores de registros después de 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 los archivos de entrada sean:

Archivo Entrada 1

```
Jump end
subi $t2, $zero, 2
subi $t3, $zero, 3
sub $t1, $t2, $t3
sub $t3, $t2, $t1
end:
```

Archivo Entrada 2

```
Regdst x
Jump 0
Branch x
Memread x
Memtoreg x
Aluop x
Memwrite x
Alusrc x
Regwrite x
```

Como podemos observar en el segundo archivo de entrada, podemos dar cuenta que la línea Jump se encuentra con problema de Stuck-at 0, por lo tanto todas las operaciones se ejecutaran en el camino de datos con la línea Jump en 0.

Archivo Salida 1

```

Jump end
subi $t2, $zero, 2
subi $t3, $zero, 3
sub $t1, $t2, $t3
sub $t3, $t2, $t1

```

Archivo Salida 2

Instrucción	\$t0	\$t1	\$t2	\$t3
Jump end	0	0	0	0
subi \$t2, \$zero, 2	0	0	-2	0
subi \$t3, \$zero, 3	0	0	-2	-3
sub \$t1, \$t2, \$t3	0	1	-2	-3
sub \$t3, \$t2, \$t1	0	1	-2	-3

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.

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 (estándar ANSI C).
- El programa y el informe deben ser entregados como una carpeta comprimida en formato `zip`.
- 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 a la nota final.
- 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.
- Utilizar la plantilla de LaTeX disponible en el Moodle del curso.
- Consultar a los ayudantes y en Moodle del curso.

2.3 DESCUENTOS

- Si el programa no compila sera evaluado con nota mínima.
- Por cada exigencia no cumplida, se descontarán dos décimas a la nota, a excepción las que ya mencionan su descuento.
- Por cada dos falta 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 23 de abril del año 2018, hasta las 23:59 hrs, La no entrega o entrega tardía se evaluará con la nota mínima. Bajo ningún concepto se postergara la entrega de este laboratorio.
- En caso de dudas o problemas en el desarrollo, comunicarse con su ayudante de laboratorio.

CAPÍTULO 3. ANEXOS

Camino de datos de ejemplo:

