

Arquitectura de Computadores

Laboratorio 1

30 de Marzo, 2023

Resultados de aprendizaje

1. Usar MARS (un *IDE* para MIPS) para escribir, ensamblar y depurar programas MIPS.
2. Escribir programas MIPS incluyendo instrucciones aritméticas, de salto y de acceso a memoria.
3. Realizar llamadas de sistema en MIPS mediante uso de `syscall` empleando procedimientos.
4. Implementar algoritmos en MIPS para resolver problemas matemáticos sencillos.

1 Aritmética de antaño

Hace algunas décadas, un equipo de arqueólogos de Chile descubrió una tumba antigua que contenía una serie de inscripciones extrañas. Después de varios meses de investigación, descubrieron que estas inscripciones eran en realidad números representados en un formato especial, y que los antiguos habitantes de la región los utilizaban para realizar cálculos aritméticos.

Intrigados por este descubrimiento, los arqueólogos comenzaron a estudiar más a fondo estas inscripciones y descubrieron que contenían una serie de problemas aritméticos, incluyendo sumas, restas, multiplicaciones y divisiones. Al parecer, los antiguos habitantes de la región utilizaban estas operaciones aritméticas en su vida diaria, ya sea para llevar un registro de sus bienes o para realizar cálculos para sus negocios.

Los arqueólogos se dieron cuenta de que si querían entender mejor esta civilización antigua, necesitaban ser capaces de realizar estas operaciones aritméticas utilizando el formato empleado por dicha civilización. Es por ello que se le solicitó a los estudiantes de la asignatura de Arquitectura de Computadores, que desarrollaran un programa en MIPS que permita realizar las 4 operaciones aritméticas básicas (suma, resta, multiplicación y división), con el formato específico que utilizaba la civilización antigua. Sin embargo, el equipo con el que trabajan los arqueólogos está un poco obsoleto, por lo tanto, el desarrollo del programa debe regirse por ciertas limitaciones.

1.1 Formato de números

Los arqueólogos, luego de bastante tiempo, lograron entender el formato utilizado por la civilización que estaban investigando. Este formato en particular separa cada número en sus dígitos, pero estos se ordenan de forma inversa. Es decir:

$$a = 4321 = [1, 2, 3, 4]$$

$$b = 654 = [4, 5, 6]$$

1.2 Operaciones aritméticas

1.2.1 Suma

Considerando los números a y b mostrados anteriormente, la operación de suma quedaría de la siguiente manera.

$$a + b = 4975 = [5, 7, 9, 4] \quad (1)$$

1.2.2 Resta

Utilizando los mismos a y b , la resta sería:

$$a - b = 3667 = [7, 6, 6, 3] \quad (2)$$

1.2.3 Multiplicación

Para la multiplicación de a y b quedaría:

$$a \cdot b = 2825934 = [4, 3, 9, 5, 2, 8, 2] \quad (3)$$

1.2.4 División

Finalmente la división de a y b :

$$a \div b = 6,6070 = \begin{cases} [6] & \text{parte entera} \\ [0, 7, 0, 6] & \text{parte decimal} \end{cases} \quad (4)$$

1.3 Implementación

Dada la particularidad del formato utilizado por los antiguos pobladores, la implementación a realizar en MIPS debe seguir el flujo de la figura 1.

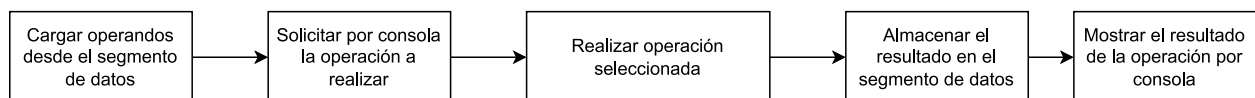


Figure 1: Flujo de proceso del programa

El programa debe extraer los operandos a utilizar en cada operación aritmética del segmento de datos de MIPS. Estos estarán almacenados con el mismo formato utilizado por la civilización antigua. Por ejemplo, el número 321 estará almacenado a partir de la dirección 0x10010000. Es decir:

Table 1: Tabla ejemplo de segmento de datos en MARS

Address	Value(+0)	Value(+4)	Value(+8)	Value(+c)	Value(+10)	Value(+14)	Value(+18)	Value(+1c)
0x10010000	0x00000001	0x00000002	0x00000003	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Por ende, cuando se realice la carga de los operandos desde el segmento de datos, éstos se deben almacenar temporalmente en registros para su posterior operación.

Luego de realizar la operación pertinente el resultado debe ser almacenado en el segmento de datos en una dirección específica, la cual será detallada más adelante.

Cabe tener en cuenta que en el segmento de datos no se encontrará el signo de los operandos. Por ende, éstos se encontrarán en unos registros en particular que se detallarán más adelante. Para denotar el signo del número se utilizará el 1 como indicador de que el número es negativo y un 0 en el caso de que el número sea positivo. Por ejemplo, si el registro `$t1 = 1` se utiliza para el signo del primer operando, y se toma como referencia el número de la tabla 1, el operando sería -321.

Finalmente, la ejecución del programa termina al mostrar por consola el resultado obtenido, por medio de `syscall`.

1.4 Restricciones

El programa que el estudiantado creará, deberá estar sujeto a las siguientes restricciones, con la finalidad de que el código resultante sea fácil de entender. Además, esto permitirá acotar el alcance de cómputo para cada operación. A continuación se enumeran las restricciones con las cuales se debe diseñar y escribir el código.

1. Los registros que se utilizarán para almacenar el signo de cada operando y del resultado deben ser los siguientes: `$s1`, `$s2`, `$s3`. Donde `$s1` corresponde al signo del primer operando, `$s2` para el signo del segundo operando y `$s3` para el signo del resultado.
2. Como se mencionó antes, los operandos se encuentran en el segmento de datos. Por lo tanto, las direcciones a utilizar son las siguientes: `0x100100a0` , `0x100100c0` , `0x100100e0`. Para el primer operando, segundo operando y resultado respectivamente.
3. **No se pueden utilizar** instrucciones de multiplicación, división y desplazamiento: `mul`, `mul.d`, `mul.s`, `mulo`, `mulou`, `mult`, `multu`, `mulu`, `div`, `divu`, `rem`, `sll`, `sllv`, `sra`, `srav`, `srl`, `srlv`; sino que se debe implementar una técnica de división y multiplicación basada en otras operaciones matemáticas.
4. En el caso de la división, si esta no corresponde a una división entera, la parte decimal (con un máximo de 4 dígitos) debe ser almacenada con el mismo formato que la parte entera pero en la dirección `0x10010100`.
5. Para la multiplicación y la división, se debe considerar que los operandos no tienen más de 4 dígitos.
6. Para las demás operaciones (suma y resta), los operandos deben tener un máximo de 7 dígitos.

Entrega

El/la estudiante deberá subir el código desarrollado, junto con su informe, a través de la plataforma Campus Virtual del curso. El archivo con código MIPS Assembly debe poder ensamblarse y ejecutarse en el simulador

MARS, y deben estar debidamente comentados. El código debe ser entregado con el siguiente nombre `codigo_NombreApellido.asm`.

Informe

El informe a entregar debe contar con lo siguiente:

1. Introducción que incluya el problema, solución y objetivos de esta experiencia.
2. Marco teórico que explique los conceptos necesarios para entender el trabajo desarrollado.
3. Explicación breve del desarrollo de la solución y cómo se llegó a esta.
4. Resultados del laboratorio.
5. Conclusiones.

El detalle de la rúbrica para la evaluación de esta experiencia estará disponible en la sección respectiva de Moodle.

Exigencias

1. El código MIPS a realizar, debe se debe crear a partir de los conocimientos presentados en cátedra.
2. El código será ejecutado y probado en el simulador MARS. Por lo que, si el código escrito no funciona en dicho simulador, se considerará como nota mínima el código.
3. El informe escrito debe ser entregado en formato PDF y no puede exceder 5 páginas de contenido. En caso contrario, por cada página extra se descontará 5 décimas a la nota final.
4. El informe se debe entregar con el siguiente formato de nombre `informe_NombreApellido.pdf`
5. El informe se debe desarrollar en base a la plantilla que se dejará disponible en Campus Virtual.
6. Tanto el código fuente como el informe deben ser enviados en un archivo comprimido, cuyo nombre debe incluir el RUT de el/la alumno/a (ej: lab1_12345678-9.zip)

Recomendaciones

1. Consultar al ayudante en las sesiones de laboratorio.

Descuentos

1. Por cada día de atraso se descontará un punto a la nota final del laboratorio.
2. Por cada tres faltas ortográficas o gramaticales en el informe, se descontará una décima a la nota del informe.
3. Por cada falta de formato en el informe se descontará una décima a la nota del informe.
4. Por cada página extra en el informe se descontarán cinco décimas a la nota final del informe.

Evaluación

1. La nota del laboratorio será el promedio aritmético entre la nota del informe y la nota del código fuente.
2. En caso de que no se entregue el informe y/o el laboratorio, se evaluará con la nota mínima.
3. Este laboratorio debe ser entregado el **13 de Abril** de 2023, a las 23:59 hrs por Campus Virtual.