

Laboratorio de Electrónica Digital II

Práctica No. 3: Operaciones Básicas Sobre Datos Numéricos Empleando Lenguaje Ensamblador para ARM

Profesor Teoría

Felipe Cabarcas Jaramillo. (felipe.cabarcas@udea.edu.co)

Profesores Laboratorio

Andrés Benavides Arévalo (bernardo.benavides@udea.edu.co)

John Byron Buitrago (john.buitrago@udea.edu.co)

Julio 18, 2022



Fecha de entrega:	Agosto 02-05 de 2022
Medio de entrega:	Classroom
Sustentación:	Horario de Laboratorio
Valor Práctica:	8% del curso

1 Introducción

En esta práctica de laboratorio, el grupo de estudiantes hará uso del lenguaje ensamblador para procesadores ARM de 32-bits, con el propósito de desarrollar un programa que sea capaz de realizar varias operaciones sobre datos numéricos. Los datos de entrada al programa estarán ubicados en la memoria en posiciones específicas. Una vez el programa realice sus operaciones, deberá llevar

los datos, resultado de las diferentes operaciones, a la memoria. El funcionamiento del programa se comprobará mediante el uso de un simulador apropiado.

2 Objetivo de la Práctica

Desarrollar un programa para el procesador ARM, empleando lenguaje ensamblador, que realice un conjunto de operaciones específicas sobre datos numéricos ubicados en la memoria y realizar la correcta simulación para comprobar su funcionamiento.

3 Procedimiento

Para el correcto desarrollo del programa para el procesador ARM es necesario leer completamente esta guía. Se sugiere seguir el procedimiento indicado a continuación:

- a. Comprender la arquitectura del conjunto de instrucciones del procesador ARM.
- b. Simular los programas de ejemplo disponibles en la página del curso, empleando el simulador del procesador ARM sugerido en esta guía, y comprender su funcionamiento.
- c. Elaborar los Pseudocódigos o diagramas de flujo para el programa en ensamblador a desarrollar, el cual debe llevar a cabo la funcionalidad que se indicará más adelante.
- d. Desarrollar el programa solicitado empleando lenguaje ensamblador para el procesador ARM basado en los Pseudocódigos o diagramas de flujo realizados previamente.
- e. Simular el programa desarrollado para el procesador ARM y verificar el correcto funcionamiento del mismo. Llevar a cabo las correcciones pertinentes, si fuese el caso.
- f. Enviar el código fuente de su programa en lenguaje ensamblador para el procesador ARM, junto con un breve reporte, antes de la fecha límite.
- g. Sustentar el diseño en el horario de laboratorio correspondiente.

4 Funcionamiento

El programa a desarrollar debe estar en capacidad de realizar cualquiera de las siguientes operaciones sobre un conjunto N de valores numéricos:

- (i) Ordenar datos numéricos ascendentemente.
- (ii) Ordenar datos numéricos descendientemente.
- (iii) Buscar múltiplos de M entre los datos numéricos.

Los valores N y M , así como los N valores numéricos a operar, estarán ubicados en la memoria. La operación a realizar cuando el programa se ejecute en el simulador será determinada a partir de un valor ubicado también en la memoria, denominado OP. Los rangos de valores para los datos que serán entregados al programa son los siguientes:

- (i) N: valor entero en el rango $2 \leq N \leq 50$.
- (ii) M: valor entero en el rango $2 \leq M \leq 7$.
- (iii) Datos: valores enteros en el rango $-2^{31}(-2147483648) \leq VAL \leq 2^{31} - 1(+2147483647)$.
- (iv) OP: valor entero en el rango $1 \leq OP \leq 3$.

Todos los valores antes mencionados se cargarán, junto con las instrucciones del programa, en el momento en que inicie la simulación. Durante la ejecución, el programa almacenará en memoria los valores generados por las diferentes operaciones. En la siguiente sección se dará a conocer la forma de especificar tanto los valores de entrada, como la posición de memoria donde se ubicarán los resultados.

5 Diseño y Simulación

El código a desarrollar se basará en la plantilla que se indica en la Fig. 1.

```
.global _start
.equ    MAXN,    50
.text
_start:
/* Inicio de Programa:
 * Inicialización de registros y lectura de valores requeridos desde la memoria
 */

/* Cuerpo del programa:
 * Código principal, subrutinas y llamados a subrutinas desde el código principal
 * para realizar la operación dada por OP sobre los datos en memoria
 */

/* Fin de Programa:
 * Bucle infinito para evitar la búsqueda de nuevas instrucciones
 */
finish:
b finish

.data
N:      .dc.l    12
M:      .dc.l    5
Datos:  .dc.l    1,15,-79,35,96,-564,8542,-89542,12021,54215,12,-35
OP:     .dc.l    1

DatosOut: .ds.l    MAXN
```

Fig. 1: Plantilla para el código a desarrollar

Para los datos que se leen desde la memoria tenga en cuenta que:

- (i) N se encuentra ubicado en la dirección dada por la etiqueta N.
- (ii) M se encuentra ubicado en la dirección dada por la etiqueta M.
- (iii) El primer valor de los datos se encuentra ubicado en la dirección dada por la etiqueta Datos, el segundo en Datos + 4, el tercero en Datos + 8 y así sucesivamente.
- (iv) OP se encuentra ubicado en la dirección dada por la etiqueta OP.

Por otro lado los datos, resultado de cualquier operación, se escribirán en la memoria así: el primer dato de la lista de valores generados por la operación seleccionada (ascendente, descendente, múltiplo de M) se almacenará en la dirección de memoria dada por la etiqueta DatosOut, el segundo en DatosOut + 4, el tercero en DatosOut + 8 y así sucesivamente.

Adicionalmente, tenga en cuenta que la arquitectura ARM que se está estudiando en el curso (ARMv4), así como la versión disponible en el simulador sugerido (ARMv7), no dispone de instrucciones de división (UDIV, SDIV). Por tal razón, si usted requiere realizar divisiones en su programa, deberá desarrollar una subrutina empleando otras instrucciones para realizar la división de dos números. Se recomienda emplear un algoritmo eficiente para este fin.

Tenga en cuenta que el uso de subrutinas es obligatorio. El grupo de trabajo deberá desarrollar por lo menos tres subrutinas, una para cada operación a realizar, a las cuales se les deberá pasar uno o varios argumentos para su funcionamiento.

El grupo de trabajo deberá guardar su programa en un archivo de nombre **program.yy.asm**. Para la simulación, utilice el simulador dado en las referencias (al final de este documento) y compruebe el funcionamiento del programa con sus propios valores de prueba.

6 Entrega

El grupo de trabajo deberá escribir un breve reporte (2-3 páginas) que contenga los siguientes elementos:

- a. **Abstract:** resumen del diseño del programa en ensamblador para el procesador ARM.
- b. **Pseudocódigo:** Pseudocódigo o diagrama de flujo del programa realizado en lenguaje ensamblador.
- c. **Conclusiones:** relacionadas con el objetivo de la práctica. Evite hablar sobre cuestiones generales o ya conocidas. Indique también el tiempo que le tomó realizar este desarrollo.

Cree un archivo comprimido que incluya el reporte y los archivos importantes de su programa en ensamblador como se describe a continuación:

- a. Reporte: archivo con extensión .pdf
- b. Archivos esenciales de su programa *.asm, *.txt, *.hex, etc.

El nombre del archivo comprimido debe tener el siguiente formato:

p3-primerapellidointegrante1-primerapellidointegrante2-horariolaboratorio.zip.

Ejemplo: si el primer apellido de ambos integrantes es **Benavides** y **Buitrago**, respectivamente, y el laboratorio es el Martes 9-12, entonces el archivo debe ser nombrado: *p3-benavides-buitrago-m9-12.zip*.

7 Evaluación

La evaluación de la práctica se divide en tres partes: funcionamiento (40%), sustentación (40%) y reporte (20%). La nota del funcionamiento se asigna por igual a todos los integrantes del grupo de trabajo (máximo dos personas por equipo), mientras que la nota de sustentación es individual. En caso de que un estudiante obtenga una nota inferior a 3.0 en la sustentación, la nota final de la práctica para el estudiante en mención será la que obtuvo en la sustentación, es decir, no se tendrá en cuenta el funcionamiento en el cálculo.

Cada grupo de trabajo deberá sustentar la práctica en un tiempo de 15 minutos, 10 minutos para revisar la simulación y 5 minutos para preguntas. Durante la sustentación, el profesor hará un par de preguntas a cada uno de los integrantes del grupo de trabajo.

8 Referencias

- a. Harris, Sarah and Harris, David. Digital Design and Computer Architecture: ARM Edition. Morgan Kaufmann Publishers Inc. 2015. ISBN: 0128000562
- b. CPULATOR Computer System Simulator
<https://cpulator.01xz.net/?sys=arm>