Guía de ejercicios # 2 Primeros pasos en programación-Arquitectura Q1

Organización de Computadoras 2017

UNQ

1 Introducción

Comenzaremos a partir de esta práctica a dar nuestros primeros pasos en programación en lenguaje ensamblador, el mismo posee diferente tipo de instrucciones que nos posibilitarán que nuestra computadora, realice las operaciones indicadas. Con este propósito utilizaremos una Arquitectura conceptual que llamaremos Q, la misma incorporará a lo largo de nuestra cursada, las herramientas necesarias para ir desarrollando nuestros programas en su lenguaje ensamblador.

1.1 Ejercitación:

En relación a lo conversado en clase y la bibliografía de la materia contesta las las siguientes preguntas:

a) ¿Qué es un programa?

b) ¿Cuál es el ciclo que realiza la computadora cuando ejecuta una instrucción?

2 Ensamblado y desensamblado:

¿Cómo se transforma el código máquina a código fuente y viceversa?

Proceso de ensamblado y desensamblado:

Cuando:

- Código máquina se desensambla $\longrightarrow$ se transforma en código fuente
- Código fuente se ensambla $\longrightarrow$ se transforma en código máquina

3 Arquitectura Q1:

Ahora sí te vamos a presentar como se ve nuestra Arquitectura Q en su primera versión Q1, Recordá que por ahora te presentamos estas especificaciones que irán ampliándose a medida que avancemos en la materia.

Características:

- Tiene 8 registros de uso general de 16 bits: R0..R7
- Tiene instrucciones de 2 operandos.

Instrucciones de 2 operandos

Operación	Cod Op	Efecto
MUL	0000	Dest ← Dest * Origen
MOV	0001	Dest ← Origen
ADD	0010	Dest ← Dest + Origen
SUB	0011	Dest ← Dest - Origen
DIV	0111	Dest ← Dest % Origen

Nota: El caracter % denota el cociente de la división entera. El resultado de la operación `MUL` ocupa 32 bits, almacenándose los 16 bits menos significativos en el operando destino y los 16 bits mas significativos en el registro `R7` .

Modos de direccionamiento

Sus modos de direccionamiento son:

Modo	Codificación
Inmediato	`000000`
Registro	`100rrr`

Donde `rrr` es una codificación (en 3 bits) del número de registro.

Formato de instrucción

El siguiente es el formato de las instrucciones de **Q1**, las cuales tienen dos operandos (origen y destino). Los tamaños de los campos estan expresados en bits.

Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Origen (16b)

A continuación te presentaremos conceptos que te ayudarán a comprender cada una de las partes que la componen.

Observación:

Cuando escribamos una instrucción que contenga un valor inmediato, lo expresaremos en hexadecimal, para los cual escribiremos el símbolo 0X que indica la notación hexadecimal, y luego la cadena que represente el valor. Ej MOV R5, 0x0002

4 Instrucciones:

El funcionamiento del procesador está determinado por las instrucciones que ejecuta. Cada una de ellas se representa por una secuencia de bits, denominándose **instrucciones máquina**.

Cada instrucción debe contener la información que necesita el procesador para su ejecución, dichos elementos son: Código de operación, Referencia a los operandos origen, Referencia al resultado y Referencia a la siguiente instrucción. Como ya te mencionamos, cuando presentamos la **Arquitectura Q1**, en una instrucción, luego del código de operación, se encuentra en primer término el operando **destino**(Donde dirigimos el dato) y en segundo término el **origen** (desde donde).

Por ej MOV (COD OP) R3 (DESTINO),R5 (ORIGEN).

Observación:
El operando destino no podrá contener un dato de tipo inmediato.

4.1 Ejercitación:

a) A continuación enumeramos una serie de instrucciones de la Arquitectura Q1, algunas escritas de modo correcto y otras no. Tu tarea será indicar cuales de estas instrucciones son correctas y cuáles no. En caso de ser incorrectas deberás escribir su modo correcto para que pueda ser interpretada por la computadora.

Instrucción	Correcta/Incorrecta
MOV R4, 0X0003	
ADD 0X0001, R2	
SUMAR R2, R3	
DIV R2,R3	
MULTI R2,OX0012	

5 Modos de direccionamiento:

Las instrucciones de un programa en código máquina, necesitan datos para funcionar y generan resultados que es necesario almacenar. Por ejemplo, una operación de suma necesitara conocer donde se encuentran las dos cantidades que se desean sumar y el lugar donde deberá almacenar el resultado, una vez finalizada la operación.
Existen varios mecanismos para indicar estas posiciones, y estos reciben el nombre de **modos de direccionamiento**.
En el caso de nuestra Arquitectura Q1, veremos por ahora dos:
Direccionamiento Inmediato: El operando está presente en la propia instrucción.
Direccionamiento Registro: El operando se encuentra en un registro del sistema en lugar de estar en memoria principal.

5.1 Ejercitación:

a) Deberás indicar en cada una de las instrucciones que te presentamos a continuación cuales son los modos de direccionamiento que se utilizan en los operandos destino y origen:

Instrucción	Modo Direccionamiento
ADD R1, 0XBEBE	
MOV R2, R0	
MUL R1, R2	
DIV R2,0X00AC	

6 Ejercitación integradora

Ahora que tenemos todas las herramientas necesarias, comenzaremos a realizar nuestros primeros programas.

6.1 Ejemplo integrador:

Escribimos un programa que cargue en el registro R1 el valor 3 en BSS (16):

Efecto de la instrucción:

MOV R1, 0X0003: Copia el valor 3 en el registro 1

Cod operación: 0001

Modo destino: Registro

Modo Origen: inmediato

Ensamblamos la instrucción:

0001(MOV) 100001 (REGISTRO1) 000000 (INMEDIATO) 0000000000000011 (VALOR EN BINARIO).

Observación:
Recordá que si realizamos el proceso de desensamblado, debemos obtener la instrucción original: MOV R1, OX0003

6.2 Ahora sí!

Escribimos programas cortos:

1. Escribir un programa que cargue el registro R0 con la cadena que representa el valor 3 en BSS(16).
2. Escribir un programa que cargue el registro R0 con la cadena que representa el valor 15 en BSS(16).
3. Escriba un programa que cargue el registro R1 con la cadena que representa el valor 16 en BSS(16).
4. Escriba un programa que cargue el registro R2 con la cadena que representa el valor 255 en BSS(16).
5. Escribir un programa que duplique el valor contenido en el registro R0.
6. Escribir un programa que le sume el valor 3 al contenido del registro R1.
7. Escribir un programa que multiplique por 12 el contenido del registro R0.

8. Escribir un programa que calcule el valor de la expresión 22 + 65. Nota: No debe resolver la cuenta, sino hacer un programa que lo haga.

9. Escribir un programa que sume los valores de los registros R1 y R0, y ponga el resultado en R2 (sin modicar R1 y R0).

10. Escribir un programa que a R5 le reste 2 veces el valor que tiene R6.

11. Escribir un programa que calcule el promedio entre los registros R2 y R3.

12. Escribir un programa que a R4 le sume los valores de R1, R2 y R3; y le reste los valores de R5, R6 y R7.

13. Tenemos una pequeña empresa de Software y contamos con la siguiente información a cierre de balance, acumulada en los registros de la siguiente forma:

    COSTOS FIJOS (CF) en R0, COSTOS VARIABLES (CV) en R1, PRECIO DE VENTA POR CADA PRODUCTO (Q): R4

    Con esta información resolver los siguientes planteos:

    a) Escribir un programa que calcule los costos totales de la compañía (CT=CF +CV) y los almacene en el registro R2.

    b) Si se comercializaron 300 productos, Escribir un programa que calcule los ingresos totales que representa almacenándolo en el registro R3 (Ingreso= precio de venta * cantidad)

    c) Queremos calcular la Ganancia del período (Ganancia= Ingresos – CostosTotales) para lo cual escribiremos un programa que lo realice (los datos pedidos los calculamos y guardamos en el inciso a y b). Almacenar el valor resultante en el registro R5.

    d) Completar el siguiente esquema con cada uno de los programas realizados en los incisos a y b. Completarlo como el modelo resuelto en el ejercicio integrador:

    Efecto de la instrucción:
    Código de operación:
    Modo Destino:
    Modo Origen:
    Ensamblado:

14. **Programas que calculan intensidades:**

    Te invitamos a resolver cada uno de los ejercicios y luego mostrar cómo quedarían ensamblados los programas resultantes.

    a) La intensidad de una corriente eléctrica se mide como la Tensión sobre la resistencia (I= T/R), escribir un programa que calcule dicha función matemática y la almacene en el registro R2. Sabemos que la Tensión la almacenamos en R0 Y la Resistencia en R1.

    b) La intensidad de un terremoto se mide mediante sismógrafos, estos dibujan una traza con una cierta amplitud A, cuando dicha amplitud varía indica que se produce una variación en los temblores de la superficie terrestre. La expresión es M(A)= Log A+3 donde el valor del log de A lo tenemos almacenado en el registro R0. Realizar un programa que pueda calcular la magnitud de un terremoto y la almacene en el registro R1.

15. **Desensamblamos:** Tu tarea en este ejercicio es desensamblar el código máquina, y obtener como resultado el programa en código fuente.

    <u>Observación:</u>
    Recordá que para esto debes tomar cada cadena de bits y observar la distribución del formato de instrucción de Q. Por Ej.0011000001100000 desensamblado es: MOV R1, R0

    a) 0001100000100001
    0000100001100000
    0010100001000000
    0000000000000101

    b) 0001100101000000
    1010101100100011

    c) 0010100011100110
    0001100011100000
    0011100011000000
    1101111000010001

16. **Ensamblar y ejecutar:**

    En cada una de los programas que presenta el cuadro planteado a continuación, realizaremos las tareas de ensamblado y ejecución mostrando el valor final que tiene cargado cada registro al finalizar dicha ejecución, para lo cual debemos tener en cuenta la condición supuesta en cada caso:

Instrucción	Ensamblado	Condición supuesta	Resultado luego ejecución almacenado en el/los registros
ADD R0,0xFAFF		R0 contiene el valor 0x0001	
SUB R6,0x5678		R6 contiene el valor 0x00FF	
ADD R0,R0 MUL R0,R1		Ro contiene el valor 0x0001, r1 contiene el valor ox000F	

References

[1] Williams Stallings, *Computer Organization and Architecture*, octava edición, Editorial Prentice Hall, 2010. Capitulos 10 y 11

[2] Representacion de Instrucciones (apunte de la materia), `http://orga.blog.unq.edu.ar/wp-content/uploads/sites/5/2015/04/orga_clase2_apunteRrepresentacionInstrucciones.pdf`

[3] Modos de direccionamientos (apunte de la materia) `http://orga.blog.unq.edu.ar/wp-content/uploads/sites/5/2015/04/orga_ModosDeDireccionamiento.pdf`.