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

Organización de Computadoras 2018

UNQ

Objetivos

Que el estudiante pueda:

- Entender qué es programar y cuál es el ciclo de vida de un programa
- Escribir un programa utilizando los elementos de Q1
- Relacionar la ejecución de los programas con los circuitos

Para resolver esta práctica se aconseja consultar el apunte de la materia *Ejecución de programas y representacion de instrucciones*, disponible en `http://orga.blog.unq.edu.ar/descargas/`.

Los ejercicios marcados con ♣ forman un conjunto minimal para aprender e integrar los conceptos. Los demás ejercicios son redundantes y permiten seguir entrenando.

Introducción

Comenzaremos a partir de esta práctica a dar los primeros pasos en la programación. Haremos hincapíe en que para que la computadora pueda llevar a cabo la ejecución de las instrucciones, es necesario que pueda decodificar las mismas en algún código binario.

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.

Relacionaremos a su vez las diferentes etapas que intervienen en el ciclo de vida de un programa, relacionando los conceptos abordados en las prácticas anteriores.

Te recomendamos resolverla apoyado en las clases presenciales y el apunte del tema que se encuentra en el blog de la materia.

Ejercitación:

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

1. ¿Qué es un programa?

2. Explica que sucede con la computadora internamente, cuando tengo un programa que deba sumar dos valores contenidos en b y c respectivamente. ¿Qué unidades,circuito/s,etc participan?

3. Enuncia y explica cuáles son los pasos que forman parte del ciclo de vida de un programa.

4. ¿Cuándo finaliza un programa?

1 Instrucciones:

Un conjunto de instrucciones conformarán un programa y cada instrucción a su vez, deberá contener la información que necesita el procesador para su ejecución. Dichos elementos son: Código de operación, Referencia a los operandos, Referencia al resultado y Referencia a la siguiente instrucción.

Instrucciones de 2 operandos

La Arquitectura Q, presenta las siguientes instrucciones de 2 operandos Destino y Origen.

Operación	Efecto
MUL	Dest ← Dest * Origen
MOV	Dest ← Origen
ADD	Dest ← Dest + Origen
SUB	Dest ← Dest - Origen
DIV	Dest ← Dest % Origen

En todas estas instrucciones se asume que luego de la operación se indica el **operando destino**(es un operando y también en donde se almacenará el resultado) y luego el **operando origen**.

Por ejemplo, en la instrucción `ADD R3, R5`, se sumará el contenido del registro R3 al contenido de R5 y el resultado se almacenará en R3 (operando destino).

Ejercitación:

5. 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	

6. Explicá donde se guarda la Instrucción de un programa en la etapa de Búsqueda de Instrucción.

2 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 **registro e inmediato.**

Los campos Modo Destino y Modo Origen responden a la siguiente codificación de modos de direccionamiento:

Modo	Codificación
Inmediato	`000000`
Registro	`100rrr`

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

Ejercitación:

7. Indicar en cada una de las Siguientes instrucciones cuales son los modos de direccionamiento que se utilizan en los operandos destino y origen:

Instrucción	Modo Destino	Modo Origen
`ADD R1, 0XBEBE`		
`MOV R2, R0`		
`MUL R1, R2`		
`DIV R2,0X00AC`		

8. Indicar ahora que conocés cuál es el efecto de aplicar cada una de las intrucciones del cuadro anterior:

Instrucción	Efecto de la instrucción
`ADD R1, 0XBEBE`	
`MOV R2, R0`	
`MUL R1, R2`	
`DIV R2,0X00AC`	

3 Programación en Q1

Con las herramientas adquiridas deberás resolver los programas que se te piden a continuación:

9. Escribir un programa que cargue el registro R0 con la cadena que representa el valor 3.

10. Escribir un programa que cargue el registro R0 con la cadena que representa el valor 15.

11. Escriba un programa que cargue el registro R1 con la cadena que representa el valor 16.

12. Escriba un programa que cargue el registro R2 con la cadena que representa el valor 255.

13. Escribir un programa que duplique el valor contenido en el registro R0.
14. Escribir un programa que le sume el valor 3 al contenido del registro R1.
15. Escribir un programa que multiplique por 12 el contenido del registro R0.
16. 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.
17. Escribir un programa que sume los valores de los registros R1 y R0, y ponga el resultado en R2 (sin modificar R1 y R0).
18. Escribir un programa que a R5 le reste 2 veces el valor que tiene R6.
19. Escribir un programa que calcule el promedio entre los registros R2 y R3.
20. 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.
21. 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 (P): 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= P * cantidad)
    (c) Queremos calcular la Ganancia del período (Ganancia= Ingreso – CT) 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.

4 Ensamblado (Código máquina)

El proceso de ensamblado es la traducción del código fuente (programas escritos en lenguaje legible para el ser humano) a código máquina (la codificación que entiende la computadora).

Formato de instrucción

Las cadenas de bits de los campos que representan las instrucciones de **Q1**, las cuales tienen dos operandos (origen y destino) es el siguiente:

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

El campo **codop** es un código que identifica la operación, siguiendo la siguiente tabla:

Operación	Cod Op
MUL	0000
MOV	0001
ADD	0010
SUB	0011
DIV	0111

En algunos ejercicios usaremos un cuadro como el que sigue para explicar como se ensamblan las instrucciones. Ej: Si consideramos un programa que cargue en el registro R1 el valor 3 en $BSS(16)$, entonces el cuadro se completará de la siguiente forma:

Instrucción	Efecto	Codop	M.Dest	M.Origen
`MOV R1,0x0003`	R1 ← 3	0001	100001	0000000

El resultado de **ensamblar** esta instrucción a código máquina se ve así:
`00011000010000000000000000000011`

Ejercitación

22. 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 (estado inicial) en cada caso:

Instr.	Código Máquina	Estado inicial	Estado final
`ADD R0,0xFAFF`		R0=0x0001	
`SUB R6,0x5678`		R6=0x00FF	
`ADD R0,R0` `MUL R0,R1`		R0=0x0001, R1=0x000F	

5 Desensamblamos (Código Fuente)

El proceso de desensamblado es el pasaje del código máquina (la codificación que entiende la computadora) a código fuente (programas escritos en lenguaje legible para el ser humano).

Ejercitación

23. 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 es el codigo máquina de: MOV R1, R0

    a 0001100000100001
    0000100001100000
    0010100001000000
    0000000000000101

    b 0001100101000000
    1010101100100011

    c 0010100011100110
    0001100011100000
    0011100011000000
    1101111000010001

6 Ejercicios adicionales de programación:

Te invitamos a resolver cada uno de los programas que se presentan a continuación y luego completar para cada uno el siguente cuadro:

Instrucción	Código Máquina	Estado inicial	Estado final

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.

References

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