Trabajo práctico N° 5

El software

Fecha de finalización: 19 de Junio

Introducción a la computación
Departamento de Ingeniería de Computadoras
Facultad de Informática - Universidad Nacional del Comahue

Objetivo: Comprender la organización y el funcionamiento básico de una computadora simple. Se involucran conocimientos de los componentes hardware y sus interacciones para ejecutar instrucciones.

Recursos bibliográfico:

- *Andrew S. Tanenbaum.* Organización de computadoras: un enfoque estructurado. Cuarta edición, editorial Pearson Educación, 2000. ISBN 970-170-399-5.

Lectura obligatoria:

- Apuntes de cátedra. Capitulo 5: Arquitectura y Organización de Computadoras. Disponible en *PEDCO*: `https://pedco.uncoma.edu.ar/mod/url/view.php?id=203642`
- Apuntes de cátedra. Capitulo 7: El Software. Disponible en *PEDCO*: `https://pedco.uncoma.edu.ar/mod/url/view.php?id=203642`

1. Modelo Computacional Binario Elemental (MCBE)

1. Para los programas que se describen a continuación:

   a) Completar la columna de "Contenido binario" con la codificación en *Lenguaje Máquina* de la instrucción.

   b) Realice una traza del programa e intente identificar que tarea realiza el programa. Busque una explicación de alto nivel (por ejemplo: *"el programa pide ingresar un número y lo multiplica por 3"*, o *"el programa imprime los números del 5 al 1"*).

1)

#	Rotulo	Mnemónico	Argumento	Contenido Binario
0		LD	IN	0101 1110
1		SUB	DATO	
2		SUB	DATO	
3		SUB	DATO	
4		ST	OUT	
5		HLT		0010 0000
6	DATO:	2		0000 0010

2)

#	Rotulo	Mnemónico	Argumento	Contenido Binario
0	SALTO2:	LD	DATO	0100 1001
1		JZ	SALTO1	
2		LD	DATO2	0100 1011
3		ST	OUT	0111 1111
4		LD	DATO	
5		SUB	UNO	
6		ST	DATO	0110 1001
7		JMP	SALTO2	
8	SALTO1:	HLT		0010 0000
9	DATO:	5		0000 0101
10	UNO:	1		0000 0001
11	DATO2:	8		0000 1000

2. Para los programas que se describen a continuación:

   a) Completar la columna de "Mnemónico" con la codificación en *Lenguaje de Ensamblador* de la instrucción.

   b) Realice una traza del programa e intente identificar que tarea realiza el programa. Busque una explicación de alto nivel (por ejemplo: *"el programa pide ingresar un número y lo multiplica por 3"*, o *"el programa imprime los números del 5 al 1"*).

1)

#	Contenido Binario	Rotulo	Mnemónico	Argumento
0	0101 1110			IN
1	0110 0110			DATO
2	0101 1110			IN
3	1000 0110			DATO
4	0111 1111			OUT
5	0010 0000			
6	0000 0000	DATO:		

2)

#	Contenido Binario	Rotulo	Mnemónico	Argumento
0	0101 1110			IN
1	1010 0110			DATO1
2	0111 1111			OUT
3	1000 0111			DATO2
4	0010 0000			
5	0000 0100	DATO1:		
6	0000 1001	DATO2:		

3)

#	Contenido Binario	Rotulo	Mnemónico	Argumento
0	0101 1110			IN
1	0110 1001			DATO
2	1000 1001			DATO
3	1000 1001			DATO
4	1000 1001			DATO
5	1000 1001			DATO
6	1010 1010			DATO2
7	0111 1111			OUT
8	0010 0000			
9	0000 0000	DATO:		
10	0000 0011	DATO2:		

3. Dado el siguiente programa:

#	Rotulo	Mnemónico	Argumento	Contenido Binario
0		LD	IN	0101 1110
1	SIGUE:	JZ	FIN	1110 0100
2		ST	OUT	0111 1111
3		SUB	UNO	1010 0110
4		JMP	SIGUE	1101 1101
5	FIN:	HLP		0010 0000
6	UNO:	1		0000 0001

a) ¿Qué tarea realiza el programa?

b) ¿Qué modificación seria necesaria realizarle al código ensamblador para que el programa imprima cada número dos veces? ¿Se modifican la misma cantidad de celdas en el programa en código maquina?

4. Dado el siguiente programa:

#	Rotulo	Mnemónico	Argumento
0		LD	IN
1		ST	NUM
2	SIGUE:	LD	CANT
3		JZ	IMPRIME
4		SUB	UNO
5		ST	CANT
6		LD	NUM
7		SUB	DOS
8		ST	NUM
9		JMP	SIGUE
10	IMPRIME:	LD	NUM
11		ST	OUT
12	FIN:	HLT	
13	TEST:	-64	
14	NUM:	0	
15	CANT:	5	
16	UNO:	1	
17	DOS:	2	

a) ¿Qué tarea realiza el programa?

b) ¿Cual sera el comportamiento del programa si se cambia la instrucción en la dirección nueve por "**JMP TEST**"?

5. Analice y responda las siguientes preguntas, justificando su respuesta:

a) ¿Cuál es la dirección de la primera instrucción que ejecutará la máquina?

b) ¿Cuál es el rango de valores del acumulador del *MCBE*?

c) ¿A qué distancia máxima puede "saltar" el control del programa?

d) ¿Cuál es la dirección más alta donde puede encontrarse una instrucción a ser ejecutada?

e) ¿Puede el *MCBE* encontrar una instrucción que no sea capaz de decodificar?

f) ¿Qué utilidad tendría un programa que no hiciera ningún uso de la Unidad de *Entrada/Salida* para comunicar sus resultados?

g) Supongamos que hemos almacenado en la posición 14 un dato numérico que representa la edad de una persona. ¿Qué pasa si en algún momento de la ejecución el *PC* contiene el número 14? ¿Qué pasará si esa persona tiene 33 años? ¿Qué pasará si tiene 65?

h) ¿Qué pasa si el programa no contiene una instrucción *HLT*?

i) Un programa ¿Puede modificarse a sí mismo? ¿Esto es útil? ¿Conveniente? ¿Peligroso?

j) ¿Podría aumentarse la capacidad de memoria de el *MCBE*? ¿Esto requeriría algún cambio adicional a la máquina?

k) ¿Cómo se podría aumentar la cantidad de instrucciones diferentes de el *MCBE*? ¿Esto tendría algún efecto sobre la longitud de los programas que puede ejecutar la máquina?

l) Si una nueva arquitectura tiene instrucciones con 5 bits para el código de operación y 10 para operando (donde en el caso de las instrucciones de carga y almacenamiento indican la dirección de la celda origen o destino) ¿Cuantas operaciones distintas puede tener como máximo esta arquitectura? ¿Cuantas celdas de memoria puede tener?

m) Se desea establecer el tamaño mínimo para una arquitectura que tiene 14 instrucciones distintas, y cada instrucción tiene tres operandos, donde cada operando referencia una de las 512 celdas de memoria ¿Cuál es la cantidad mínima de bits que puede tener una instrucción?

Anexo

Descripción del Modelo Computacional Binario Elemental (MCBE)

Memoria: consta de 32 posiciones de 8 bits. Las direcciones 0 a 29 corresponden a direcciones que pueden ser escritas y leídas. La dirección 30 es de **sólo lectura**, permite leer datos del dispositivo de entrada, por ejemplo un teclado. La dirección 31 es de **sólo escritura**, permite escribir datos en el dispositivo de salida, por ejemplo en una pantalla o una impresora.

Registro PC: registro de 8 bits, contiene la dirección de la próxima instrucción a ejecutar. Se inicializa en cero.

Registro IR: registro 8 bits donde se guarda la instrucción que se esta decodificando o ejecutando.

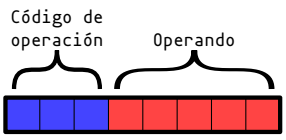
Registro acumulador: registro de 8 bits donde se almacena un número entero representado en *complemento a 2*.

Etiquetas predefinidas:

IN: dirección 30, entrada, dirección de solo lectura.

OUT: dirección 31, salida, dirección de solo escritura.

Instrucciones: de 8 bits, los 3 bits más significativos almacenan el código de operación, y los 5 menos significativos almacenan el operando.



Mnemónico	**Código de operación** *3 bits*	**Operando** *5 bits*	**Descripción**
`LD`	010	*dirección*	**Memoria → Acumulador**. Copia un byte desde la dirección de memoria al acumulador.
`ST`	011	*dirección*	**Acumulador → Memoria**. Copia el contenido del acumulador en esa dirección de memoria.
`ADD`	100	*dirección*	**Suma**. El contenido de la dirección se suma al acumulador, y el resultado se almacena en el acumulador.
`SUB`	101	*dirección*	**Resta**. El contenido de la dirección se resta al acumulador, y el resultado se almacena en el acumulador.
`JMP`	110	*desplazamiento*	**Salto incondicional**. Se suma (en complemento a 2) el desplazamiento al **PC**.
`JZ`	111	*desplazamiento*	**Salto condicional**. Si el acumulador es cero, se suma (en complemento a 2) el desplazamiento al **PC**, en caso contrario el **PC** se incrementa en uno.
`HLT`	001	*(sin uso)*	**Detiene la maquina**. No se ejecutan nuevas instrucciones. Los registros y la memoria quedan con el último valor que tenían.
`NOP`	000	*(sin uso)*	**No operación**. No tiene ningún efecto sobre el acumulador ni memoria. El **PC** se incremente en uno.