

Guía de ejercicios # 6

Organización de Computadoras 2013

UNQ

Arquitectura Q4

Características

- Tiene 8 registros de uso general de 16 bits: R0..R7.
- La memoria utiliza direcciones son de 16 bits.
- Tiene un contador de programa (*Program counter*) de 16 bits.
- *Stack Pointer* de 16 bits. Comienza en la dirección FFEF.
- Flags: Z, N, C, V (Zero, Negative, Carry, oVerflow).
- Todas las instrucciones alteran los Flags excepto MOV, CALL, RET, JMP, Jxx, PUSH y POP.
- De las que alteran los Flags, todas dejan C y V en 0 a excepción de ADD, SUB y CMP.

Instrucciones de dos operandos

Formato de Instrucción				
CodOp (4b)	Modo Destino (6b)	Modo Origen (6b)	Destino (16b)	Origen (16b)
Tabla de códigos de operaciones				
Operación	Cod Op	Efecto		
MUL	0000	Dest \leftarrow Dest * Origen ¹		
MOV	0001	Dest \leftarrow Origen		
ADD	0010	Dest \leftarrow Dest + Origen		
SUB	0011	Dest \leftarrow Dest - Origen		
CMP	0110	Dest - Origen		
DIV	0111	Dest \leftarrow Dest % Origen ²		

Instrucciones de un operando origen

Formato de Instrucción			
CodOp (4b)	Relleno (000000)	Modo Origen (6b)	Operando Origen (16b)
Tabla de códigos de operaciones			
Operación	Cod Op	Efecto	
CALL	1011	[SP] \leftarrow PC; SP \leftarrow SP - 1; PC \leftarrow Origen	
JMP	1010	PC \leftarrow Origen	

Instrucciones sin operandos

Formato de Instrucción		
CodOp (4b)	Relleno (00000000000000)	
Tabla de códigos de operaciones		
Operación	CodOp	Efecto
RET	1100	SP \leftarrow SP + 1; PC \leftarrow [SP]

Salto condicionales

$\text{Cod_Op (8)} \mid \text{Desplazamiento(8)}$ donde los primeros cuatro bits del campo **Cod_Op** corresponden a la cadena 1111_2 . Las instrucciones en este formato son de la forma Jxx(salto relativo condicional). Si al evaluar la condición de salto en los Flags el resultado es 1, el efecto es incrementar al PC con el valor de los 8 bits de desplazamiento, representado en complemento a 2 de 8 bits. En caso contrario la instrucción no hace nada.

Codop	Op.	Descripción	Condición de Salto
0001	JE	Igual / Cero	Z
1001	JNE	No igual	not Z
0010	JLE	Menor o igual	Z or (N xor V)
1010	JG	Mayor	not (Z or (N xor V))
0011	JL	Menor	N xor V
1011	JGE	Mayor o igual	not (N xor V)
0100	JLEU	Menor o igual sin signo	C or Z
1100	JGU	Mayor sin signo	not (C or Z)
0101	JCS	Carry / Menor sin signo	C
0110	JNEG	Negativo	N
0111	JVS	Overflow	V

Modos de direccionamiento

Modo	Codificación
Inmediato	000000
Directo	001000
Registro	100rrrr ³

1. Flags y saltos

1.1 Diseñar un circuito que calcule el **flag Z** a partir de una suma en *BSS*(4). Considerar que se tiene disponible un sumador de 4 bits.

1.2 Diseñar un circuito que calcule el **flag N** a partir de una suma en *BSS*(4). Considerar que se tiene disponible un sumador de 4 bits.

1.3 Diseñar un circuito que calcule el **flag V** a partir de una suma en *BSS*(4). Considerar que se tiene disponible un sumador de 4 bits.

1.4 Diseñar un circuito que calcule el **flag V** a partir de una resta en *BSS*(4). Considerar que se tiene disponible un restador de 4 bits.

1.5 Dar los valores de R3 y R4 (y calcular los flags de la primer instrucción) que hagan que se ejecute la instrucción *CALL*:

```
rutina: CMP R3, R4
        JLE fin
        CALL boom
fin: RET
```

1.6 Dar los valores de R3 y R4 (y calcular los flags de la primer instrucción) que hagan que se ejecute la instrucción *CALL*:

```
rutina: CMP R3, R4
        JLEU fin
        CALL boom
fin: RET
```

1.7 Dar los valores de R3 y R4 (y calcular los flags de la primer instrucción) que hagan que se ejecute la instrucción *CALL*:

```
rutina: CMP R3, R4
        JL fin
        CALL boom
fin: RET
```

1.8 Dar los valores de R3 y R4 (y calcular los flags de la primer instrucción) que hagan que se ejecute la instrucción *CALL*:

```
rutina: CMP R3, R4
        JCS fin
        CALL boom
fin: RET
```

2. Ensamblado de saltos

2.1 Considere el siguiente programa.

```
SUB R0, 0x1
JE pisar
MOV R3, [0A0A]
pisar: MOV R3, 0xFFFF
```

- a) Ensamblar a a partir de la celda FFOE.
- b) ¿Que valor tiene el desplazamiento del salto *JE*?
- c) ¿A que celda queda asociada la etiqueta *pisar*?
- d) Explique que hace el programa

2.2 Considere el siguiente programa.

```
SUB R0, 0x1
JE escero
ADD R0, 0xB
SUB R0, 0xA
JLEU mayoradiez
escero: MOV R3, 0xFFFF
mayoradiez: MOV R3, 0xFFFF
```

- a) Ensamblar a a partir de la celda FFOE.
- b) ¿Que valor tiene el desplazamiento del salto *JE*?
- c) ¿Que valor tiene el desplazamiento del salto *JLEU*?
- d) ¿A que celda queda asociada la etiqueta *escero*?
- e) ¿A que celda queda asociada la etiqueta *mayoradiez*?
- f) Explique que hace el programa

2.3 Dado el siguiente programa:

```
MOV [R0], R2
ADD R0, [1C00]
JE alla
SUB R3, 0x0001
alla: MOV R1, [1B00]
```

- a) Ensamblar a a partir de la celda 2001h.
- b) ¿Que valor tiene el desplazamiento del salto *JE*?
- c) ¿A que celda queda asociada la etiqueta *alla*?

2.4 Dado el siguiente mapa de memoria, simule la ejecución de un programa que comienza en la celda A893, asumiendo que R0 = 0000 y R1 = F0000

	...
A893	6821
A894	FC04
A895	1980
A896	FFFF
A897	A000
A898	A89B
A899	1980
A89A	AAAA
A89B	C000
	...

2.5 Simule la ejecución del programa en 2.4, asumiendo que R0 = F000 y R1 = 0000

- 2.6 Simule la ejecución del programa en 2.4, reemplazando el valor de la celda A894 por el valor FA04 y asumiendo que R0 = 0000 y R1 = F0000.
- 2.7 Dado el siguiente mapa de memoria, simule la ejecución de un programa que comienza en la celda A89A, asumiendo que el valor de R0 es 2.

	...
A893	3800
A894	0001
A895	F102
A896	A000
A897	A893
A898	C000
A899	B000
A89A	A893
	...

3. Estructura condicional

Nota: Documente todas las rutinas y programas.

- 3.1 Escribir un programa que, si el valor en R0 es igual al valor en R1, ponga en R2 un 1 ó 0 en caso contrario
- 3.2 Escribir un programa que, si el valor en R7 es negativo, le sume 1, o le reste 1 en caso contrario
- 3.3 Escribir un programa que ponga en R2 el máximo valor entre R0 y R1. Considerar que los valores están en *BSS()*
- 3.4 Escribir un programa que ponga en R2 el máximo valor entre R0 y R1. Considerar que los valores están en *CA2()*
- 3.5 Escribir un programa **absBSS** que determina si el valor almacenado en el registro R7 es negativo (en BSS). En caso de serlo, lo reemplaza por su valor absoluto.
- 3.6 Escribir un programa **absCA2** que determina si el valor almacenado en el registro R7 es negativo (en CA2). En caso de serlo, lo reemplaza por su valor absoluto.
- 3.7 Escribir un programa que utilice la rutina **avg** de la práctica número 5 para determinar si el promedio de los valores almacenado en las celdas 1000 y 1004 es mayor al valor 1010, en dicho caso ponga 1 en R1
- 3.8 Escribir un programa que utilice la rutina **sumaDos** de la práctica número 5 y determine si el valor resultado es mayor al contenido de R4, en ese caso poner un 1 en la celda 00FE
- 3.9 Escribir un programa que, si el valor en R1 es 0 ponga en R7 el valor almacenado en la celda 0FFE. En caso contrario guarde en R7 la suma de R2 y R3
- 3.10 Escribir una rutina **init** que, si el valor almacenado en R1 es FFFF, inicialice con 0 el resto de los registros.