

Guía de ejercicios # 8

Organización de Computadoras 2014

UNQ

Arquitectura Q6

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). Instrucciones que alteran Z y N: ADD, SUB, CMP, DIV, MUL, AND, OR, NOT. Las 3 primeras además calculan C y V.

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	$\text{Dest} \leftarrow \text{Dest} * \text{Origen}^1$
MOV	0001	$\text{Dest} \leftarrow \text{Origen}$
ADD	0010	$\text{Dest} \leftarrow \text{Dest} + \text{Origen}$
SUB	0011	$\text{Dest} \leftarrow \text{Dest} - \text{Origen}$
AND	0100	$\text{Dest} \leftarrow \text{Dest} \vee \text{Origen}$
OR	0101	$\text{Dest} \leftarrow \text{Dest} \wedge \text{Origen}$
CMP	0110	$\text{Dest} - \text{Origen}$
DIV	0111	$\text{Dest} \leftarrow \text{Dest} \% \text{Origen}^2$

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	$[\text{SP}] \leftarrow \text{PC}; \text{SP} \leftarrow \text{SP} - 1;$ $\text{PC} \leftarrow \text{Origen}$
JMP	1010	$\text{PC} \leftarrow \text{Origen}$

Instrucciones de un operando destino

Formato de Instrucción			
CodOp (4b)	Modo Origen (6b)	Relleno (000000)	Operando Origen (16b)

Tabla de códigos de operaciones		
Operación	Cod Op	Efecto
NOT	1001	$\text{Dest} \leftarrow \neg \text{Dest}$

Instrucciones sin operandos

Formato de Instrucción	
CodOp (4b)	Relleno (00000000000000)

Tabla de códigos de operaciones		
Operación	CodOp	Efecto
RET	1100	$\text{SP} \leftarrow \text{SP} + 1; \text{PC} \leftarrow [\text{SP}]$

Salto condicionales

$\boxed{\text{Cod_Op (8)}} \mid \boxed{\text{Desplazamiento(8)}}$ donde los primeros cuatro bits del campo **Cod_Op** es la cadena 1111₂. Si **al evaluar la condición de salto** el resultado es 1, se le suma al PC el valor del desplazamiento, representado en C42(8). 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
Indirecto	011000
Registro	100rrr ³
Registro	110rrr ⁴

1. Máscaras

1. ¿Para que utilizamos las máscaras?

2. ¿Cuál es el resultado de las siguientes operaciones?

$$a) \quad \begin{array}{r} \text{AND} \quad 1101 \\ \quad \quad 0111 \\ \hline \quad \quad ? \end{array}$$

$$b) \quad \begin{array}{r} \text{OR} \quad 0101 \\ \quad \quad 1001 \\ \hline \quad \quad ? \end{array}$$

$$c) \quad \begin{array}{r} \text{NOT} \quad 0100 \\ \quad \quad ? \end{array}$$

$$d) \quad \begin{array}{r} \text{XOR} \quad 1011 \\ \quad \quad 1110 \\ \hline \quad \quad ? \end{array}$$

$$e) \quad \begin{array}{r} \text{AND} \quad 1010 \\ \quad \quad 1100 \\ \hline \quad \quad ? \\ \text{OR} \quad 0101 \\ \quad \quad ? \\ \hline \quad \quad ? \\ \text{XOR} \quad 1100 \\ \quad \quad ? \\ \hline \quad \quad ? \end{array}$$

3. Complete con el operador adecuado (AND, OR, XOR, NOT) en las siguientes operaciones

$$a) \quad 1000 \dots\dots 1011 = 1011$$

$$b) \quad 1011 \dots\dots 1000 = 1000$$

$$c) \quad 1101 \dots\dots 1001 = 0100$$

$$d) \quad 1111 \dots\dots 0011 = 0011$$

$$e) \quad \dots\dots 0011 = 1100$$

4. Dado una cadena de bits formada por $(X_7, X_6, X_5, X_4, X_3, X_2, X_1, X_0)$ ¿qué resultado obtendrá al aplicarle la siguiente operación lógica?

$$a) \quad X \text{ OR } 10101010$$

$$b) \quad X \text{ OR } 11111000$$

$$c) \quad X \text{ AND } 10101010$$

$$d) \quad X \text{ AND } 10001111$$

$$e) \quad X \text{ XOR } 10101010$$

$$f) \quad X \text{ XOR } 00001111$$

$$g) \quad X \text{ OR } 10000000, \text{ al resultado AND } 11110000, \text{ y al resultado XOR } 00011110$$

$$h) \quad X \text{ AND } 10101111, \text{ al resultado OR } 11110000, \text{ y al resultado XOR } 00011110$$

$$i) \quad X \text{ XOR } 10101010, \text{ al resultado AND } 11110000, \text{ y al resultado OR } 00011110$$

5. Escriba un programa que calcule la operación lógica **nand** entre los valores almacenados en los registros R6 y R7 y deja el resultado en el registro R7.

6. Escriba un programa que calcule la operación lógica **nor** entre los valores almacenados en los registros R6 y R7 y deja el resultado en el registro R7.

7. Escriba un programa que calcule la operación lógica **xor** entre los valores almacenados en los registros R6 y R7 y deja el resultado en el registro R7.

8. Escribir un programa que dada una cadena de 16 bits en R0, ponga un 0 en el primer bit, dejando el resto sin modificar. Utilice máscaras y operaciones lógicas.

9. Escribir un programa que dada una cadena de 16 bits en R1, invierta el valor de las posiciones impares. Utilice máscaras y operaciones lógicas.

10. Escribir un programa que dada una cadena de 16 bits en R1, ponga un 1 en el segundo bit, dejando el resto sin modificar. Utilice máscaras y operaciones lógicas.

11. Escriba un programa que ponga en 0 el sexto bit de R1.

12. Escriba un programa que ponga en 1 el cuarto bit de R1.

13. Escriba un programa invierta los 4 últimos bits de R1.

14. Escriba un programa que obtenga los últimos 4 bits de R1 y guarde ese valor en la posición 999A de memoria.

15. Escriba un programa calcule el opuesto aditivo del número almacenado en el registro R2. Dicho número está representado en C' A2(16).

16. Escriba un programa que determine si el contenido de R2 y R3 son iguales, usando operaciones lógicas

17. Escriba un programa que guarde en la posición FF00 de memoria el contenido de R0, solo si su bit menos significativo es 0.

18. Escriba un programa que copie los 4 bits menos significativos de R0 en R1, y complete los restantes bits con 0.

19. Escriba un programa que copie los 4 bits menos significativos de R0 en R1, y complete los restantes bits con 1.

20. Escriba un programa que copie el byte mas significativo de la celda 0348 en el registro R1

21. Escribir una rutina **desplazarIzq** que desplace los bits de la cadena contenida en R0 un lugar hacia la izquierda, guardando el resultado en R1.
22. Escribir una rutina **desplazarDer** que desplace los bits de la cadena contenida en R0 un lugar hacia la izquierda, guardando el resultado en R1.
23. Escriba un programa que cuente la cantidad de dígitos 1 en la cadena que está en R6
24. Escribir un programa que cuente cuantos números de un arreglo son pares y cuántos tienen el bit 5 en 1. El arreglo comienza en la celda 4486 y la longitud del arreglo está en la celda 4485.
25. Documente el programa y las subrutinas que definió en el ejercicio anterior.
26. En una fábrica de ventanas las características de los productos de cada venta se codifican mediante cadenas de 16 bits. Cada bit representa una cualidad y se coloca un 1 si cumple con dicha característica, por ejemplo:

- Con el **bit 3** se indica si está pintada
- Con el **bit 9** se indica si lleva el vidrio de seguridad

(Nota: Tener en cuenta que los demás bits también representan distintas características, pero para el ejercicio solo nos interesan esos dos).

En caso que la ventana esté pintada o lleve vidrio de seguridad es necesario usar un embalaje distinto, y para esto se pide escribir un programa que determine colocando un 1 en R7 si el pedido que está en la celda 0019 requiere un embalaje *premium*, es decir, si se trata de una ventana pintada o con vidrio de seguridad.

27. Se tiene un arreglo de pedidos a partir de la celda 0001, cuya longitud está en la celda 0000. Escriba un programa que recorra el arreglo para contar en R6 los pedidos que **no necesitan embalaje premium según el ejercicio anterior**.
28. En un arreglo se tienen codificados los pedidos de una rotisería en cadenas de 16 bits, donde el byte mas significativo representa el tipo de producto y el byte menos significativo la cantidad de unidades. Los códigos de producto son:
 - Si bit 15 (mas significativo) es 1: empanadas de carne
 - Si bit 14 es 1: empanadas de pollo
 - Si bit 13 es 1: empanadas de jamón y queso
 - Si bit 12 es 1: pizza napolitana

Por ejemplo:

Cadena	Interpretación
1000000 00000110	6 empanadas de carne
0010000 00001010	10 empanadas de jamón y queso

Se necesita calcular el trabajo de los cocineros, totalizando la cantidad de empanadas de cada tipo. Escriba las siguientes subrutinas:

contarEmpCarne Cuenta en R4 la cantidad pedida de empanadas de carne

contarEmpPollo Cuenta en R5 la cantidad pedida de empanadas de pollo

contarEmpJyQ Cuenta en R6 la cantidad pedida de empanadas de jamón y queso

contarPizzaNapo Cuenta en R5 la cantidad pedida de pizzas napolitanas

Considere que el arreglo de pedidos comienza en la celda 5310, y termina con el primer valor 0.

29. Considerando la rotisería descrita en el ejercicio anterior, escriba un programa que calcule la ganancia a partir del arreglo de pedidos y un arreglo de precios unitarios que ocupa el rango de celdas 5200..5203 (pues son 4 productos).

2. Sistemas de Punto Fijo

30. Suponer un sistema de punto fijo $BSS(10, 4)^5$. Interprete las cadenas:

- a. 0100000000
- b. 0101010101
- c. 1000000000
- d. 1111111000
- e. 1111111111
- f. 1010101010
- g. 0111111111
- h. 0110011000

31. Interprete las cadenas del ejercicio anterior en un sistema $BSS(10, 3)$

32. Interprete las cadenas del ejercicio anterior en un sistema $SM(10, 3)^6$.

33. Suponer un sistema $BSS(10, 4)$.

- a) ¿Cuántos números se pueden representar?
- b) ¿Cuál es la resolución del sistema?
- c) ¿Cuáles son el máximo y el mínimo número representables?
- d) ¿Cuáles son el máximo y el mínimo número representable en el intervalo $(0,1)^7$?

⁵Sistema de punto fijo en *Binario Sin Signo* con 10 bits en total, 4 de los cuales son fraccionarios

⁶Sistema de punto fijo en *Signo-Magnitud* con 6 bits de parte entera y 1 bit de signo, 3 bits de parte fraccionaria

⁷Intervalo desde el 0 hasta el 1, **ambos excluidos**

34. Responder las preguntas anteriores para un sistema $BSS(8, 3)$.
35. Suponer un sistema $BSS(10, 4)$. Represente los siguientes números:
- a) 1,2
 - b) 1,25
 - c) 35
 - d) 1,0625
 - e) 13,763
 - f) 1,4
43. Escribir una rutina que **redondeo** que obtenga el valor entero mas cercano al valor almacenado en R1, este último se encuentra representado en $BSS(16, 3)$. Guarde el resultado en R2.
44. Escribir un programa que compare los valores almacenados en R1 y R2, guardando 1 en R6 en caso de que R1 sea mayor a R2. Los valores se encuentran representados en $BSS(16, 8)$.

Si alguno no se puede representar, justifique. Calcule el error absoluto y relativo en cada caso.

36. Represente los siguientes números en $SM(10, 4)$.

- a) 24,0
- b) 1,25
- c) -15.25
- d) 1,0625
- e) -13,763
- f) 1,4

Si alguno no se puede representar, justifique. Calcule el error absoluto y relativo en cada caso.

37. Suponer un sistema $BSS(4, 1)$. ¿Cuál es el máximo error absoluto que puede ocurrir al representar un valor dentro del rango?
38. Suponer un sistema $BSS(4, 2)$. ¿Cuál es el máximo error absoluto que puede ocurrir al representar un valor dentro del rango?
39. Suponer un sistema $BSS(4, 3)$. ¿Cuál es el máximo error absoluto que puede ocurrir al representar un valor dentro del rango?
40. Suponer un sistema $BSS(4, 4)$. ¿Cuál es el máximo error absoluto que puede ocurrir al representar un valor dentro del rango?
41. Considere el siguiente arreglo que contiene valores codificados en $BSS(16, 8)$:

	...
3000	FF00
3001	087E
3002	8990
	...

¿Cuántos de estos valores son mayores a 0?

42. Escribir una rutina que **ptoFijoAEntero** que obtenga la parte entera del valor almacenado en R1, dicho valor se encuentra representado en $BSS(16, 8)$. Guarde el resultado en R2.