

Guía de ejercicios # 7

Estructura Condicional, Flags y Saltos

Organización de Computadoras 2019

UNQ

Los objetivos de esta práctica son:

- Comprender qué son y para qué se utilizan los Flags
- Conocer qué operaciones modifican los Flags y cómo los modifican
- Comprender los conceptos de saltos absolutos o condicionales
- Introducir conceptos de iteración o bucles en programas
- Notar la relación entre invocación a rutinas y saltos

Para resolver esta práctica se aconseja consultar el apunte de la materia *Flags y Saltos*, disponible en <http://orga.blog.unq.edu.ar/descargas/>

1 Flags y Saltos

Ejemplo de cálculo de Flags:

Considerando la operación de suma $1001 + 1110$ en $BSS(4)$, es posible ver:

$$\begin{array}{r} + 1001 \\ \text{acarreo} \quad 1110 \\ \hline 0111 \end{array}$$

Los valores de los flags luego de la operación, son los siguientes:

- **Z (Zero)** = 0. Ya que el es distinto de 0000.
- **N (Negative)** = 0. El resultado comienza con 0.
- **C (Carry)** = 1. La operación tiene acarreo.
- **V (Overflow)** = 1. Ambos operandos representarían números negativos (a pesar de que estemos trabajando en $BSS()$), por lo que la suma, debiera ser también un número negativo, y en ese caso no lo es.

1. Realizar las siguientes operaciones en $BSS(4)$ y calcular los flags a partir de los resultados de las mismas.

(a) $1010 + 1001$

(b) $1011 - 1011$

(c) $0010 + 1101$

(d) $0010 - 0111$

(e) $1100 - 1000$

2. Dar los valores de R3 y R4 (calculando los flags de la primer instrucción) que evite que se llame a la rutina "boom":

(a)

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

(b)

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

(c)

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

(d)

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

3. 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.

4. 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.

5. 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.

6. 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.

7. Indique verdadero o falso. Justifique.

- (a) La instrucción JMP no modifica el SP.
- (b) Los flags se modifican con todas las instrucciones de dos operandos.
- (c) La instrucción JE es un salto incondicional.

2 Ensamblado de saltos

8. Considere el siguiente programa.

```
rutina: MOV R3, [0x0A0A]
        SUB R0, 0x0001
        JE fin
        MOV R3, 0xFFFF
fin: RET
```

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

9. Considere el siguiente programa.

```
rutina: CMP R0, 0x0000
        JL menoracero
        CMP R0, 0x000A
        JG mayoradiez
        MOV R3, 0x0001
        JMP fin
menoracero: MOV R3, 0x0000
        JMP fin
mayoradiez: MOV R3, 0x0002
        fin: RET
```

- (a) Ensamblar a a partir de la celda FOCA.
- (b) ¿Que valor tiene el desplazamiento del salto JL?
- (c) ¿Que valor tiene el desplazamiento del salto JG?
- (d) ¿A que celda queda asociada la etiqueta menoracero?
- (e) ¿A que celda queda asociada la etiqueta mayoradiez?
- (f) ¿A que celda queda asociada la etiqueta fin?
- (g) Explique que hace el programa

10. Dado el siguiente programa:

```
rutina: MOV R1, 0x0000
        CMP [0x00FF], 0x0000
        JNEG negativo
        JMP fin
negativo: MOV R1, 0x0001
        fin: RET
```

- (a) Ensamblar a a partir de la celda 2017.
- (b) ¿Que valor tiene el desplazamiento del salto JNEG?

- (c) ¿A que celda queda asociada la etiqueta negativo?
- (d) ¿A que celda queda asociada la etiqueta fin?
- (e) Explique que hace el programa

11. Dado el siguiente mapa de memoria, simule la ejecución de un programa que comienza en la celda A893, asumiendo que R0 = 0000 y R1 = F000

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

- 12. Simule la ejecución del programa en 11, asumiendo que R0 = F000 y R1 = 0000
- 13. Simule la ejecución del programa en 11, reemplazando el valor de la celda A894 por el valor FA04 y asumiendo que R0 = 0000 y R1 = F000.
- 14. Dado el siguiente mapa de memoria, simule la ejecución de un programa que comienza en la celda A899, 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

- 15. Escribir un programa que, si el valor en R0 es igual al valor en R1, ponga un 1 en R2, 0 en caso contrario.
- 16. Escribir un programa que, si el valor en R7 es negativo, le sume 1, o le reste 1 en caso contrario.
- 17. Escribir un programa que, si la suma entre R3 y R4 es menor a 512, guarde el resultado en la celda 1000, sino en la celda 2000.
- 18. Escribir un programa que, si el valor en R1 es 0 guarde el valor almacenado en la celda 0FFE en R7, en caso contrario que guarde la suma entre R2 y R3.
- 19. Dada la documentacion de la rutina avg:

```
; REQUIERE: Valores en R1 y R2.
; MODIFICA: R1
; RETORNA: En R3 el promedio entre R1 y R2.
```

Escribir un programa que determine si el promedio de los valores almacenados en las celdas 1000 y 1004 es mayor al valor 1010, en dicho caso ponga 1 en R1 o un 0 en caso contrario.

20. Implementar las siguientes rutinas según su documentación asumiendo que los valores que manejan están en CA2(16):

(a) ; ----- max
; REQUIERE: Valores a comparar en R0 y R1.
; MODIFICA: ???
; RETORNA : En R5 el valor maximo entre
; R0 y R1.

(b) ; ----- min
; REQUIERE: Valores a comparar en R0 y R1.
; MODIFICA: ???
; RETORNA : En R5 el valor mínimo entre
; R0 y R1.

(c) ; ----- multiplo
; REQUIERE: Valores en R1 y R2.
; MODIFICA: ???
; RETORNA : Un 1 en R0 si el numero que esta
; en R2 es múltiplo de R1, un 0 en
; caso contrario.

(d) ; ----- negativo
; REQUIERE: Un valor en R1.
; MODIFICA: ???
; RETORNA : Un 1 en R0 si el valor de R1 es
; negativo, 0 en caso contrario.

21. Usando la rutina `multiplo`, hacer un una rutina `esPar` que dado un numero en R1, retorne en R0 un 1 si el número de R1 es par, un 0 en caso contrario. Documente la rutina.

22. Usando la rutina `min`, hacer un programa que le sume R3 el valor más chico entre lo que está guardado en la celdas CAFE y 1882.

23. Escribir una rutina `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. Documentar la rutina.

24. Escribir una rutina `añoBiciesto` que dado un año en R0, retorne en R1 un 1 si el año es bisiestro, 0 en caso contrario. Un año es bisiestro si el número es múltiplo de 4 pero no múltiplo de 100 o sí es múltiplo de 100 y también de 400.

4 Ciclo de ejecución y accesos

25. Suponer que la instrucción `CMP [AAAA],R1` está ensamblada a partir de la celda 0000.

- (a) ¿Qué celdas se acceden durante la búsqueda de instrucción?
- (b) ¿Qué celdas se acceden durante la búsqueda de operandos?
- (c) ¿Qué celdas se acceden durante la almacenamiento de operandos?

26. Suponer que la instrucción `JE esIgual` está ensamblada a partir de la celda 0000

- (a) ¿Qué celdas se acceden durante la búsqueda de instrucción?
- (b) ¿Qué celdas se acceden durante la búsqueda de operandos?
- (c) ¿Qué celdas se acceden durante la almacenamiento de operandos?

27. Suponer que la instrucción `JMP salir` está ensamblada a partir de la celda 0000

- (a) ¿Qué celdas se acceden durante la búsqueda de instrucción?
- (b) ¿Qué celdas se acceden durante la búsqueda de operandos?
- (c) ¿Qué celdas se acceden durante la almacenamiento de operandos?