

Guía de ejercicios # 8

Estructura Condicional, Flags y Saltos

Ejercicios de programación

1. Completar las siguientes oraciones según corresponda:

- (a) Para accionar en base a una condición, primero se debe utilizar la instrucción y luego indicar algún
- (b) Los saltos condicionales, en bajo nivel, se clasifican como saltos
- (c) Los saltos el de PC.
- (d) La instrucción JMP se utiliza para “saltar” a una
- (e) La instrucción CMP compara los operandos por medio de una
- (f) Los saltos a diferencia del CMP no alteran la
- (g) En la ejecución de la instrucción CMP se calculan los
- (h) Para saber si una rutina funciona correctamente se defina una de

2. Se cuenta con la siguiente documentación de la rutina promedio (ya programada):

promedio	
Requiere	en R0 y R1 dos valores en BSS(16)
Retorna	en R3 el promedio entre ambos valores
Modifica	R1

Documentar y definir la rutina promedioMayorADiez que retorne un 1 en R1 si el promedio entre los valores de las celdas con dirección 1000 y 1004 es mayor a 10, o un 0 en caso contrario.

- 3. A partir de la rutina promedio, documentar y definir una rutina que calcule el promedio entre los valores de las celdas con dirección ABBA y ACDC si R6 es mayor a R5; en caso contrario, el promedio debe ser entre los valores de las celdas con dirección BABA y 50DA.
- 4. Definir la rutina minSinSigno para que cumpla con la siguiente documentación:

minSinSigno	
Requiere	en R3 y R4 dos valores en BSS(16)
Retorna	en R0 el valor mínimo de ambos. De ser iguales, retorna el primero.
Modifica	COMPLETAR

5. ¿Qué modificaciones se deben realizar, tanto en la documentación como en la rutina, si los valores se encuentran en el sistema CA2(16)?

6. En base a la rutina modificada (del punto anterior) que opera en CA2(16), documentar y definir una rutina que sume al valor contenido en R3, el valor mínimo entre los valores de las celdas con dirección 1976 y 1982.

7. Definir la rutina múltiplo a partir de la siguiente documentación:

múltiplo	
Requiere	en R0 y R1 dos valores en CA2(16)
Retorna	en R5 un 1 si el valor de R0 es múltiplo del valor de R1, y un 0 en caso contrario.
Modifica	COMPLETAR

8. En base a la rutina múltiplo, documentar y definir la rutina esPar, que dado un valor en R1, retorne un 1 en la celda con dirección 9A11 si dicho valor es par, o un 0 en caso contrario.

9. Documentar y definir la rutina esNegativo para que cumpla con la siguiente documentación:

esNegativo	
Requiere	en R4 un valor en CA2(16)
Retorna	en R0 un 1 si el valor dado es negativo, o un 0 en caso contrario.
Modifica	COMPLETAR

Prueba de programas

- 10. Documentar y definir una rutina de test para probar la rutina definida en el punto 6.
- 11. Documentar y definir una rutina de test para probar la rutina múltiplo definida en el punto 7.
- 12. Documentar y definir una rutina de test para probar la rutina minR1aR7 (ya programada) que cumpla con la siguiente documentación:

maxR1aR7	
Requiere	en R1 hasta R7 valores en CA2(16)
Retorna	en R0 el máximo valor de todos
Modifica	[7E57] y [1810]

Ejercicios de bajo nivel (Flags)

13. Realizar las siguientes operaciones en *CA2(4)* y calcular los flags a partir de sus resultados:

- (a) 1010 + 1001
- (b) 1011 - 1011
- (c) 0101 + 0100
- (d) 0010 - 0111
- (e) 0011 - 1000

14. Para cada rutina, asignar valores a R3 y R4 (calculando los flags correspondientes), de manera de evitar la invocación a la rutina “boom”:

- (a) `explosionMenorIgual: CMP R3, R4`
`JLE fin`
`CALL boom`
`fin: RET`
- (b) `explosionMayor: CMP R3, R4`
`JG fin`
`CALL boom`
`fin: RET`
- (c) `explosionMayorSS: CMP R3, R4`
`JGU fin`
`CALL boom`
`fin: RET`
- (d) `explosionMayorIgual: CMP R3, R4`
`JGE fin`
`CALL boom`
`fin: RET`

15. Considerar la siguiente rutina:

```
ejemplo: CMP R0, 0x0001
          JE fin
          MOV R3, [0x0A0A]
          fin: RET
```

- (a) Ensamblar y alojarla en memoria a partir de la celda con dirección *CAFE*.
- (b) ¿Qué valor tiene el desplazamiento del salto *JE*?
- (c) ¿A qué celda queda asociada la etiqueta *fin*?

16. A partir del siguiente mapa de memoria, simular la ejecución de la rutina enumerando las instrucciones en el orden en que se van ejecutando e indicando su efecto. Para ello, asumir que:

PC=A893, R0=0000, R1=F000, y SP=FFEE.

	...
A893	6821
A894	FC04
A895	1980
A896	FFFF
A897	A000
A898	A89B
A899	1980
A89A	AAAA
A89B	C000
	...
FFEF	0500
	...

17. Para cada instrucción dada, responder a qué celdas se acceden durante las etapas de búsqueda de instrucción (BI), búsqueda de operandos (BO) y almacenamiento de resultado (AR).

Asumir que cada instrucción está ensamblada a partir de la celda con dirección 0000.

- `CMP [AAAA],R1`
- `JE esIgual`, donde el valor de su desplazamiento es 4.
- `JMP salir`, donde su etiqueta está ensamblada en la celda con dirección 9006.

18. 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.