

Guía de ejercicios # 4

Memoria, Buses, Q2 y Accesos

Organización de Computadoras 2016

UNQ

Arquitectura Q3

Características

- Tiene 8 registros de uso general de 16 bits: R0..R7.
- La memoria utiliza direcciones de 16 bits.
- Tiene un contador de programa (*Program counter*) de 16 bits.
- *Stack Pointer* de 16 bits. Comienza en la dirección 0xFFEF.

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 ← Dest * Origen ¹
MOV	0001	Dest ← Origen
ADD	0010	Dest ← Dest + Origen
SUB	0011	Dest ← Dest - Origen
DIV	0111	Dest ← 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] ← PC; SP ← SP - 1; PC ← Origen

Instrucciones sin operandos

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

Tabla de códigos de operaciones		
Operación	CodOp	Efecto
RET	1100	SP ← SP + 1; PC ← [SP]

¹El resultado de la operación MUL ocupa 32 bits, almacenándose los 16 bits menos significativos en el operando destino y los 16 bits mas significativos en el registro R7

²El caracter % denota el cociente de la división entera

Modos de direccionamiento

Modo	Codificación
Inmediato	000000
Directo	001000
Registro	100rrr ³

Ejercicios

1. ¿Que relación existe entre los registros IR y PC?
2. Explique detalladamente como funciona el CALL y el RET.
3. Dado un programa que quiere invocar a una rutina:
 - (a) ¿Quién debe hacer el call: el programa o la rutina?
 - (b) ¿Quién debe hacer el ret: el programa o la rutina?
4. Escriba una rutina `mulPorDos`, que multiplique por 2 el contenido de R1 y guarde el resultado en R1

Documente la rutina especificando:

- ¿Qué requiere? (dónde recibe los parámetros y cuáles son sus precondiciones)
 - ¿Que modifica? (qué variables auxiliares utiliza)
 - ¿Qué retorna? (dónde devuelve el resultado)
5. Utilice la rutina anterior para escribir un **programa** que calcule 2 elevado a la 5.
 6. Escriba una rutina `mulPorCinco`, que multiplique por 5 el contenido de R1 y guarde el resultado en R1 . **Documente la rutina.**
 7. Utilice la rutina anterior para escribir un **programa** que calcule 5 elevado a la 3.
 8. Escriba una rutina `mulPorSiete`, que multiplique por 7 el contenido de R1 y guarde el resultado en R1 . **Documente la rutina.**
 9. Utilice la rutina anterior para escribir un **programa** que calcule 7 elevado a la 2.
 10. Escriba una rutina `mulPorCuatro`, que multiplique por 4 el contenido de R1 y guarde el resultado en R1 . **Documente la rutina.**
 11. Utilice la rutina anterior para escribir un **programa** que calcule 4 elevado a la 4.

³rrr es una codificación (en 3 bits) del número de registro

12. Sabiendo que la rutina `mulPorCuatro` se encuentran ensamblada a partir de la celda `0x2000`, el programa anterior está ensamblado a partir de la celda `0xF000`, y que la pila está vacía, simule los cambios que ocurren en el PC, en el SP y en el contenido de la pila durante la ejecución del mismo.

13. Escriba una rutina `aLaQuinta`, que eleve a la 5ta potencia el contenido del registro `R0`. **Documente la rutina.**

14. Utilizando la rutina del ejercicio anterior, escriba un programa que calcule n^5 para los números que están en las celdas `0x1000`, ..., `0x1007`.

15. Escriba una rutina `esPar`, que determine si el contenido de `R0` es par o impar. Si es par, debe guardar un 0 en `R1`, en caso contrario, debe guardar un 1 en dicho registro. **Documente la rutina.**

16. Utilizando la rutina del ejercicio anterior, escriba un programa que grabe un 0 o un 1 en `0x0000`, ..., `0x0005` según si los números de `0xF000`, ..., `0xF005` son pares o no.

17. Escribir una rutina `swapR0R1` que intercambie los valores de los registros `R0` y `R1`. Documente su rutina.

18. Escribir un programa que utilice la rutina anterior para intercambiar los valores de las celdas consecutivas entre la `0x3000` y la `0x3009`. Esto es: intercambiar el valor de `x` con el valor de `x+1`, siendo `x` una celda cuya dirección es par, en el rango `[0x3000, 0x3009]`.

19. Escribir una rutina `avg` que calcule el promedio entre `R1` y `R2`, guardando el calculo en `R3`, Documente su rutina

20. Sabiendo que en `R1`, `R2`, `R4` y `R5` se encuentran almacenadas las edades de los profes de Orga, calcular el promedio total utilizando la rutina anterior.

21. Sabiendo que las rutinas `avg` se encuentran ensamblada a partir de la celda `0x2000`, el programa anterior está ensamblado a partir de la celda `0xF000`, y que la pila está vacía, simule los cambios que ocurren en el PC, en el SP y en el contenido de la pila durante la ejecución del programa.

22. Considere la siguiente rutina:

```
rutina1: MOV R1, R0
         RET
```

y el siguiente programa:

```
programa: CALL rutina1
          CALL rutina1
```

Sabiendo que:

- `rutina1` está ensamblada a partir de la celda `0x00E0`

- el programa está ensamblado a partir de la celda `0x1000`
- `PC=1000`
- la pila está vacía⁴

simule los cambios que ocurren en el PC, en el SP y en el contenido de la pila durante la ejecución del programa.

23. Considere las siguientes rutinas:

```
rutina1: MOV R1, R0
         RET
```

```
rutina2: RET
```

y el siguiente programa:

```
programa: CALL rutina1
          CALL rutina2
```

Sabiendo que:

- `rutina1` está ensamblada a partir de la celda `0x00E0`
- `rutina2` está ensamblada a partir de la celda `0x00A1`
- el programa está ensamblado a partir de la celda `0x1000`
- `PC=0x1000`
- la pila está vacía

simule los cambios que ocurren en el PC, en el SP y en el contenido de la pila durante la ejecución del programa.

24. Considere las siguientes rutinas:

```
rutina1: MOV R0, R1
         CALL rutina2
         RET
```

```
rutina2: MOV R2, R3
         RET
```

y el siguiente programa:

```
principal: CALL rutina2
           CALL rutina1
```

Sabiendo que:

- `rutina1` está ensamblada a partir de la celda `0x00F0`
- `rutina2` está ensamblada a partir de la celda `0x00A1`
- el programa está ensamblado a partir de la celda `0x0E00`
- `PC=0x0E00`
- la pila está vacía

⁴Es decir, el valor de SP es `0xFFEF`

¿cómo varía la pila durante la ejecución del programa principal?

25. Considere las siguientes rutinas:

```
rutina1: MOV R1, R0
         CALL rutina2
         RET
```

```
rutina2: CALL rutina3
         RET
```

```
rutina3: MOV R2, R1
         RET
```

y el siguiente programa:

```
programa: CALL rutina1
          CALL rutina2
          CALL rutina3
```

Sabiendo que:

- rutina1 está ensamblada a partir de la celda 0x00E0
- rutina2 está ensamblada a partir de la celda 0x00A1
- rutina3 está ensamblada a partir de la celda 0x0101
- el programa está ensamblado a partir de la celda 0x1000
- PC=0x1000
- la pila está vacía

simule los cambios que ocurren en el PC, en el SP y en el contenido de la pila durante la ejecución del programa.

26. Escribir una rutina `sumaDos` que dados dos valores almacenados en R1 y R2 sume su contenido.

27. Se cuenta con la siguiente documentación de la rutina `aplicarDescuento`:

```
; REQUIERE: El precio unitario en la celda 0xA000
;           El porcentaje a aplicar
;           en la celda 0xA001
; MODIFICA: R0
; RETORNA: El precio con el descuento
;           aplicado en R0.
```

Utilizando las rutinas `aplicarDescuento` y `sumaDos`, escriba un programa que calcule el precio final a pagar por una persona que compra dos productos, cuyos precios unitarios están almacenados en R6 y R7 y se le debe aplicar un descuento del 10%.

28. Dada la siguiente secuencia de números 2, 4, 6, 8, 10 escribir un programa que calcule la sumatoria utilizando la rutina `sumaDos`.

29. Se cuenta con la subrutina `maxInt` que calcula el máximo entre los valores `BSS(16)` de R6 y R7, dejando el resultado en R6.

Escribir un programa que calcule el máximo de los registros R1 al R7.

30. Ensamblar el siguiente programa a partir de la celda 0xFF0E

```
SUB R0, 0x0001
CALL swap
MOV R3, [0x0A0A]
MOV R3, 0xFFFF
```

sabiendo que `swap` se encuentra ensamblado a partir de la celda 0x1000. Compruebe sus respuestas utilizando QSim.

31. Utilizando QSim, ejecute paso a paso el siguiente programa, observando el comportamiento de la pila y el registro SP. Indique con qué valor queda el registro R2 al finalizar la ejecución de la cuarta instrucción (`MOV R2, R0`):

```
MOV R0, 0x000A
MOV R1, 0x000B
CALL sumamult
MOV R2, R0
```

```
sumamult:
  ADD R0, 0x000C
  ADD R1, 0x000C
  CALL mult
  RET
```

```
mult:
  MUL R0, R1
  RET
```

32. En el ejercicio anterior, ¿cómo se traducen las etiquetas `sumamult` y `mult`?

33. Pruebe en QSim, qué efecto tiene utilizar `CALL` con modos de direccionamiento registro o directo en lugar de una etiqueta, por ejemplo:

```
MOV [0x000A], 0x001A
CALL [0x000A]
```

```
MOV R0, 0x000A
CALL R0
```

34. Utilizando QSim, pruebe el siguiente programa:

```
MOV R0, 0x0001
MOV R1, 0x0002
CALL factInfinito
```

```
factInfinito:
  MUL R0, R1
  ADD R1, 0x0001
  CALL factInfinito
```

Luego responda las siguientes preguntas:

- ¿Qué se está calculando en el registro R0?
- ¿Qué defecto tiene el programa?
- ¿Cómo crece la pila? ¿Qué valores se apilan?
- ¿Dada la arquitectura de 16 bits de QSim, hasta qué punto tiene sentido que el programa continúe calculando?