

Organización del Computador

Práctica - Assembler - Sumas y restas

1. Escriba un código para obtener en r0 la suma de los números contenidos en r1, r2, r3 y r4. Con ayuda de gdb ejecute paso por paso y verifique cómo se actualizan los valores contenidos en los registros.
2. Teniendo en cuenta las siguientes definiciones
N1: .word 18
N2: .word 33
N2: .word 40
escriba fragmentos de código para:
 - a. obtener en r0 el resultado de la operación $N1 + N2 - N3$.
 - b. hacer el mismo cálculo, almacenando el resultado en N1.
 - c. intercambiar los valores asignando a N1 el valor de N2, a N2 el de N3 y a N3 el de N1.
3. Escriba un fragmento de código assembler que usando registros r0 y r1:
 - a. sume dos valores positivos tal que se produzca un overflow
 - b. sume dos valores negativos tal que se produzca un overflow
 - c. agregar un salto condicional para detectar el overflow y almacenar en el registro r2 si hubo
o no overflow.
4. Escriba un programa en assembler para cada uno de los siguientes problemas
 - a. La etiqueta `VECTOR1` define 10 posiciones en memoria. Obtener en r0 la suma de esas 10 posiciones.
 - b. Sumar en r1 los elementos definidos en `VECTOR2` con la siguiente directiva
`vector2: .word 128, 314, 1024, 127, 16000, 65000`
Tener en cuenta que cada elemento ocupa cuatro bytes (word)
5. Escriba una sola instrucción que encienda el flag Z.
6. Efectúe una suma que encienda el flag V.
7. Efectúe una resta que encienda el flag N.
8. Efectúe una instrucción que encienda el flag N y deje apagado el flag C.
9. Usando la instrucción AND modifique el registro r0 poniendo en cero los 4 bits menos significativos preservando el valor de los 4 bits más significativos. Por ejemplo, si el número contenido en r0 es (00000000 00000000 00000000 01111010 binario), debe quedar (00000000 00000000 00000000 01110000 binario).
10. Usando AND, determine si el bit más significativo del registro r0 está encendido (1). Aclare cuál es la condición (flag) que indica si dicho bit vale 0 o 1.
11. Muestre cómo usar la instrucción AND para encender el bit más significativo del registro r0.
12. Muestre cómo usar la instrucción OR para encender el bit más significativo del registro r0.
13. Escriba un fragmento de código para obtener el producto entre los números 18 y 34. ¿Dónde queda almacenado el resultado?

Organización del Computador

Práctica - Assembler - Sumas y restas

14. Las etiquetas N1, N2 y N3 representan números sin signo. Escriba fragmentos de código assembler para almacenar en N3:
- la suma de N1 y N2.
 - el producto de N1 por N2.
 - el cociente N1 dividido el valor constante 2 (explorar las operaciones de desplazamiento).
15. Suponiendo que la etiqueta VALORES define una secuencia de cuatro números enteros sin signo, escriba las instrucciones necesarias para calcular el promedio (truncado a un número entero) de esos cuatro números. Defina una etiqueta PROMEDIO para almacenar el valor calculado.
16. Escriba un código assembler para obtener en r0 la cantidad de unos de la representación binaria del número contenido en r1 aplicando las operaciones de shift.
17. La siguiente etiqueta contiene un carácter ASCII que es un dígito decimal.
- ```
DIGITO: .byte '8'
```
- Escriba un programa que almacene el valor numérico correspondiente al dígito decimal en el registro r0.
18. Suponiendo que la etiqueta X apunta a la representación en texto ASCII de un número de 3 dígitos en base 10 (un carácter para cada dígito), obtenga en r0 el valor de ese número. Tener en cuenta que los caracteres '0' a '9' tienen código ASCII 30h a 39h.
19. Suponiendo que el registro r0 contiene un número entre 0 y 9, efectúe las operaciones necesarias para obtener el carácter ASCII que representa el dígito decimal correspondiente a dicho valor.
20. Agregue al ejercicio anterior las instrucciones necesarias para que se muestre en la pantalla el dígito contenido en el registro r0.