

Organización de Computadoras Guía de Ejercicios

No 2 - ASM Risc V

1. Escribe un programa en ensamblador que:
 - a. Cargue dos números en registros.
 - b. Realice suma, resta, multiplicación y división.
 - c. Almacene los resultados en memoria.
 - d. Verifica en "Register View" que los valores sean correctos.
2. Implementa una función que recorra un array de 5 enteros y sume todos sus elementos usando direccionamiento indexado.

Para los puntos 3 a 6 observa en el "Pipeline View" como se ejecutan los saltos, analiza el flujo de ejecución y responde:

- a) ¿Cómo afecta el salto condicional?
 - b) ¿Cuántos ciclos tarda en ejecutarse un bucle?
3. Implemente un programa que multiplique los primeros 4 números enteros positivos. (Empleando bucles)
 4. Escriba un programa que reciba un número y cuente regresivamente hasta 0 con un WHILE.
 5. Crea un programa en ensamblador que compare dos valores y guarde el mayor en memoria.
 6. Escribir un programa que implemente un contador ascendente hasta un valor determinado usando un ciclo FOR.

Para los siguientes puntos (7 y 8)

- a) Explica qué registros se usan más y por qué.
 - b) Usa "Memory View" para verificar el almacenamiento de datos ¿En qué direcciones se almacenan los valores?
7. Implemente un programa en ensamblador que determine si un número es positivo, negativo o cero y almacene un valor en memoria según el resultado.
 8. Implemente una subrutina que reciba dos números y devuelva su **máximo**.

Para los dos últimos puntos escribe dos versiones del mismo programa

- a) Una versión usando la pila (PUSH y POP).
- b) Otra sin usar la pila, solo registros.
- c) Compara el número de instrucciones ejecutadas en el pipeline.

9. Cálculo del factorial ($n!$)

- a. Pide un número (simulado con una variable en memoria).
- b. Calcula el factorial usando un bucle.
- c. Almacena el resultado en memoria.
- d. Revisa cómo varía el registro del contador en "Register View".

10. Fibonacci en ensamblador

- a. Genera la secuencia de Fibonacci hasta $n=10$.
- b. Guarda los valores en memoria.