Tecnología de Computadores

Ejercicios de programación en ensamblador

En esta relación se proponen varios problemas de programación en el lenguaje ensamblador de la máquina r-MIPS cuyo conjunto de instrucciones se detalla a continuación.

El procesador r-MIPS posee un banco de 32 registros de 32 bits nombrados de r0 a r31. El registro r0 es un registro de solo lectura que contiene el valor 0.

Conjunto de instrucciones

Instrucciones de transferencia:

- **LW (carga palabra):** Lee una palabra de la memoria de datos y la carga en un registro del procesador (rd). La dirección de la palabra de memoria a cargar se encuentra en la posición indicada por un registro más un desplazamiento entero (rf+desplazamiento).

  LW rd, desp.(rf) : rd ← MEM[rf+desp]

- **SW (almacena palabra):** Almacena el contenido de un registro (rd) en la memoria de datos. La posición de memoria se calcula sumando un desplazamiento entero al contenido de un registro (rf+desplazamiento).

  SW rd, desp.(rf) : MEM[rf+desp] ← rd

- **LUI (carga inmediato en mitad más significativa):** Carga la constante en la mitad superior del registro rd. Los bits de la mitad inferior del registro se ponen a 0.

  LUI rd, constante : rd ← constante, 0000000000000000

Instrucciones de transformación de la información:

- **ADD (suma), SUB (resta), OR (or lógico), AND (and lógico):** Instrucciones aritmético-lógicas con dos operandos registro. Leen los registros rf1 y rf2, realizan la operación correspondiente con la ALU y el resultado lo escriben en el registro rd.

  ADD rd, rf1, rf2 : rd ← rf1 + rf2
  SUB rd, rf1, rf2 : rd ← rf1 – rf2
  OR rd, rf1, rf2 : rd ← rf1 or rf2
  AND rd, rf1, rf2 : rd ← rf1 and rf2

- **ADDI (suma con inmediato):** Suma el contenido de un registro rf con una constante entera de 16 bits, guardando el resultado en el registro rd.

A la constante entera representada en complemento a 2 se le realiza una extensión de signo para completar los 32 bits.

ADDI rd, rf, constante : rd ← rf + constante

- **ORI (OR con inmediato):** Guarda la OR lógica del contenido del registro rf con la constante de 16 bits, extendida con ceros, en el registro rd.

ORI rd, rf, constante : rd ← rf or 0000000000000000,constante

- **SRL (desplazamiento lógico a la derecha), SLL (desplazamiento lógico a la izquierda), SRA (desplazamiento aritmético a la derecha):** Desplazan el contenido del registro rf según indica cada instrucción y el resultado lo almacenan en el registro rd.

SRL rd, rf : rd ← 0,rf[31:1]
SLL rd, rf : rd ← rf[30:0],0
SRA rd, rf : rd <- rf[31],rf[31:0]

- **SLT (comparación):** Compara el contenido de los registros rf1 y rf2. Si rf1 es menor que rf2 pone un 1 en rd, y un 0 en caso contrario.

SLT rd, rf1, rf2 : if (rf1<rf2) rd ← 1, else rd ← 0

Instrucciones de bifurcación:

- **BEQ (salto si igual):** Ejecuta un salto sólo si los valores almacenados en ambos registros (rf, rd) son iguales. En tal caso el destino de salto será el valor de PC+1 más un desplazamiento. Si no son iguales, se continúa con la siguiente instrucción dentro del flujo secuencial del programa.

BEQ rf, rd, desp. : if (rd = rf) PC ← PC+1+desp.

- **J (salto incondicional):** Esta instrucción fuerza un salto a la posición resultante de la concatenación de los 6 bits más significativos de PC+1, con la constante de 26 bits que se encuentra en la propia instrucción (dirección)

J dirección : PC ← (PC+1)[31:26],dirección[25:0]

Problemas

Problema 1:

Escribe un programa que inicialice a 1 el registro r1 y a 2 el registro r2.

Problema 2:

Supón inicializados los registros r1, r2 y r3 con números enteros. Escribe un programa que almacene en r4 el valor mayor de los almacenados en dichos registros y en r5 el valor menor.

Problema 3:

Los registros r1 y r2 contienen dos números naturales. Escribe un programa que guarde en r3 la media de los valores almacenados en r1 y r2.

Problema 4:

Escribe un programa igual que el anterior pero suponiendo que los números almacenados en r1 y r2 son números enteros.

Problema 5:

Hay 50 números enteros almacenados en las posiciones 100 a 149 de la memoria. Escribe un programa que los lea y almacene el valor máximo en el registro r1 y el mínimo en r2.

Problema 6:

Tenemos 250 números enteros en las posiciones 500 a 749 de memoria. Escribir un programa que almacene en r1 el número de posiciones de memoria con un valor mayor que cero, en r2 las de valor igual a 0 y en r3 las de valor menor que cero.

Problema 7:

Realiza un programa que multiplique el contenido del registro r1 por el contenido del registro r2 y el resultado lo almacene en r3.

Problema 8:

Realiza un programa que haga la división entera del contenido del registro 1 entre el contenido del registro r2, dejando en r3 el cociente y en r4 el resto.