

Práctico 4

Ejercicio 1

Se tiene un microprocesador RISC de 16 bits con 16 registros y el siguiente conjunto de instrucciones:

Instrucción	Descripción
NOP	No hace nada
MOV lnm, Reg1	Carga el inmediato lnm en los bits menos significativos de Reg1, dejando los bits más significativos en 0
MOVR Reg1, Reg2	Carga en Reg2 el contenido de Reg1
LOAD Reg1, Reg2	Carga el contenido de memoria apuntado por Reg1 en Reg2
SAVE Reg1, Reg2	Guarda el registro Reg1 en la dirección de memoria apuntada por Reg2
ADD Reg1, Reg2, Reg3	Suma Reg1 y Reg2 y guarda el resultado en Reg3
SUB Reg1, Reg2, Reg3	Realiza la operación Reg1-Reg2 y guarda el resultado en Reg3
SHL Reg1, Reg2, Reg3	Desplaza Reg1 a la izquierda Reg2 lugares y guarda el resultado en Reg3
SHR Reg1, Reg2, Reg3	Desplaza Reg1 a la derecha Reg2 lugares y guarda el resultado en Reg3
AND Reg1, Reg2, Reg3	Realiza un AND bit a bit entre Reg1 y Reg2 y guarda el resultado en Reg3
OR Reg1, Reg2, Reg3	Realiza un OR bit a bit entre Reg1 y Reg2 y guarda el resultado en Reg3
XOR Reg1, Reg2, Reg3	Realiza un XOR bit a bit entre Reg1 y Reg2 y guarda el resultado en Reg3
NOT Reg1, Reg2	Realiza un NOT bit a bit de Reg1 y guarda el resultado en Reg2
JMP Reg1	Hace un salto incondicional a la dirección apuntada por Reg1
JC Reg1	Hace un salto a la dirección apuntada por Reg1 si la bandera de C vale 1
JZ Reg1	Hace un salto a la dirección apuntada por Reg1 si la bandera de Z vale 1

Aclaraciones:

- Las flags de Z (cero) y C (acarreo) son afectadas por las operaciones aritméticas y lógicas.
- Para el caso de los desplazamientos, la flag de C corresponde al último bit que fue desplazado hacia fuera del registro.
- Los 16 registros llevan el nombre R1 hasta R16.

a) Defina un formato de instrucción para esta CPU. Indique el largo de instrucción de la misma.

b) Implemente en pseudocódigo un programa que sume el contenido de memoria de las direcciones comprendidas en el rango 0xABC7 - 0xABCF y almacene en la dirección 0x00FF el doble de dicha suma.

c) Implemente en lenguaje ensamblador el programa de la parte b)

nota: El programa estará cargado en memoria a partir de la dirección 0x0000.

Ejercicio 2

Sea una CPU de 16 bits con las siguientes características que se describen a continuación:

- Las instrucciones se codifican en 16 bits.
- Tiene 8 registros de uso general de 16 bits.
- Posee las siguientes instrucciones (se indica el código mnemónico y la semántica):
 1. **LOAD REG1, REG2** -> Instrucción que guarda en REG2 el contenido de la dirección de memoria almacenada en REG1.
 2. **STORE REG1, REG2** -> Instrucción que escribe el contenido de REG1 en la dirección de memoria almacenada en REG2.
 3. **NOP** -> Instrucción que solo consume ciclos de reloj (no hace nada).
 4. **SETLO INM, REG** -> Instrucción para cargar el valor constante INM de 8 bits en la parte baja de REG.
 5. **SETHI INM, REG** -> Instrucción para cargar el valor constante INM de 8 bits en la parte alta de REG.
 6. **OP REG1, REG2, REG3** -> Instrucciones aritméticas para suma (ADD) y resta (SUB), y operaciones lógicas and (AND) y or (OR) bit a bit. Estas operaciones trabajan sobre REG1 y REG2 como operandos y el resultado lo guardan en REG3.
 7. **SL (SR) REG1, INM, REG2** -> Instrucción que mueve a la izquierda (*derecha*) tantos bits de REG1 como indique el inmediato INM (de 4 bits) y el resultado lo pone en REG2
 8. **CMP REG1, REG2** -> Instrucción de comparación que realiza la resta REG1-REG2, actualizando las banderas de condición.
 9. **JZ (JM) INM** -> Instrucción de salto condicional, si la bandera de Zero (*Negative*) está en 1 salta INM (INM es un numero con signo de 12 bits) instrucciones desde la posición actual
 10. **JMP INM** -> Instrucción de salto incondicional, salta INM instrucciones (INM es un numero con signo de 12 bits) desde la posición actual.

Se pide:

- a) Diseñar el formato de instrucción presentado e indicar la codificación de cada instrucción.
- b) Implemente en pseudocódigo un programa que guarde en la dirección de memoria 0xFE11 el valor decimal 345 si el valor guardado en la dirección 0x1111 es igual al valor guardado en la dirección 0x1112. en caso contrario, el programa debe guardar en la dirección 0xFE12 el valor cero.
- c) Implemente en lenguaje ensamblador el programa de la parte b)

nota: El programa estará cargado en memoria a partir de la dirección 0x0000.

Ejercicio 3

Implemente el ciclo de instrucción y determine que etapas se realizan y cuáles no de las siguientes funciones: MOVR, LOAD, SAVE y ADD del ejercicio 1.