

## PROBLEMAS DE FORMATOS DE INSTRUCCIONES Y MODOS DE DIRECCIONAMIENTO

**1.** Sea una arquitectura de 20 bits (tamaño de palabra y de registro) con 8 registros y 120 instrucciones diferentes. El contenido de los registros y del contador de programa en hexadecimal es el siguiente:

R0 = 00000 R3 = 0000B R6 = 010FF  
R1 = 00000 R4 = 00001 R7 = 00100  
R2 = 003F5 R5 = 00010 PC = 0F000

En cualquier instrucción máquina de este computador, el orden en el que pueden aparecer los distintos campos sigue siempre la siguiente disposición de izquierda (bits más significativos) a derecha (bits menos significativos):

1. Código de operación
2. Dirección destino
3. Dirección origen
4. Desplazamiento
5. Valor inmediato

Todas las instrucciones tienen 20 bits. En el caso de existir un desplazamiento o un valor inmediato, ambos campos deben tener el mayor tamaño posible de modo que no queden bits en la instrucción sin utilizar. En caso contrario, los bits de la instrucción sin utilizar ocupan las posiciones menos significativas (a la derecha) de la palabra. El formato numérico empleado para almacenar todos los valores es el binario natural.

Dada la siguiente operación:

SUMA DESTINO ORIGEN -> DESTINO=DESTINO+ORIGEN

donde DESTINO hace referencia a una posición de memoria, registro o acumulador, y ORIGEN hace referencia a una posición de memoria, registro o valor inmediato. A continuación se enumeran distintas instrucciones implementadas en la arquitectura, así como su codificación en binario:

| Instrucción | Operación   | Codificación                  |
|-------------|---|-------------------------------|
| ADDI        | Suma un valor inmediato sobre registro                                  | <b>1011 10011110 00000000</b> |
| ADDR        | Suma en operaciones registro a registro                                 | <b>1011 00010100 00000000</b> |
| ADDM        | Suma con direccionamiento absoluto en operaciones de registro a memoria | <b>1011 00100111 11000000</b> |
| ADDRB       | Suma con direccionamiento relativo a registro base                      | <b>1011 01011110 10001100</b> |
| ADDPC       | Suma con direccionamiento relativo a contador de programa               | <b>1111 01010010 01011100</b> |

|         |  |                        |
|---------|--|------------------------|
| ADDACUM | Suma el contenido del registro al acumulador | 1111 11011000 00000000 |
|---------|--|------------------------|

Se pide:

- 1) Indicar el formato de cada una de las instrucciones anteriores, teniendo en cuenta que el código de la instrucción (OPCODE) tiene un tamaño fijo.
- 2) ¿Tiene sentido que el código de operación sea distinto cuando todas las instrucciones realizan una suma?
- 3) ¿A qué posición de memoria (en binario natural) accede la instrucción ADDM? ¿Cuál es el rango de posiciones distintas a las que podría acceder dicha instrucción?
- 4) ¿A qué posición de memoria (en binario natural) accede la instrucción ADDRb?
- 5) ¿A qué posición de memoria (en binario natural) accede la instrucción ADDPC?

**2.** Dado un procesador de 16 bits conectado a una memoria. El tamaño de la memoria principal viene determinado por el conjunto de posiciones direccionables en una palabra. El banco de registros incluye los registros de propósito general visibles al usuario (denotados por R0-R15). El juego de instrucciones de este ordenador incluye 180 instrucciones de máquina entre las que se encuentran las siguientes:

1. LOADI R, inm: Carga en el registro R el valor inmediato inm.
2. STORE R, dir: Almacena el contenido de R en la dirección de memoria dir.
3. ADDI R1, R2, inm: Suma el contenido de R2 con el valor inm y guarda el resultado en R1.
4. ADD R1, R2: Suma el contenido de R2 con el de R1 y guarda el resultado en R1.
5. BNEZ R, dir: Si el contenido del registro R es distinto de cero, salta a dir.

Las direcciones y valores inmediatos que se incluyen en las instrucciones son enteros de 16 bits. Indicar un formato posible para las instrucciones anteriores.

**3.** Dado un procesador de 16 bits que dispone de un banco de 6 registros, un espacio de direccionamiento de 64 kB. Asumiendo un código de operación de longitud fija, diseñar el formato de instrucción que permita construir un juego de instrucciones con al menos:

- a) 40 instrucciones aritmético-lógicas tipo registro-registro.
- b) 5 instrucciones de direccionamiento absoluto.
- c) 5 instrucciones de direccionamiento relativo a registro base con un desplazamiento máximo de 128 bytes.
- d) 2 instrucciones de direccionamiento indirecto.

**4.** Tenemos un procesador de 16 bits capaz de ejecutar 60 instrucciones máquina. El procesador tiene a su disposición 8 registros. Los programas pueden direccionar 64 KB de memoria principal. Se pide indicar el formato que debería tener la instrucción máquina:

ADDM R1, R2, M

donde R1 y R2 son registros y M es la dirección de una posición de memoria principal.

**5.** Dado un computador de 32 bits con 1GB de memoria y 16 registros. El juego de instrucciones de este computador incluye 200 instrucciones de máquina. Indicar el formato de las siguientes instrucciones:

1. LOAD\_ABS R, dir: Carga en el registro R el contenido de la dirección de memoria dir (direccionamiento absoluto).
2. LOAD\_PC R, desplz: Carga en el registro R el contenido de la posición de memoria cuya dirección se calcula como: contenido del PC + desplazamiento (direccionamiento relativo a PC)
3. LOAD\_REG R, Rb, desplz: Carga en el registro R el contenido de la posición de memoria cuya dirección se calcula como: contenido del registro Rb + desplazamiento (direccionamiento relativo a registro base).
4. LOAD\_IND R, dir: Carga en el registro R el contenido de la dirección de memoria apuntada por dir (direccionamiento indirecto).

**6.** Dado un computador con un banco de registros con 32 registros (R0-R31) de uso general de 32 bits y con un juego de 1100 instrucciones. El computador dispone de la siguiente instrucción:

POP displ(\$reg)

Dicha instrucción saca de la pila el dato que se encuentra en la cima y lo almacena en la dirección de memoria indicada por su operando. Obsérvese que el operando de la instrucción utiliza direccionamiento relativo a registro base. El desplazamiento debe poder expresar saltos positivos y negativos de hasta 1 MB. Se pide proponer un formato de instrucción para la instrucción POP.

**7.** Sea una arquitectura de 32 bits con una memoria de 4GB y un banco de 128 registros. El juego de instrucciones lo comprenden 200 instrucciones diferentes. Indicar un posible formato para las siguientes instrucciones:

1. STORE1 R, dir: almacena el contenido del registro R en memoria, empleando direccionamiento directo.
2. STORE2 R, Rb, desplz: almacena el contenido del registro R en memoria, empleando direccionamiento relativo a registro base.
3. STORE3 R: almacena el contenido del registro R en la pila.

**8.** Sea una arquitectura de 16 bits, con 16 registros y 40 instrucciones diferentes. Se desea implementar la instrucción ADDM, la cual suma el contenido de un registro con el contenido de una dirección de memoria principal, almacenando el resultado en esta posición de memoria. Indicar el formato de la instrucción para cada uno de los siguientes modos de direccionamiento de memoria: direccionamiento directo, direccionamiento relativo a registro base, direccionamiento relativo a contador de programa y direccionamiento indirecto. ¿Cuántos accesos a memoria se debe realizar en cada caso?

**9.** Sea un computador con palabras de 32 bits y 16 registros de 32 bits. La memoria es de 256 MB. El juego de instrucciones de esta máquina se compone de 120 instrucciones, entre las cuales está la siguiente:

ADD destino, operando1, operando2

Los modos de direccionamiento permitidos son: a) direccionamiento inmediato, b) direccionamiento directo, c) direccionamiento relativo a registro base, d) direccionamiento indirecto y e) direccionamiento registro-registro. Se pide diseñar el formato de la instrucción para cada uno de estos 5 modos de direccionamiento.

**10.** Sabiendo que un computador tiene una longitud de palabra de 32 bits, 8 registros, y que el repertorio de instrucciones tiene 16 instrucciones diferentes; si se supone que la instrucción de suma correspondiente queda definida en una palabra de memoria, calcular la máxima memoria direccionable en los siguientes casos:

a) ADD\_ACUM dir: suma el operando que está guardado en memoria en la posición dir con el contenido del acumulador, y guarda el resultado en el acumulador.

b) ADD R, dir: suma el operando que está guardado en memoria en la posición dir con el contenido del registro R, y guarda el resultado en R,

c) ADD R, dir1, dir2: suma dos operandos que están guardados en memoria (en las direcciones dir1 y dir2) y guarda el resultado en el registro R.

**11.** Sea un computador de 16 bits, con modelo de ejecución Reg-Mem, del que se conocen las siguientes características:

- Repertorio de instrucciones que permita realizar 64 operaciones distintas.
- Cuatro modos de direccionamiento diferentes: Inmediato (con operandos de tamaño de 16 bits), Directo, Indirecto, Relativo a Contador de Programa (el desplazamiento es de 8 bits).
- Banco de 16 registros.
- Memoria principal de 64 KB.

Indicar los distintos formatos de instrucción.

**12.** ¿Cuál es la diferencia entre una instrucción de direccionamiento inmediato, una de direccionamiento directo y una de direccionamiento indirecto? ¿Cuántas referencias de memoria se necesitan para cada uno de estos tipos de instrucciones para traer un operando a un registro del procesador?

**13.** Un computador de 16 bits, tiene en total 512 instrucciones diferentes y 20 registros de propósito general. Diseñe un formato de instrucciones para:

a) Una instrucción con dos operandos, uno de ellos está en un registro y otro se localiza en memoria, con direccionamiento directo.

b) Una instrucción de salto incondicional con direccionamiento relativo a PC.

**14.** Se considera una máquina de 16 bits con 25 instrucciones. La memoria principal es de 2 KB. Hay 32 registros. Algunas operaciones que debe poder ejecutar son:

1. STA1: Almacena el contenido del acumulador en un registro.

2. STA2: Almacena el contenido del acumulador en una dirección de memoria.

3. LDA1: Almacena en el acumulador un dato inmediato.

4. LDA2: Almacena en el acumulador el contenido de una posición de memoria.

5. LDA3: Almacena en el acumulador el contenido de un registro.

Diseñar los formatos de representación de estas instrucciones minimizando el espacio de representación, e indicando los modos de direccionamiento que puede admitir cada instrucción.