LABORATORIO DE ARQUITECTURA DE COMPUTADORES

GUÍA DEL ALUMNO

ISA and Pipelining

Objetivo

- Repaso de características relevantes de un ISA de tipo CISC y RISC

Modos de direccionamiento

Los modos de direccionamiento se describen en:

- Apuntes sobre ISA de la profesora en BB-Diapos, Slides & Docs
- Hennessy-Patterson, Computer Architecture: A Quantitative Approach, 1ª edición: cap. 3, 6ª edición: apéndice A
- Anasagasti, Fundamentos de los Computadores, 9ª edición cap. 6
- Prácticamente cualquier texto sobre Arquitecura de Computadores describirá modos de direccionamientos para los repertorios de instrucciones

Notación RTL

En todo el curso se utiliza la notación de transferencia entre registros (RTL) para describir tanto los pasos elementales de ejecución de una instrucción como para definir la operación que realiza una instrucción de tipo proceso (destino ← operando1 operación operando2), de tipo transferencia (destino ← fuente) o de tipo control (PC ← dirección de siguiente instrucción) siendo PC el Contador de programa, y destino y fuente registros o posiciones de memoria. M(X) significa el contenido de memoria en la dirección X.

Supongamos que debemos sumar dos números a y b y dejar el resultado en c. Las tres variables son posiciones de memoria que contienen algún valor entero. Para hacer esto en un procesador, según soporte el modelo de ejecución R-R, R-M o M-M sería:

R-R	**R-M**	**M-M**
R1 ← M(a)	R1 ← M(a)	R1← M(a) + M(b)
R2 ← M(b)	R1 ← R1 + M(b)	M(c) ← R1
R1 ← R2 + R1	M(c) ← R1	o simplemente:
M(c) ← R1		M(c) ← M(a) + M(b)

Para acceder a una posición de memoria apuntada por un registro (modo indirecto de registro), digamos R1, escribiríamos M(R1). Para el modo relativo con un desplazamiento inmediato, escribiríamos M(R1+despl).

Uso de repertorio CISC (Ej. 8086)

El siguiente ejemplo suma la constante 7 a cada elemento de la lista nums en un ISA 8086. En este repertorio, el registro RB se utiliza como apuntador y por ello recibe el nombre de registro base. Igualmente el registro SI se utiliza como índice para recorrer una lista de valores a partir de una dirección dada. Observar la notación RTL al lado de cada instrucción.

```
.DATA
nums  DB  1,2,3,4,5,6
.CODE
Ini:
MOV AX, @DATA       ; AX ← Dirección .DATA
MOV DS, AX          ; DS ← AX  (segmt Datos)
; ---------------------  Utilización del modo relativo a registro con el registro índice SI
MOV  SI, 0          ; SI  ← 0
MOV  CX, 6          ; CX ← 6   (número de elementos)
Bucle:
    ADD nums[SI], 7     ; M(SI+nums)← M(SI+nums) + 7
    INC SI              ; SI ← SI +1
LOOP Bucle          ; CX ← Cx-1 ; Si CX>0 go to bucle


; - - - - - - - - - - - - - - - - - utilización de un registro en modo indirecto de registro (Reg. base BX)
MOV  CX, 6          ; CX ← 6   (número de elementos)
LEA BX, nums        ; BX ← nums     (En DS:BX dirección de nums)
Bucle:
  MOV DL, [BX]      ; DL← M(BX)
  ADD DL, 7         ; DL ← DL + 7
  MOV [BX], DL      ; M(BX) ← DL
  INC BX            ; BX ← BX + 1
LOOP Bucle          ; CX ← Cx-1 ; Si CX>0 go to bucle
```

En otros repertorios sería posible incrementar el registro BX o SI en la propia instrucción en que se usa para acceder a la memoria: MOV [BX++], DL o ADD nums[SI++], 7, ahorrándose las instrucciones consiguientes. Esos modos de direccionamiento se denominas autoincrementados y son típicos de CISC.

Ejercicio 1: reescribir el bucle anterior utilizando modos autoincrementados.

Ejercicio 2: siguiendo las descripciones RTL que se muestran a continuación elaborar el programa correspondiente utilizando el repertorio de una máquina CISC con 8 registros GPR (R1-R8), de dos direcciones que soporta los modos de ejecución R-R y R-M, y modos de direccionamiento inmediato, directo de registro y relativo a registro con desplazamiento inmediato (sin autoincremento). Suponer mnemónicos **ADD** y **MOVE.** Para instrucciones de salto condicional usar "**BNEZ bucle**" (y no LOOP). En cada línea debe haber una instrucción en ensamblador y su descripción RTL. Se puede copiar la declaración de la zona de datos del ejemplo del 8086.

```
R1 ← 6
R2 ← 0
bucle:
 R3← M(R2+nums)
 R3 ← R3 + 7
  M(R2+nums) ←R3
  R2 ← R2 + 1
  R1 ← R1 - 1
Si R1>0 go to bucle
```

Ejercicio 3: Realizar con el repertorio CISC especificado en el ejercicio 2 junto con la notación RTL por cada instrucción, el programa correspondiente a este sencillo bucle en pseudocódigo:

```
int i=20, s, j;
int a[20] = {1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,20}

j=0;
s=0;
while i>0
  i--;
  s = s + a(j++);
end-while
```