

El Lenguaje Ensamblador

Moisés García Villanueva

19 de Septiembre de 2013

1. Estructura del procesador pentium, tipos de registros y modos de direccionamiento

1.1. Estructura de una computadora

La estructura interna o arquitectura de una computadora varía enormemente de unos modelos a otros sobre todo de la tarea a la que esté orientado el equipo. Así, una computadora de las denominadas “personales”, es decir, orientado a realizar tareas relativas a una persona, puede tener una estructura diferente a una computadora que utilizado para el almacenaje masivo y acceso al conjunto de datos de una empresa.

Como es imposible mostrar la gran variedad de arquitecturas que existen hoy en día en el mercado de computadoras, se ha optado por presentar un modelo simplificado de la arquitectura de una computadora personal, también conocida como PC que es la abreviatura del término en inglés de “personal computer”. La figura 1 ilustra este modelo.

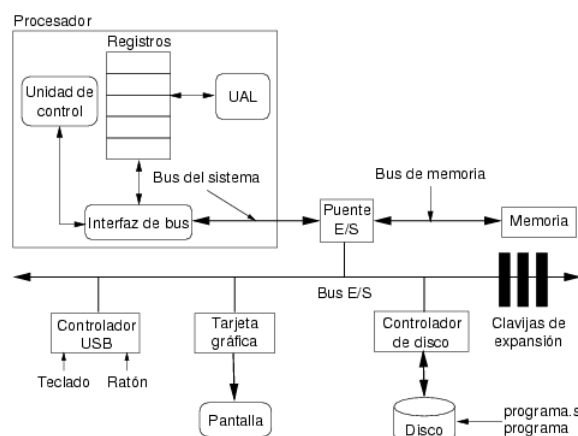


Figura 1: Estructura de una computadora.

1.2. Tipos de registros

Un **registro** es una memoria de alta velocidad y poca capacidad, integrada en el microprocesador, que permite guardar transitoriamente y acceder a valores muy usados, generalmente en operaciones matemáticas.

- Los **registros de datos** son usados para guardar números enteros. En algunas computadoras antiguas, existía un único registro donde se guardaba toda la información, llamado acumulador.
- Los **registros de memoria** son usados para guardar exclusivamente direcciones de memoria. Eran muy usados en la arquitectura Harvard, ya que muchas veces las direcciones tenían un tamaño de palabra distinto que los datos.
- Los **registros de propósito general** (en inglés GPRs o General Purpose Registers) pueden guardar tanto datos como direcciones. Son fundamentales en la arquitectura de von Neumann. La mayor parte de las computadoras modernas usa GPR.

- Los **registros de coma flotante** son usados para guardar datos en formato de coma flotante.
- Los **registros constantes** tienen valores creados por hardware de sólo lectura. Por ejemplo, en MIPS el registro cero siempre vale 0.
- Los **registros de propósito específico** guardan información específica del estado del sistema, como el puntero de pila o el registro de estado.

También existen registros banderas y de base

1.3. Modos de Direccionamiento

Definición: Los **modos de direccionamiento** de un procesador son las diferentes formas de transformación del campo de operando de la instrucción en la dirección del operando.

En esta definición el término dirección debe interpretarse en su sentido más general de localización del operando, en cualquier lugar, y no en el sentido más estricto de dirección de memoria.

1.3.1. Direccionamiento Inmediato

En este modo el operando es especificado en la instrucción misma. En otras palabras, una instrucción de modo inmediato tiene un campo de operando en vez de un campo de dirección. El campo del operando contiene el operando actual que se debe utilizar en conjunto con la operación especificada en la instrucción. Las instrucciones de modo inmediato son útiles para inicializar los registros en un valor constante.

Cuando el campo de dirección especifica un registro del procesador, la instrucción se dice que está en el modo de registro.

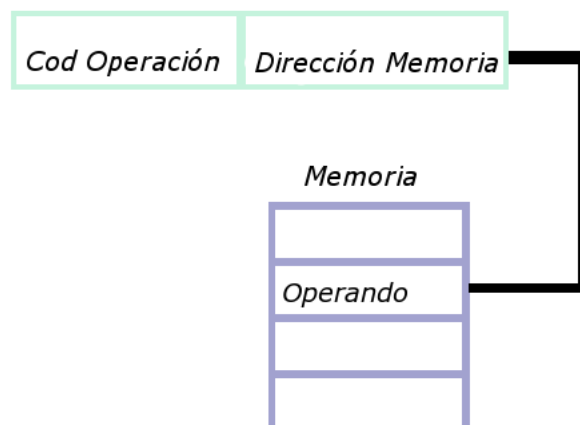
Ejemplo:

```
MOV A , #17H
```

1.3.2. Direccionamiento Directo

El campo de operando en la instrucción contiene la dirección en memoria donde se encuentra el operando. En este modo la dirección efectiva es igual a la parte de dirección de la instrucción. El operando reside en la memoria y su dirección es dada directamente por el campo de dirección de la instrucción.

Direccionamiento Directo



Ejemplo:

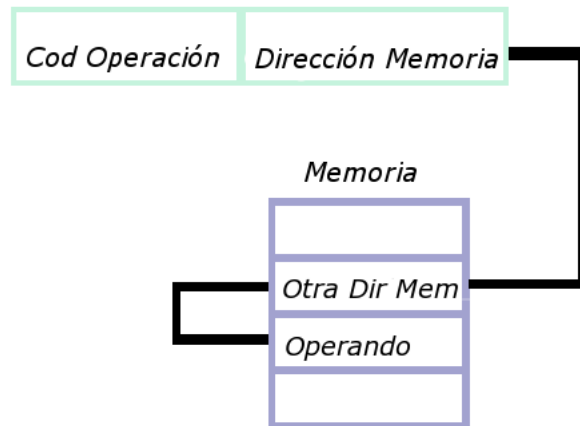
```
MOV A , 17H
```

1.3.3. Direccionamiento Indirecto

El campo de operando contiene una dirección de memoria, en la que se encuentra la dirección efectiva del operando. Ejemplo:

```
MOV A , @17H
```

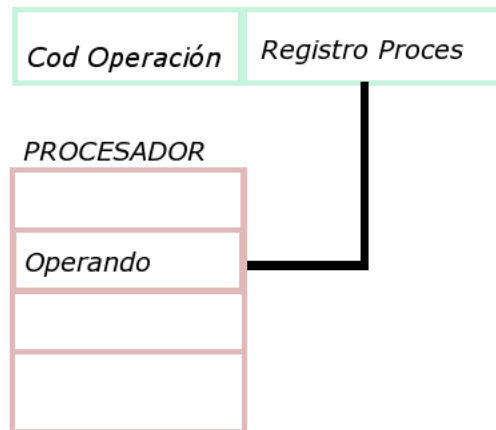
Direccionamiento Indirecto



1.3.4. Direccionamiento de Registro

Sirve para especificar operandos que están en registros.

Direccionamiento por Registro



Ejemplo:

```
MOV A , R0
```

1.4. Lenguaje Ensamblador

1.4.1. Lenguaje máquina

Se denomina Lenguaje de Máquina a un programa o secuencia de instrucciones que viene dado por una secuencia de códigos binarios.

1.4.2. Lenguaje ensamblador

El lenguaje ensamblador es una secuencia lógica de sentencias pertenecientes a alguna de las siguientes clases:

- Una línea de comentario.
- Una instrucción ejecutable.
- Una directiva de ensamblado

1.4.3. Hola Mundo con Ensamblador

Ejemplo del hola mundo en ensamblador utilizando el `nasm`.

```
section .data
    mensaje db "hola mundo",0xA
    longitud equ $ - mensaje
section .text
    global _start

_start:
    ;definimos el punto de entrada
    mov edx,longitud    ;EDX=long. de la cadena
    mov ecx,mensaje     ;ECX=cadena a imprimir
    mov ebx,1           ;EBX=manejador de fichero (STDOUT)
    mov eax,4           ;EAX=funcion sys_write() del kernel
    int 0x80            ;interrupcion 80 (llamada al kernel)
    mov ebx,0           ;EBX=codigo de salida al SO
    mov eax,1           ;EAX=funcion sys_exit() del kernel
    int 0x80            ;interrupc. 80 (llamada al kernel)

; Para obtener el codigo objeto hacemos: nasm -f elf32 hola.asm -o hola.o
; Para obtener el programa ejecutable: ld -s -o hola hola.o
```

1.4.4. Hola Mundo en el formato AT&T del ensamblador

Veamos el ejemplo del Hola Mundo para el `NASM`, esta vez en el formato `AT&T`.

```
.section .data
    mensaje: .ascii "hola mundo \n"
    longitud = . - mensaje
.section .text
    .globl _start
_start:
    movl $longitud,%edx
    movl $mensaje,%ecx #ECX=cadena a imprimir
    movl $1,%ebx       #EBX=manejador de fichero (STDOUT)
    movl $4,%eax       #EAX=FUNCION sys_write() del kernel
    int $0x80          #interrupc. 80 (llamada al kernel)
    movl $0,%ebx
    movl $1,%eax
    int $0x80
    # as -o hola.o hola.s
    # ld -o hola hola.o
```

2. Referencias

Arquitectura y programación en ensamblador del Intel Pentium Abelardo Pardo Universidad Carlos III de Madrid Departamento de Ingeniería Telemática

Avenida Universidad 30, E28911 Leganés (Madrid), SPAIN

Copyright © 2006 Universidad Carlos III de Madrid