

Conceptos básicos

Febrero 2013

Instrucción

- Es una operación que el procesador puede cumplir. Las instrucciones son almacenadas en la memoria principal, esperando a ser atendidas por el procesador.
- Las instrucciones están compuestas por:
 - El código de operación (acción que el procesador debe ejecutar)
 - El código operando (parámetros de la acción)

Tipos de instrucción

- Acceso a Memoria
 - Acceso a la memoria o transferencia de información entre registros.
- Operaciones Aritméticas
 - Operaciones tales como suma, resta, división o multiplicación.
- Operaciones Lógicas
 - Operaciones como Y, O, NO, NO EXCLUSIVO, etc.
- Control
 - Controles de secuencia, conexiones condicionales, etc.

Código de operación u opcode

- Porción de una instrucción de lenguaje máquina que especifica la operación a ser realizada.
- MOV es un código de operación que es formada por una lista de argumentos.

Ejemplo de código de operación

Instrucción:

MOV AL, 55h ; Copiar al registro bajo AL 55h (21 decimal)

Código máquina

B055

Código máquina en binario

10111000 0101010100000000 ; MOV AX, 0055h

10110100 01010101 ; MOV AH, 55h

10110 000 01010101; MOV AL, 55h

MOV

Registro
AL

55 en
binario

```
C:\8086>debug PROGRAMA.EXE
-p
AX=0CDE BX=0000 CX=0054 DX=0000 SP=0400 BP=0000 SI=0000 DI=0000
DS=0CCD ES=0CCD SS=0CE3 CS=0CDD IP=0003 NV UP EI PL NZ NA PO NC
0CDD:0003 8ED8          MOV     DS,AX
-p
AX=0CDE BX=0000 CX=0054 DX=0000 SP=0400 BP=0000 SI=0000 DI=0000
DS=0CDE ES=0CCD SS=0CE3 CS=0CDD IP=0005 NV UP EI PL NZ NA PO NC
0CDD:0005 B055          MOV     AL,55
-p
```

Conjunto de instrucciones

- Especificación detallada de instrucciones que un CPU puede ejecutar.
- Tiene aspectos del procesador que son visibles a un programador.
- Incluye:
 - Arquitectura de memoria
 - Interrupciones
 - Tipos de datos nativos
 - Instrucciones
 - Registros

Ejemplo de conjunto de instrucciones (8088)

- Aritmética
 - ADC, ADD, DIV, IDIV, IMUL, MUL, SBB, SUB
- Lógica
 - AND, NEG, NOT, OR, TEST, XOR
- Transferencia
 - MOV, MOVS (MOVSB) (MOVSW)
- Carga
 - LODS (LODSB) (LODSW), LAHF, LDS, LEA, LES

Ejemplo de conjunto de instrucciones (8088)

- Salto
 - JA (JNBE), JAE (JNB) (JNC), JB (JNAE) (JC), JBE (JNA), JCXZ, JE (JZ), JG (JNLE), JGE (JNL), JL (JNGE), JLE (JNG), JMP, JNE (JNZ), JNO, JNP (JPO), JNS, JO, JP (JPE), IS
- Loop
 - LOOP, LOOPE (LOOPZ), LOOPNE (LOOPNZ)
- Stack
 - POP, POPF, PUSH, PUSHF

Ejemplo de conjunto de instrucciones (8088)

- Conteo
 - DEC, INC
- Comparación
 - CMP, CMPS (CMPSB) (CMPSW)
- Bandera:
 - CLC, CLD, CLI, CMC, STC, STD, STI

¿Por qué aprender a programar en lenguaje ensamblador?

- Entender la interacción del hardware y software.
- Escribir un driver para un dispositivo muchas veces resulta más sencillo escribirlo en un lenguaje de este tipo.

Términos

- Bit – Un dígito del sistema binario.
- Byte – 8 dígitos
- Nibble – 4 dígitos
- Investigar:
 - Page (página)
 - Word (Palabra)

¿Por qué aprender a programar en lenguaje ensamblador?

- Una vez que se aprende el lenguaje ensamblador, es sencillo comprender otros.

Ensamblador

- Es un programa que “traduce programas” escritos en lenguaje ensamblador en instrucciones en lenguaje máquina.
- Guarda estrecha relación con la arquitectura del hardware.

MOV AX, 0001 => B8010000

Lenguaje ensamblador

- Hace referencia directamente con el hardware (AX, BX).
- Tiende a ser específico de un circuito integrado.
- La sintaxis del lenguaje es diferente de la que se utiliza para dar instrucciones al procesador.

Lenguaje ensamblador

- Las instrucciones no ejecutables que se emplean para estructurar el código fuente toman forma de pseudo-operaciones.
- El ensamblador divide la memoria del programa.

CPU

- En el CPU se conectan circuitos electrónicos de memoria de propósito general para formar *registros*.
- Los registros en la sintaxis del lenguaje ensamblador tienen distintos nombres.

Microprocesador

Esquema del microprocesador 8088

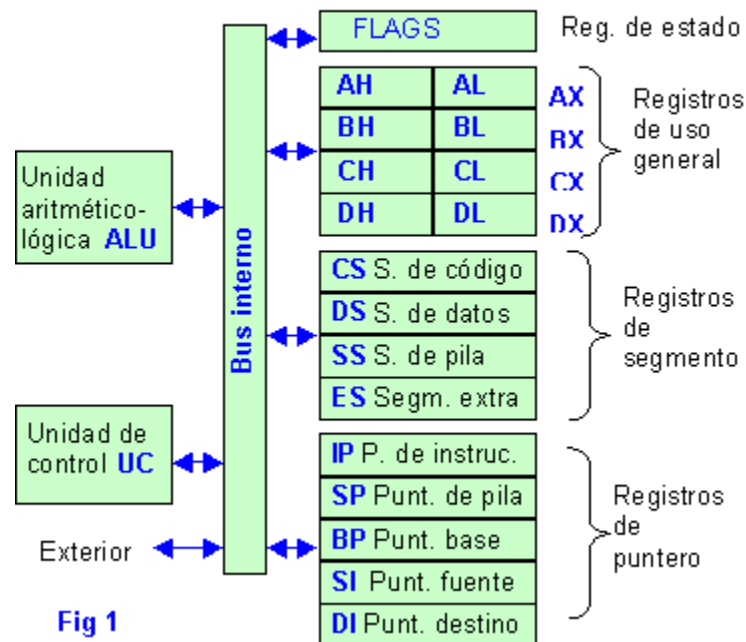


Fig 1

Registros

- Propósito general.
- Especiales, índices, apuntadores de pila, apuntador de instrucción.
- Banderas

Registros de 16 bits

Registros	Descripción
<i>AX (AH,AL)</i>	Registro acumulador
<i>BX (BH,BL)</i>	Registro base
<i>CX (CH, CL)</i>	Registro para conteo
<i>DX (DH, DL)</i>	Registro de datos
SP	Registro apuntador del stack
BP	Registro apuntador de base
SI	Registro índice del destino
CS	Registro del segmento de código
DS	Registro del segmento de datos
SS	Registro del segmento del stack
ES	Registro de segmento extra
<i>IP</i>	Apuntador de instrucciones
<i>SF</i>	Registro de estado de banderas

Registros de banderas

Registros	Descripción
CF	Bandera de acarreo.
PF	Bandera paridad, es 1 si el resultado de una operación de datos tiene un número par de bits iguales 1.
AF	Bandera de acarreo auxiliar.
ZF	Bandera activada si una operación es 0
SF	Bandera del signo, se activa si la operación es negativa.
TF	Bandera de trampa. Si la bandera se activa el 8086 ejecuta sólo una instrucción a la vez.
IF	El 8088 atenderá las interrupciones sólo cuando este se encuentre activo.
DF	Cuando es activada, los registros de índice hay decremento.
OF	Sobreflujo generado por una operación mayor que el máximo valor posible de representar.

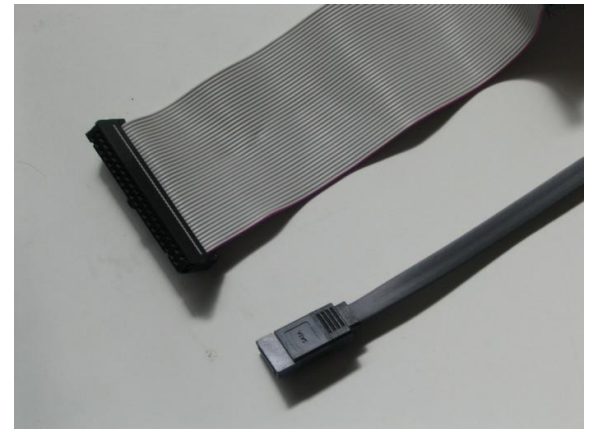
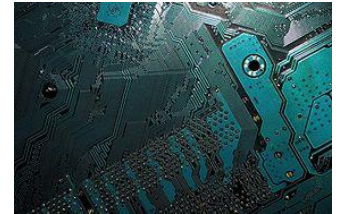
Direccionamiento a memoria

- Las direcciones están divididas en dos componentes:
 - Segmento y desplazamiento (offset).

Canales de comunicación

Bus

Vía de comunicación entre los distintos componentes de una computadora.



Bus paralelo

- Los datos son enviados por bytes al mismo tiempo, con la ayuda de varias líneas (funciones fijas).
- Líneas:
 - Bus de datos
 - Encargadas de indicar la posición de memoria o el dispositivo con el que se desea establecer comunicación.
 - Bus de dirección
 - Direccionar datos a su origen o destino
 - Bus de control
 - Envía señales de arbitraje entre los dispositivos
- Bus del procesador, los buses de discos duros, tarjetas de expansión y de vídeo, hasta las impresoras.

Bus serial

- Actualmente existe una tendencia hacia este tipo de bus.
- Los datos son enviados, bit a bit y se reconstruyen por medio de software.
- Está formado por pocos conductores.
- Existen buses para discos duros, tarjetas de expansión y procesadores.

Paralelo vs Serial

- Paralelo
 - Mayor velocidad
 - Mayor simplicidad
- Serial
 - Menor número de líneas
 - Menor costo (cuando las distancias aumentan)

Segmentos y desplazamientos

- Un segmento es un área continua de memoria que puede tener una longitud de 64K-bytes o mayor.
- El segmento debe comenzar con una localidad de memoria cuya dirección sea límite de 16 bytes (párrafo).
- La dirección de inicio de un segmento define su localización.

Segmentos

- El ensamblador permite que un programa en ejecución haga uso hasta de cuatro tipos de segmentos.
 - Código
 - Datos
 - Pila
 - Uno más de datos o segmento extra de código.

Tarea

Modos de operación del procesador

(Real, protegido y de administración)

Programas requeridos

- Editor
- MASM
- ASM
- LINK
- DEBUG

Mnemónico

- Un mnemónico es una palabra que sustituye a un código de operación (lenguaje de máquina).
- Ejemplo:
 - MOV
 - El procesador no entiende la palabra, se tiene que hacer una traducción a código objeto.

Formatos de instrucciones

[etiqueta] mnemónico de la instrucción [operando] [;comentario]

MOV AX, AX

SUB AX,AX

Formato de la instrucción substraer es:

SUB Destino, fuente

El contenido de AX (fuente) se resta de sí mismo (destino) y el resultado (cero) se guarda en AX.

Estructura de un programa en ensamblador

- Con un macroensamblador es posible procesar hasta cuatro tipos de segmentos.
- El programa requiere al menos de dos segmentos:
 - El de código
 - El de pila

Instrucciones

- ADD
- SUB
- JMP
- MOV

Actividad

- Preparar una exposición con los siguientes temas:
 - Registros de propósito general
 - Registros de segmento
 - Registro de puntero
 - Registros de banderas o estados

Código

.MODEL SMALL ;**Modelo de memoria**

.DATA

Texto DB ' 1', 0Dh, 0Ah, '\$'

Texto1 DB ' 123', 0Dh, 0Ah, '\$'

Texto2 DB '12345', 0Dh, 0Ah, '\$'

.STACK

.CODE

Programa:

MOV AX, @DATA

MOV DS, AX

MOV AX, 4C00H ; **Función para finalizar el programa**

INT 21H

END Programa

Modelos de memoria

- TINY.- Exclusivo de programas .COM, los cuales tienen sus datos, código y pila en un segmento.
- SMALL.- Exige que el código quepa en un segmento de 64K y los datos en otro segmento de 64K.
- La directiva .MODEL genera automáticamente las asociaciones necesarias para los segmentos.

Tarea

- Modelos de memoria:
 - MEDIUM, COMPACT, LARGE
- Investigar las directivas:
 - ASSUME
 - SEGMENT
 - OFFSET