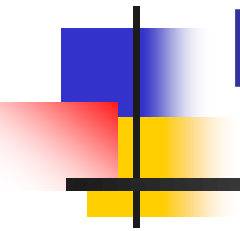
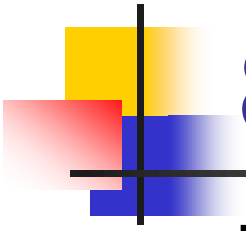


Sistemas de Procesamiento de Datos

Assembler (parte 3)

Profesor: Fabio Bruschetti
Aydte: Pedro Iriso
Ver 2019

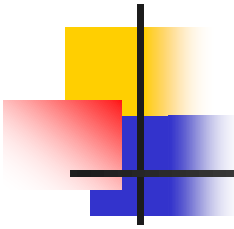


8086 – Instrucciones

- Las instrucciones de assembler son formas legibles de las instrucciones de máquina
- Se usan mnemónicos para representarlos:
  - MOV SUB ADD JMP
- Las instrucciones están compuestas por:
  - Operación Como se usan los valores de las variables
  - Operandos Que variables de estado se usarán
- Los operandos se pueden especificar de varias formas:
  - Modos simples: registro, inmediato, directo
  - Mas poderoso: Indirecto

8086 - Modos de Direccionamiento

- Modos de Direccionamiento
  - Registro,Registro
    - MOV AX, BX — AX := BX
  - Registro,Inmediato
    - ADD DX, 1 — DX := DX +1
  - Registro,Directo Memoria
    - SUB DX, [1] — DX := DX - m[DS:0001]
    - SUB DX, var — DX := DX - m[DS:var]
  - Registro Indirecto
    - INC [BX] — m[DS:BX] := m[DS:BX] + 1
  - Base Indirecto
    - MOV AX, [BX+1] — AX := m[DS:BX + 1]
  - Base-Indice Indirecto
    - MOV AX, [BX+SI] — AX := m[DS:BX + SI]
  - Base-Indice Indirecto con desplazamiento
    - MOV AX, [BX+SI+1] — AX := m[DS:BX + SI + 1]

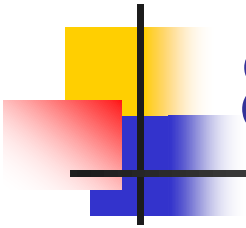


8086 - Modos de Direccionamiento

- Modos de Direccionamiento
  - Registro, Registro
    - Permite especificar un registro como operando
  - Registro, Inmediato
    - Permite especificar una constante como origen
    - La constante se codifica según el tamaño del destino 8 bits o 16 bits
  - Registro, Directo Memoria
    - Permite especificar el offset de la dirección de una variable como operando.
    - Si el offset es una constante, se codifica como parte de la instrucción
    - Durante la ejecución se combinan con DS
  - Registro Indirecto
    - Un registro (BX, SI, DI, BP) contiene el offset del operando.
    - Se combina DS y SS para BP
    - Es muy util para manejar arrays de elementos.

8086 - Modos de Direccionamiento

- Modos de Direccionamiento
  - Base Indirecto
    - Similar a registro indirecto, excepto que se especifica tambien una constante.
    - Durante la ejecución, el procesador usa un registro temporal para calcular Registro + constante.
    - Registros permitidos BX, BP, SI o DI
    - MOV AX,[BX+2] == MOV AX, 2[BX]
  - Base-Indice Indirecto
    - Similar a base indirecto, excepto que se especifica un segundo registro en lugar de la constante.
    - Restricciones:
      - Uno de los registros debe ser un registro BASE: BX (combina DS) o BP (combina SS)
      - Otro de los registros debe ser un registro INDICE: SI o DI
  - Base-Indice Indirecto con desplazamiento
    - Similar a base-Indice indirecto, excepto que se especifica una constante.
    - MOV AX,[BX+SI+1] == MOV AX, [BX][SI+1]



8086 – Instrucciones

- Cargando Direcciones en Registros:
  - Antes que las instrucciones usen direccionamiento indirecto, los registros deben ser cargados con direcciones.
  - Dos formas de calcular y cargar la dirección de memoria efectiva de 16 bits del operando
    - MOV BX, OFFSET w
    - LEA BX, w
  - Por default el segmento asociado a BX, SI y DI es DS y para BP es SS, por lo tanto:
    - MOV [BX], AL == MOV DS:[BX] , AL
    - Se puede cambiar los segmentos asociados:
      - MOV SS:[BX], AL
      - MOV ES:[BX], AL

8086 – Instrucciones

- Compatibilidad de operando con memoria
  - MOV [BX] , 1 ????
  - Calificadores de acceso a memoria
    - WORD PTR puntero a word - operando de 16 bits
    - BYTE PTR puntero a byte - operando de 8 bits
  - MOV WORD PTR [BX], 1 //Destino de 16 bits.
- Errores comunes
  - w DW 0AA33h //origen 16 bits
  - MOV AL, w //destino 8 bits

8086 – Instrucciones

- ADD (Flags afectados : AF, PF, CF, SF, OF, ZF)
  - ADD reg, reg
  - ADD mem, reg
  - ADD reg, mem
  - ADD reg, inmed
  - ADD mem, inmed
  - ADD acum, inmed
- MOV (Flags afectados : Ninguno)
  - MOV reg, reg
  - MOV mem, reg
  - MOV reg, mem
  - MOV reg16, segreg
  - MOV segreg, reg16
  - MOV reg, inmed
  - MOV mem, inmed
  - MOV mem16, segreg
  - MOV segreg, mem16
- JMP, JZ, JC, JO, JS, JP, JNZ, JNC, JNO, JNS, JNP
  - Z=zero, C=carry, O=overflow, S=sign, P=parity, N=NOT
- CMP dest, src
  - Realiza dest - src y setea los FLAGS, no almacena el resultado

8086 – Instrucciones de Salto

- Instrucciones de Salto
  - Unario (Incondicional) : Siempre salta: JMP dest
  - Simple: salta si el flag especificado está en 1 JC dest
  - SinSigno: salta cuando se comparan o testean dos numeros sin signo y resulta en una combinacion de flags de estado JA dest
  - ConSigno: salta cuando se comparan o testean dos numeros con signo y resulta en una combinacion de flags de estado JG dest
- Sin signo y Con Signo

JA	Above	JG	Greater
JAE	Above or Equal	JGE	Greater or Equal
JB	Below	JL	Less
JBE	Below or Equal	JLE	Less or Equal

8086 – Instrucciones de Salto

- Saltos Condicionales
  - AL contiene 7Fh
  - Sin signo / Con Signo

Sin signo	Con Signo
CMP AL, 80h	CMP AL, 80h
JA MasGrande	JG MasGrande

  - Cual de los saltos anteriores se realizará?
- Limitación de los Saltos
  - Condicionales restringidos a offsets relativos de 8-bits signados
    - No pueden saltar muy lejos: -128 a 127 bytes.
  - Incondicionales restringidos a offset relativos de 16 bits
    - Para saltar por una condición mas alla de los 127 bytes, se usa la instrucción de salto condicional con la condición negada y un JMP a continuación.

8086 – Instrucciones – Ciclos

- LOOPs
  - Muy util cuando se tiene una acción que se repite un número de veces for (i=max; i>0; i--)

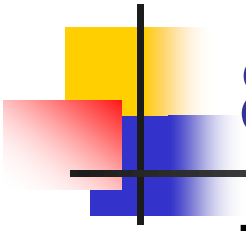
```
        MOV CX, max
L:      ......
        SUB CX, 1
        JNZ L
```

  - Funcionalmente equivalentes pero mas optimizados

```
        MOV CX, max
L:      ......
        LOOP  L
```

  - Auto-decrementa CX, y solo funciona sobre CX

8086 – Instrucciones – División



- DIV src
  - División de enteros sin signo : Acumulador / src
  - El tamaño del divisor (8bits o 16bits) se determina dependiendo de src
  - src no puede ser de direccionamiento inmediato.
- Division de 8 bits: (src es operando de 8 bits)
  - Se divide un valor de 16 bits por src sobre AX:
    - AL := AX / src
    - AH := AX mod src
  - Los flags no tienen significado alguno.
- Division de 16 bits (src es operando de 16 bits)
  - Se divide un valor de 32 bits por src sobre DX concatenado a AX:
    - AX := DX:AX / src
    - DX := DX:AX mod src

8086 – Instrucciones – Shift y Rotate

- Shift y Rotate
  - Shift a derecha lógico:
    - SHR AL, 1
  - Shift a derecha aritmetico
    - MOV CL, 2
    - SAR AL, CL
  - Shift a izquierda lógico y aritmético
    - SHL AL, 1
    - SAL AL, 1
  - Rotar Carry a izquierda
    - RCL AL, 1
  - Rotar a izquierda
    - MOV CL, 4
    - ROL AL, CL