

Arquitectura de computadores II

Repertorio de Instrucciones II: Modos de direccionamiento

Marzo de 2022

1 Modos de direccionamiento

2 Lenguaje ensamblador

- Modelo procesador 8086/8088
- Generalidades
- Direccionamiento en ensamblador
- Juegos de instrucciones
- Etiquetas, comentarios y directivas

Contenido

1 Modos de direccionamiento

2 Lenguaje ensamblador

- Modelo procesador 8086/8088
- Generalidades
- Direccionamiento en ensamblador
- Juegos de instrucciones
- Etiquetas, comentarios y directivas

Modos de direccionamiento

Tipos de direccionamiento

- **Direccionamiento inmediato:** Se envía el dato directamente
- **Direccionamiento directo:** La dirección del operando está en el campo de direcciones
- **Direccionamiento de indirecto:** El campo de direcciones apunta a la posición que contiene la dirección del operando
- **Direccionamiento a registro** Puede ser directo, indirecto o con desplazamiento

Modos de direccionamiento

Direccionamiento inmediato

El operando está presente en la instrucción:

$$\textit{Operando} = A$$

Donde A es una constante que puede ser numérica, booleana, carácter, etc. Ejemplo *ADD5*. Aquí añadimos 5 al acumulador.

Direccionamiento inmediato

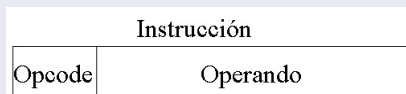


Figura 1: Direccionamiento inmediato

Modos de direccionamiento

Direccionamiento directo

El operando contiene la dirección efectiva del dato

$$EA = A$$

Donde EA es la dirección del dato A . Ejemplo $ADD[2]$, donde $[2]$ es la dirección que contiene el dato .

Modos de direccionamiento

Direccionamiento directo

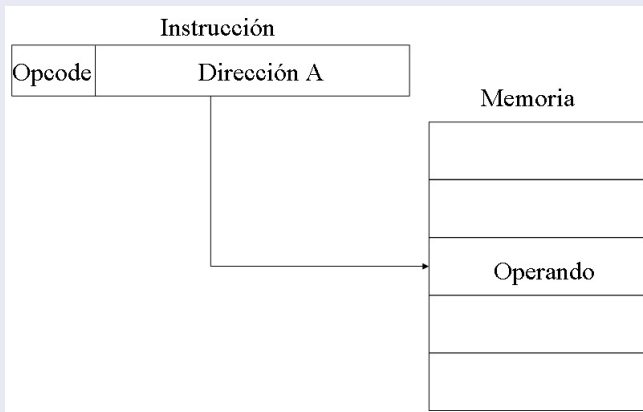


Figura 2: Direccionamiento directo

Modos de direccionamiento

Direccionamiento indirecto

El problema del direccionamiento directo es la longitud del campo de direcciones que normalmente es menor que la longitud de la palabra, limitando el rango de direcciones. Una solución es que el campo de direcciones apunte a una dirección en memoria que contiene la dirección al dato de interés, bajo este enfoque podemos tener 2^n posibles direcciones, donde n es el tamaño de la palabra en memoria, sin embargo este modo es más lento.

$$EA = (A)$$

Donde EA es la dirección del dato (A) el cual contiene la dirección al dato. Ejemplo $ADD([2])$, donde $([2])$ es una dirección a una localización de memoria que contiene la dirección del dato del operando.

Instrucciones

Direccionamiento indirecto

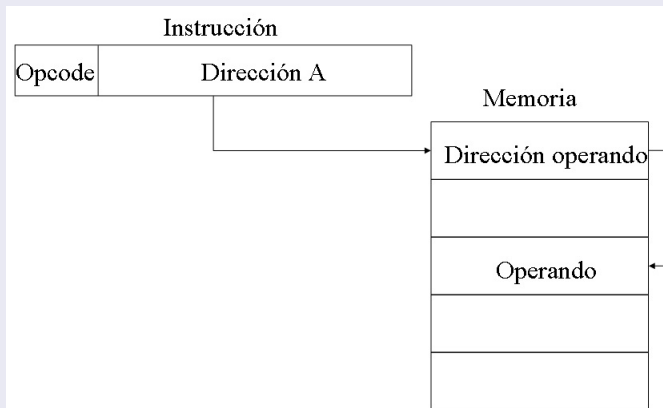


Figura 3: Direccionamiento directo

Modos de direccionamiento

Direccionamiento directo con registro

El registro contiene la dirección efectiva del dato

$$EA = R$$

Donde EA es la dirección contenida en el registro R . Ejemplo $ADD[X]$, donde $[X]$ es el registro que contiene el dato.

Modos de direccionamiento

Direccionamiento directo con registro

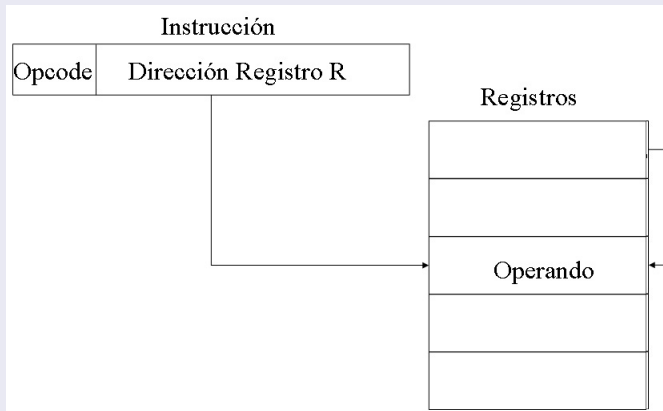


Figura 4: Direccionamiento directo con registro

Modos de direccionamiento

Direccionamiento indirecto con registro

$$EA = (R)$$

Donde EA es la dirección que contiene (R) . Ejemplo $ADD([X])$, donde $([X])$ es la dirección a la posición de memoria que contiene la dirección del dato.

Modos de direccionamiento

Direccionamiento indirecto con registro

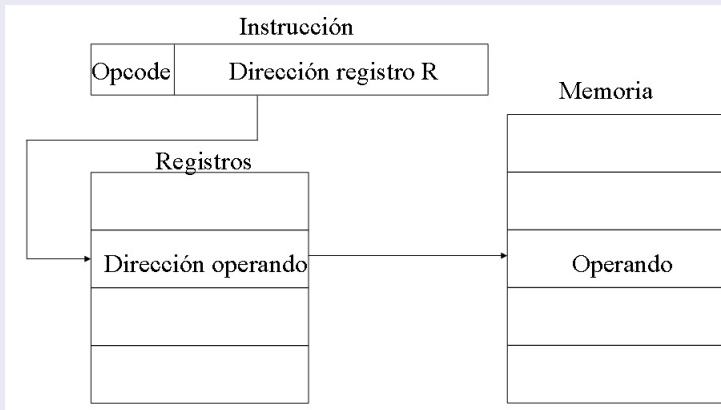
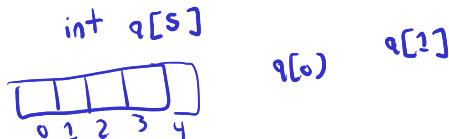


Figura 5: Direccionamiento indirecto con registro

Modos de direccionamiento



Direccionamiento con desplazamiento

$$EA = A + (R)$$

Donde EA es la dirección que resulta al sumar los contenidos del registro (R) y A .

Modos de direccionamiento

Direccionamiento con desplazamiento

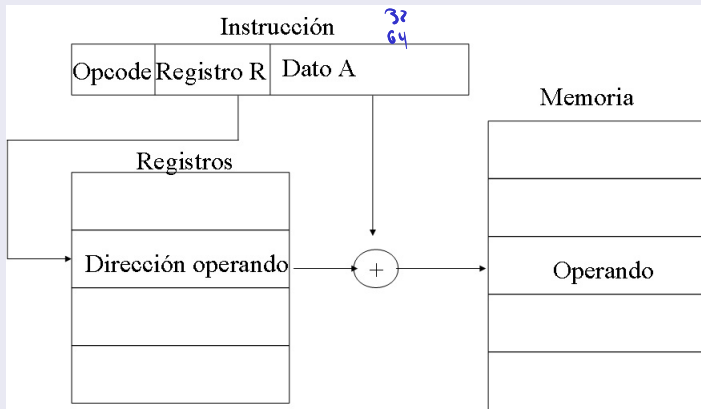


Figura 6: Direccionamiento con desplazamiento con registro

Contenido

1 Modos de direccionamiento

2 Lenguaje ensamblador

- Modelo procesador 8086/8088
- Generalidades
- Direccionamiento en ensamblador
- Juegos de instrucciones
- Etiquetas, comentarios y directivas

Contenido

1 Modos de direccionamiento

2 Lenguaje ensamblador

- Modelo procesador 8086/8088
- Generalidades
- Direccionamiento en ensamblador
- Juegos de instrucciones
- Etiquetas, comentarios y directivas

Lenguaje ensamblador

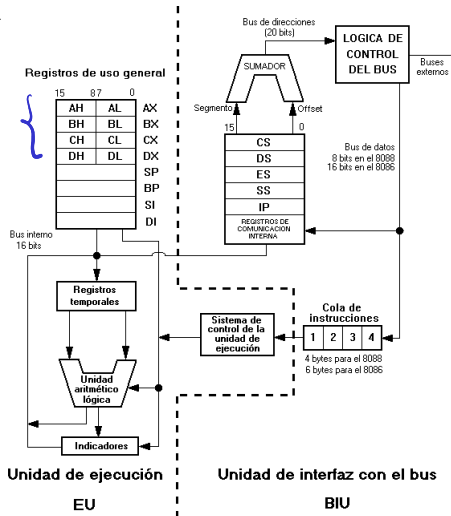
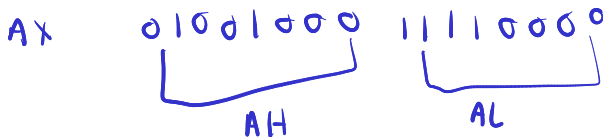


Figura 7: Modelo procesador 8086/8088

Lenguaje ensamblador

Registros

El 8086/88 dispone de 4 registros de datos, 4 registros de segmento, 5 registros de índice y 1 registro de estado.



Lenguaje ensamblador

Registros de datos

Los registros de datos son de 16 bits, aunque están divididos. lo que permite su acceso en 8 bits. Estos registros son de propósito general aunque todos tiene alguna función por defecto.

1. **AX (acumulador)** se usa para almacenar el resultado de las operaciones, es al único registro con el que se puede hacer divisiones y multiplicaciones. Puede ser accedido en 8 bits como AH para la parte alta (HIGH) y AL (LOW) para la parte baja.
2. **BX (registro base)** almacena la dirección base para los accesos a memoria. También puede accederse como BH y BL, parte alta y baja respectivamente.

Lenguaje ensamblador

Registros de datos

Los registros de datos son de 16 bits, aunque están divididos. lo que permite su acceso en 8 bits. Estos registros son de propósito general aunque todos tiene alguna función por defecto.

3. **CX (contador)** actúa como contador en los bucles de repetición. CL (parte baja del registro) almacena el desplazamiento en las operaciones de desplazamiento y rotación de múltiples bits.
4. **DX (datos)** es usado para almacenar los datos de las operaciones.

Lenguaje ensamblador

Registros de segmento

Los registros de segmento son de 16 bits y contienen el valor de segmento.

1. **CS (segmento de código)** contiene el valor de segmento donde se encuentra el código. Actúa en conjunción con el registro IP (que veremos más adelante) para obtener la dirección de memoria que contiene la próxima instrucción.
2. **DS (segmento de datos)** contiene el segmento donde están los datos

Lenguaje ensamblador

Registros de segmento

Los registros de segmento son de 16 bits y contienen el valor de segmento.

3. **ES (segmento extra de datos)** es usado para acceder a otro segmento que contiene más datos.
4. **SS (segmento de pila)** contiene el valor del segmento donde está la pila. Se usa conjuntamente con el registro SP para obtener la dirección donde se encuentra el último valor almacenado en la pila por el procesador

Lenguaje ensamblador

Registros de índice

Estos registros son usados como índices por algunas instrucciones. También pueden ser usados como operandos (excepto el registro IP).

3. **DI (índice de destino)** almacena el desplazamiento del operando de destino en memoria en algunos tipos de operaciones (operaciones con operandos en memoria).
4. **SP (índice de pila)** almacena el desplazamiento dentro del segmento de pila, y apunta al último elemento introducido en la pila. Se usa conjuntamente con el registro SS.
5. **BP (índice de base)** se usa para almacenar desplazamiento en los distintos segmentos. Por defecto es el segmento de la pila

Lenguaje ensamblador

Registros de índice

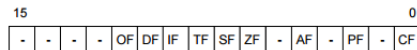
Estos registros son usados como índices por algunas instrucciones. También pueden ser usados como operandos (excepto el registro IP).

3. **ES (segmento extra de datos)** es usado para acceder a otro segmento que contiene más datos.
4. **SS (segmento de pila)** contiene el valor del segmento donde está la pila. Se usa conjuntamente con el registro SP para obtener la dirección donde se encuentra el último valor almacenado en la pila por el procesador

Lenguaje ensamblador

Registros de estado

El registro de estado contiene una serie de banderas que indican distintas situaciones en las que se encuentra el procesador



OF: Desbordamiento
DF: Dirección en operaciones con cadenas
IF: Indicador de interrupción
TF: Modo traza
SF: Indicador de signo en operaciones con signo
ZF: Indicador de cero
AF: Acarreo del bit 3 en AL
PF: Bit de paridad
CF: Acarreo

Figura 8: Estado del procesador

Lenguaje ensamblador

Registros de estado

El registro de estado contiene una serie de banderas que indican distintas situaciones en las que se encuentra el procesador

1. OF (desbordamiento) es el principal indicador de error producido durante las operaciones con signo. Vale 1 cuando:
 - La suma de dos números con igual signo o la resta de dos números con signo opuesto producen un resultado que no se puede guardar (más de 16 bits).
 - El bit más significativo (el signo) del operando ha cambiado durante una operación de desplazamiento aritmético.
 - El resultado de una operación de división produce un cociente que no cabe en el registro de resultado

Registros de estado

El registro de estado contiene una serie de banderas que indican distintas situaciones en las que se encuentra el procesador

2. **DF (dirección en operaciones con cadenas)** si es 1 el sentido de recorrido de la cadena es de izquierda a derecha, si es 0 irá en sentido contrario.
3. **IF (indicador de interrupción)** cuando vale 1 permite al procesador reconocer interrupciones. Si se pone a 0 el procesador ignorará las solicitudes de interrupción.
4. **TF (modo traza)** indica al procesador que la ejecución es paso a paso. Se usa en la fase de depuración.
5. **SF (indicador de signo)** solo tiene sentido en las operaciones con signo. Vale 1 cuando en una de estas operaciones el signo del resultado es negativo.

Lenguaje ensamblador

Registros de estado

El registro de estado contiene una serie de banderas que indican distintas situaciones en las que se encuentra el procesador

7. **ZF (indicador de cero)** vale 1 cuando el resultado de una operación es cero.
8. **AF (acarreo auxiliar)** vale 1 cuando se produce acarreo o acarreo negativo en el bit **MSB**
9. **PF (paridad)** vale 1 si el resultado de la operación tiene como resultado un número con un número par de bits a 1. Se usa principalmente en transmisión de datos.
10. **CF (bit de acarreo)** vale 1 si se produce acarreo en una operación de suma, o acarreo negativo en una operación de resta. Contiene el bit que ha sido desplazado o rotado fuera de un registro o posición de memoria. Refleja el resultado de una

Contenido

1 Modos de direccionamiento

2 Lenguaje ensamblador

- Modelo procesador 8086/8088
- Generalidades
- Direccionamiento en ensamblador
- Juegos de instrucciones
- Etiquetas, comentarios y directivas

Lenguaje ensamblador

Definiciones

1. Las instrucciones en ensamblador se convierten directamente en lenguaje de máquina
2. Tiene dos segmentos de memoria: programa y datos, recuerden el modelo de Von Neuman.
3. El programa inicia en la posición de memoria 101
4. Los datos inician en la posición de memoria 201
5. En el curso vamos a utilizar ensamblador para el modelo de procesador 8086/8088. Este lo veremos la próxima clase.

Lenguaje ensamblador

Tipos de datos

1. **Nibble:** 4 bits
2. **Byte:** 8 bits
3. **Word:** 16 bits
4. **DWord:** 32 bits

Representación datos

1. Binario: $10101b$
2. Hexadecimal: $6h$
3. Decimal 6
4. Texto *'c'* o *"casa"*

Lenguaje ensamblador

Representación textos

- Para el computador los textos son secuencias de números
- Por ejemplo 'c' es equivalente a tener 99
- Esto es muy importante, ya que en ensamblador cuando se capturan los textos del teclado se captura su valor numérico, por ejemplo si presionamos 'c' en el teclado, se obtendrá 99
- Existen dos extensiones de código ASCII, de 128 caracteres (simple) y 256 (extendido)

Lenguaje ensamblador

TABLA DE CARACTERES DEL CÓDIGO ASCII

1	25	49	73	97	121	145	169	193	217	241
2	26	50	74	98	122	146	170	194	218	242
3	27	51	75	99	123	147	171	195	219	243
4	28	52	76	100	124	148	172	196	220	244
5	29	53	77	101	125	149	173	197	221	245
6	30	54	78	102	126	150	174	198	222	246
7	31	55	79	103	127	151	175	199	223	247
8	32	56	80	104	128	152	176	200	224	248
9	33	57	81	105	129	153	177	201	225	249
10	34	58	82	106	130	154	178	202	226	250
11	35	59	83	107	131	155	179	203	227	251
12	36	60	84	108	132	156	180	204	228	252
13	37	61	85	109	133	157	181	205	229	253
14	38	62	86	110	134	158	182	206	230	254
15	39	63	87	111	135	159	183	207	231	255
16	40	64	88	112	136	160	184	208	232	PRESIONA LA TECLA
17	41	65	89	113	137	161	185	209	233	Alt
18	42	66	90	114	138	162	186	210	234	MÁS EL NÚMERO
19	43	67	91	115	139	163	187	211	235	
20	44	68	92	116	140	164	188	212	236	
21	45	69	93	117	141	165	189	213	237	
22	46	70	94	118	142	166	190	214	238	
23	47	71	95	119	143	167	191	215	239	
24	48	72	96	120	144	168	192	216	240	

Figura 10: Código ASCII

Contenido

1 Modos de direccionamiento

2 Lenguaje ensamblador

- Modelo procesador 8086/8088
- Generalidades
- Direccionamiento en ensamblador
- Juegos de instrucciones
- Etiquetas, comentarios y directivas

Lenguaje ensamblador

Direccionamiento de registro

Cuando ambos operadores son un registro

```
MOV AX,BX ; Transfiere el contenido de BX a AX
```

Lenguaje ensamblador

Direccionamiento inmediato

Cuando el operador origen es una constante

```
MOV AX, 500; Carga en AX el valor 500
```

Lenguaje ensamblador

Direccionamiento directo

Cuando el operando es una dirección de memoria

```
MOV BX, [1000]; Almacena en BX el contenido de la dirección 1000 de memoria  
MOV AX, TABLA; Almacena en AX el contenido de la dirección en la constante TABLA
```

Luego se explicará como definir constantes

Lenguaje ensamblador

Direccionamiento directo

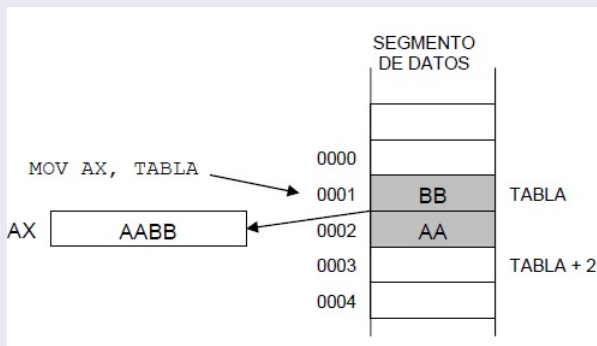


Figura 11: Lenguaje ensamblador

Lenguaje ensamblador

Direccionamiento indirecto

Cuando el operador está en memoria en una posición contenida por un registro

```
MOV AX, [BX]; Almacena en AX el contenido de la dirección  
             de memoria de BX  
MOV [BP], CX; Almacena en la dirección en memoria  
             indicada por el contenido BP el dato de CX
```

Lenguaje ensamblador

Direccionamiento directo

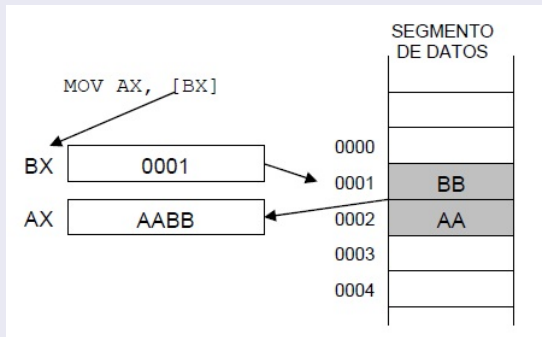


Figura 12: Lenguaje ensamblador

Lenguaje ensamblador

Direccionamiento por registro base

Cuando el operando esta en memoria en una posición apuntada por un registro al que se le añade un determinado desplazamiento

```
MOV AX, [BP] + 2; Almacena en AX el contenido de la  
dirección de memoria que se calcula al sumar 2 al  
contenido de BP
```

Lenguaje ensamblador

Direccionamiento por registro base

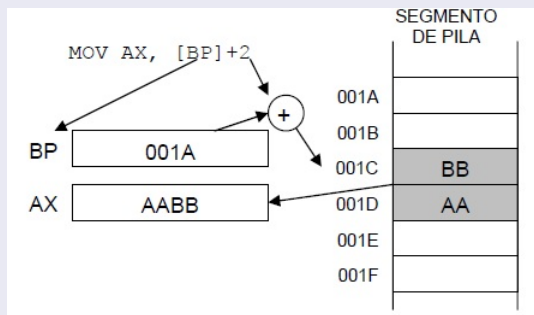


Figura 13: Lenguaje ensamblador

Lenguaje ensamblador

Direccionamiento indexado

Cuando el operando es obtenido como la suma de un desplazamiento más el contenido de un registro

```
MOV AX, TABLA[DI]; Almacena en AX el contenido de la  
dirección de memoria que se calcula al sumar TABLA al  
contenido de DI
```

Lenguaje ensamblador

Direccionamiento indexado

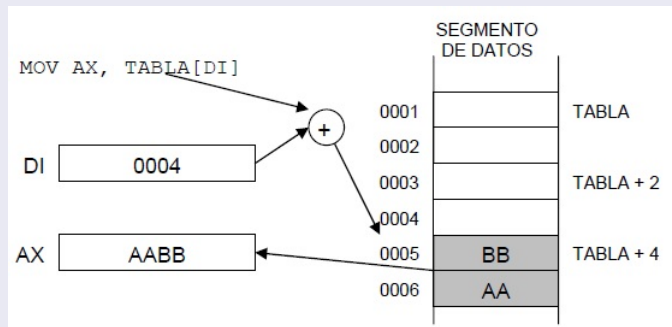


Figura 14: Lenguaje ensamblador

Lenguaje ensamblador

Direccionamiento indexado respecto a una base

Cuando el operando es obtenido como la suma de un desplazamiento más el contenido de un registro

```
MOV AX, TABLA[BX][DI]; Almacena en AX el contenido de la  
dirección de memoria que se calcula al sumar TABLA al  
contenido de DI más el contenido del registro BX
```


Contenido

1 Modos de direccionamiento

2 Lenguaje ensamblador

- Modelo procesador 8086/8088
- Generalidades
- Direccionamiento en ensamblador
- Juegos de instrucciones
- Etiquetas, comentarios y directivas

Lenguaje ensamblador

Juegos de instrucciones

Las instrucciones del 8086/8088 se dividen en:

1. Instrucciones de transferencia de datos
2. Instrucciones aritméticas
3. Instrucciones lógicas
4. Instrucciones de desplazamiento
5. Instrucciones E/S
6. Instrucciones de control de flujo

Lenguaje ensamblador

Instrucciones de transferencia de datos

1. **MOV:** Realiza la transferencia de datos de origen a destino
2. **XCHG:** Realiza el intercambio entre los valores de los operandos
3. **LEA:** Carga en un registro la dirección efectiva especificada
4. **PUSH POP:** Realizan las operaciones de apilado y desapilado en la pila del procesador

Lenguaje ensamblador

Instrucciones aritméticas

1. **ADD:** Realiza la suma
2. **SUB:** Realiza la resta
3. **NEG:** Realiza la negación de un operando
4. **MUL:** Realiza la multiplicación, para 8 bits guarda el resultado en AX, para 16 bits, guarda en la combinación DX:AX
5. **DIV:** Realiza la división sin signo, para 8 bits guarda el cociente en AL y el resto en AH, para 16 bits guarda el cociente en AX y el resto en DX

Lenguaje ensamblador

Instrucciones lógicas

1. **OR, XOR AND:** Realiza operaciones lógicas
2. **NOT:** Operación lógica de complemento

Lenguaje ensamblador

Instrucciones de comparación

1. **CMP:** Realiza la resta de dos operandos, pero no afecta a ninguno. Esta afecta a las banderas del procesador, lo que se verá más adelante.

Lenguaje ensamblador

Instrucciones de desplazamiento

1. **SAL/SHL:** Introduce un 0 al final del registro y desplaza hacia la izquierda el resto de bits. El primer bit es almacenado en el registro de estado.
2. **SAR:** Desplazamiento de la derecha

Lenguaje ensamblador

Instrucciones de E/S

Se utilizan para comunicación con periféricos

1. **IN:** Leer un puerto.
2. **OUT:** Escribir en un puerto

Lenguaje ensamblador

Instrucciones de control de flujo

Se utilizan después de ejecutar el comando **CMP**

1. **JMP:** Permite saltar a otras partes del código. Esta no requiere la ejecución de CMP.
2. **JE/JZ:** Indica que los dos operandos son iguales
3. **JA** Si el primer operando es mayor que el segundo
4. **JB** Si el primer operando es menor que el segundo
5. **JNZ:** Si los dos operandos son diferentes

Lenguaje ensamblador

Instrucciones de control de flujo

1. **LOOP:** Se utiliza para realizar bucles repetitivos, se utiliza el registro **CX** para indicar el número de veces que se realiza el ciclo, este debe apuntar a una estructura

Contenido

1 Modos de direccionamiento

2 Lenguaje ensamblador

- Modelo procesador 8086/8088
- Generalidades
- Direccionamiento en ensamblador
- Juegos de instrucciones
- Etiquetas, comentarios y directivas

Lenguaje ensamblador

Etiquetas, comentarios y directivas

1. Una etiqueta da el nombre a una instrucción y esto permite hacer referencia a ella

```
INICIO: MOV CX, DI ; inicia el contador
```

2. Los comentarios son anteceditos por ;

Lenguaje ensamblador

Etiquetas, comentarios y directivas

Una directiva sirve para definir segmentos

1. **.MODEL** Se utilizan en las directivas simplificadas:
 - **TINY** Para programas solo segmentos datos y código
 - **SMALL** Para programas son solo segmento de datos (64K memoria) y otro de código (64k)
 - **LARGE** Para programas segmento de datos y código (1MB para cada uno)
 - **MEDIUM:** Varios segmentos de código y varios de datos
 - **COMPACT:** 1 segmento de código y varios de datos
2. **.STACK n** Define el tamaño de la pila, por defecto 1KB
3. **.DATA** Abre segmento de datos
4. **.CODE** Abre segmento de código

Lenguaje ensamblador

Etiquetas, comentarios y directivas

Una directiva sirve para definir segmentos

1. **EQU:** Define una directiva

```
CONSTANTE EQU 1020; Constante toma el valor 1023
```

2. DB, DW y DD se utilizan para definir el tamaño de las variables en memoria, DB byte, DW tamaño word y DD tamaño DWORD

```
MENSAJE DB 'Este es un mensaje' ; Reserva una  
          constante de tamaño byte para el mensaje  
PESO DW ? ; Reserva una variable tamaño DWord, pero  
          no tiene valor debido a ?
```

Lenguaje ensamblador

```
.MODEL SMALL
.STACK 100H
.DATA
    max EQU 100
    cad DB max DUP ?
    dac DB max DUP ?
.CODE
    MOV AX, @DATA
    MOV DS, AX
END
```

¿Preguntas?

Siguiente clase:
Repertorio de Instrucciones:
Estructura y funcionamiento del procesador