**Guía de estudio relacionada con el Capitulo 17 / Practica / Ejercicios de Final**

**Resumen de tipos de Instrucciones**

**Instrucciones de Transferencia**

Permiten el movimiento entre distintos dispositivos de almacenamiento **Reg a Mem:** PUSH CX / **Reg a Pila:** PUSH [01AE] / **Mem a Mem:** MOV [0050],[0302] / **Mem a Pila :** PUSH [0204] / **Mem a Reg:** MOV AL,[0567] / **Reg a Reg:** MOV EAX,EBX **/ Pila a Reg:** POP DX **/Pila a Mem:** POP [859F]. Es necesario especificar: Tipo de movimiento/Dirección de operandos fuente y destino/Tamaño de datos a mover (byte, palabra, doble palabra,...)/cantidad de elementos a mover (para movimientos múltiples)

**Instrucciones Lógicas**

Permiten realizar operaciones lógicas, bit a bit, entre los operandos o manipular un bit del operando. Es necesario especificar:

Tipo de operación (AND, OR, NOT, Bit Clear, Bit Set, etc.). Tamaño de datos sobre los que se opera (byte, palabra, doble palabra. Dirección de operandos fuente y destino (0, 1, 2 ó 3, según el repertorio)

**Instrucción de salto por interrupción**

Ésta instrucción permite convocar los servicios de atención de interrupción provistos por el sistema operativo o en algunos sistemas digitales por la BIOS con sólo indicar el número de servicio luego del mnemónico INT, este número o identificación está indicado en la tabla de interrupciones del Manual de la CPU, del Manual del Sistema Operativo o del Manual de la BIOS, según sea el caso. De este modo un programa puede utilizarlos cómo parte de la definición de su ninterrupciones programadas y no son asíncronas.

|  |  |
| --- | --- |
| Lamado al Servicio de interrupción | INT número |
| Retorno del Servicio de interrupción | IRET |

**Instrucciones Aritméticas**

Permiten realizar operaciones de tipo aritmético. Es necesario especificar:

Tipo de operación (suma, resta, multiplicación, división, etc.)

Tipo de operandos y de aritmética (con signo, sin signo, entera, punto flotante, BCD, ...)

Tamaño de datos sobre los que se opera

Dirección de operandos fuente y destino (0, 1, 2 ó 3, según el repertorio)

En el siguiente cuadro se incluyen las instrucciones de desplazamiento y rotación de tipo aritmético. En todos los casos las instrucciones de desplazamiento de bits permiten desplazar o rotar un operando a la derecha o la izquierda un número determinado de bits. Es necesario especificar:

Tipo de operación (desplazamiento izda. o derecha., rotación izda. o derecha., )

Tamaño de datos sobre los que se opera (byte, palabra, doble palabra, ...)

Dirección del operando

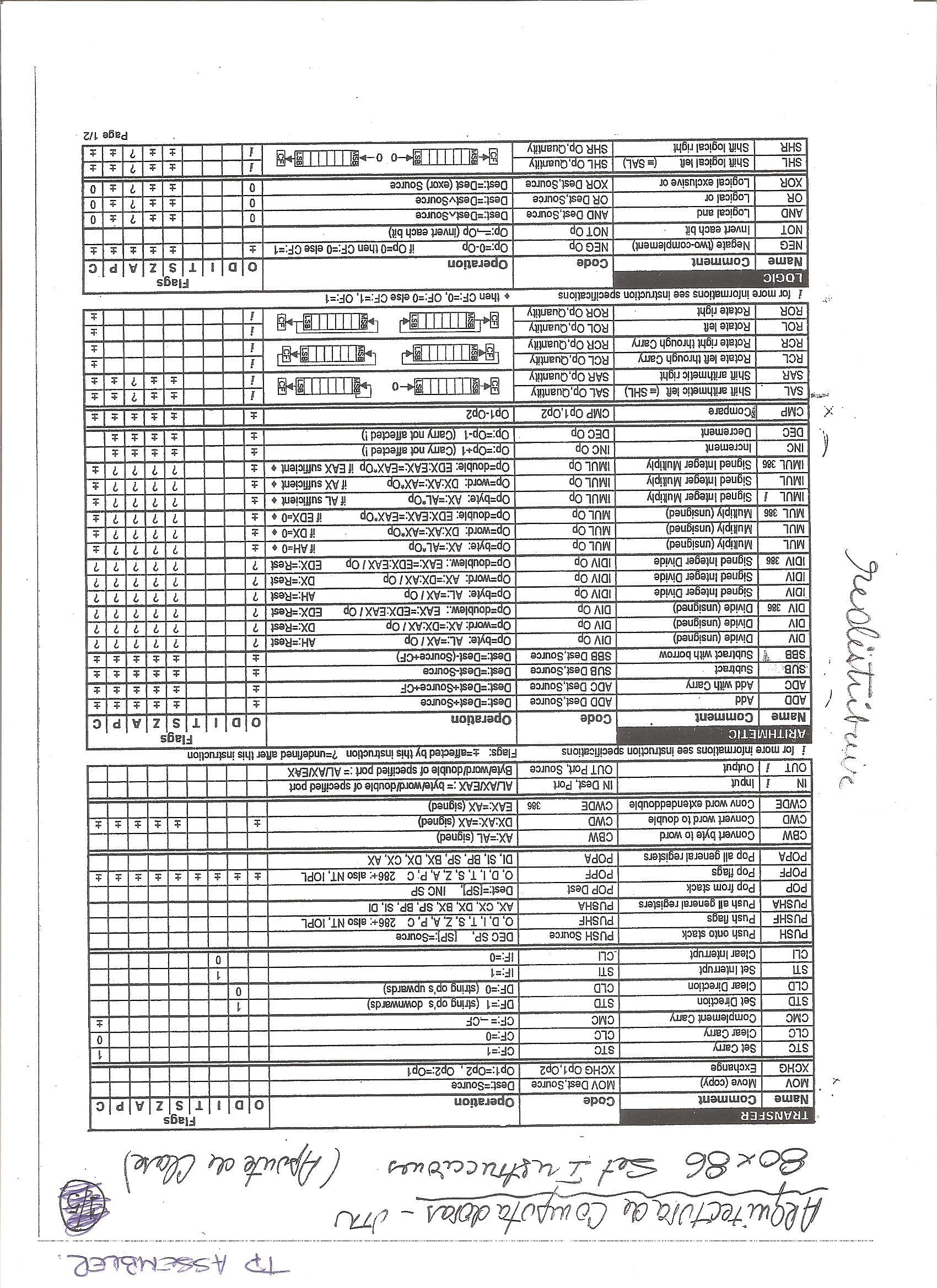
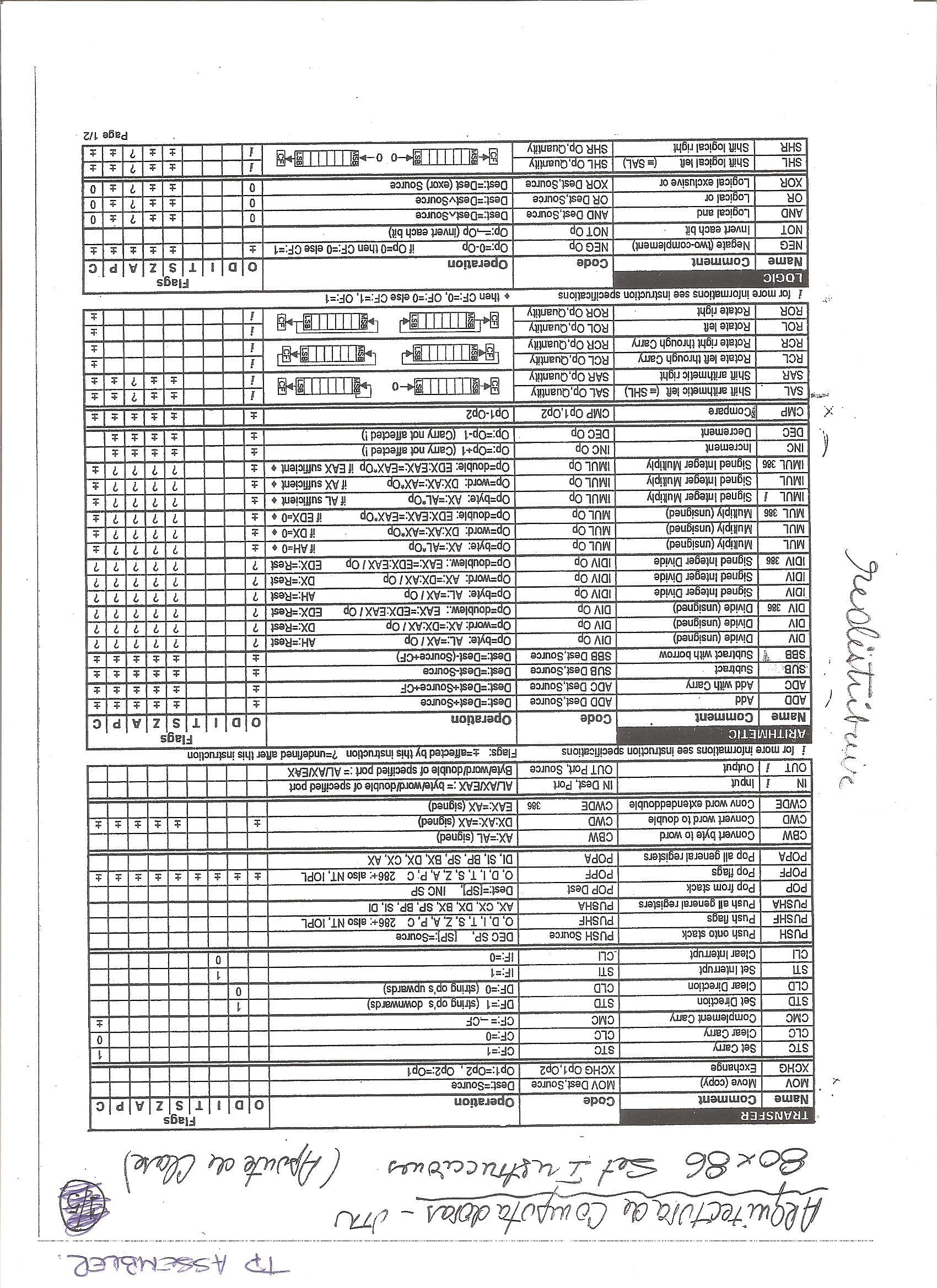
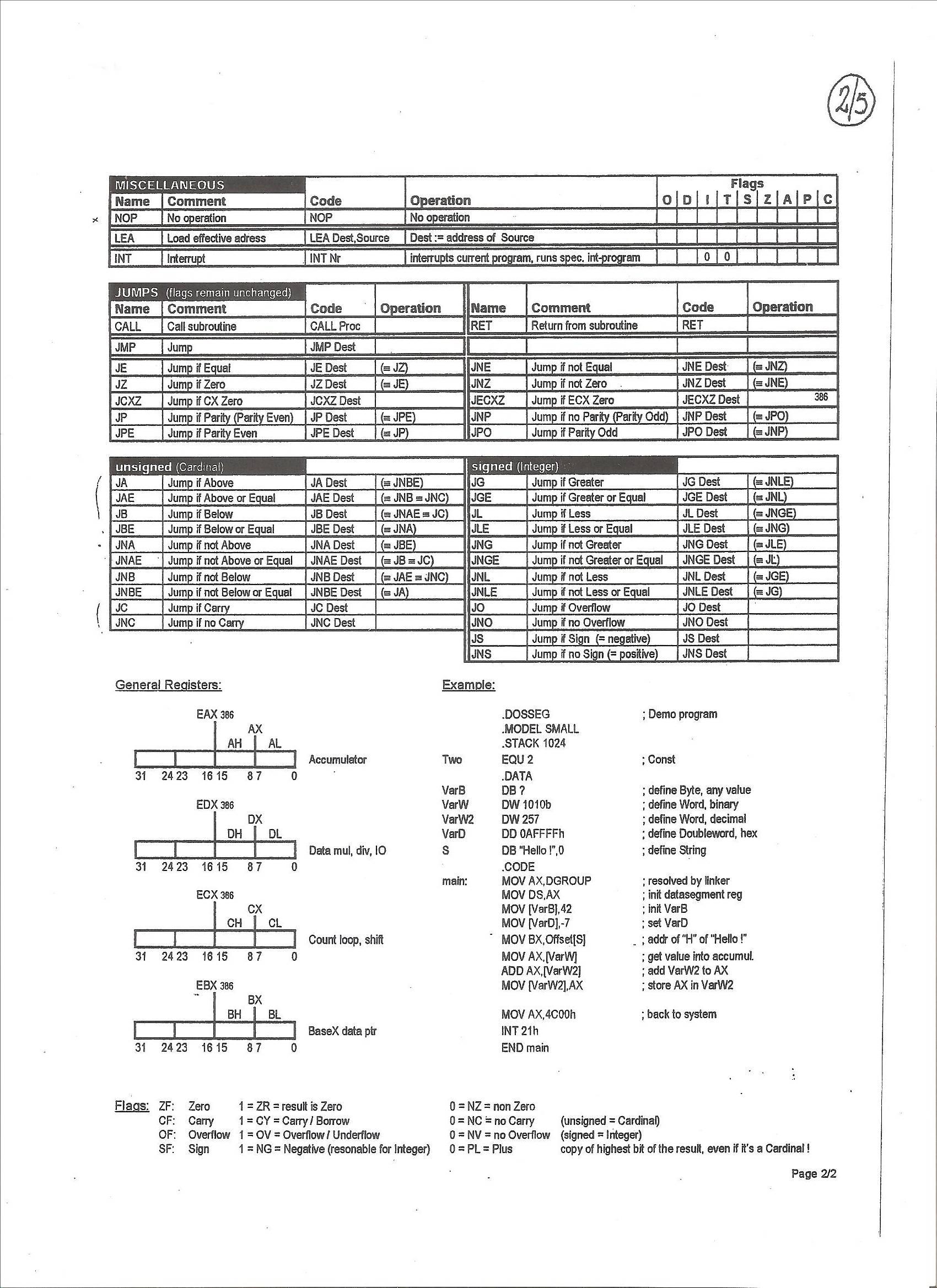
Nº de bits a desplazar o rotar

**Instrucciones de Salto Incondicional y Condicionado al estado de las banderas**

Permiten romper la secuencia normal de ejecución y saltar a una determinada dirección especificada en la instrucción o implícita. Es necesario especificar la dirección de siguiente instrucción a ejecutar si el salto es explícito.

Las instrucciones de bifurcación o salto condicional, producen el salto o no en función de una condición especificada. Esta condición se establece a partir del estado de los bits de condición (banderas) del registro de estado, que se actualizan después de la ejecución de una instrucción. El salto se produce si se cumple la condición es decir si su valor binario es 1.

Breve Resumen de Instrucciones en simbolico de màquina



**Guía y Trabajo Práctico**

1. ¿Qué es una instrucción Assembler?

Una instrucción assembler es una acción representada en código simbolico de bajo nivel

1. ¿En qué código binario se puede representar cada instrucción antes de ser compilada?

En ASCII

1. La instrucción assembler INC DH traducida a código de máquina es FEC2 represéntela en hexadecimal en el código indicado arriba antes de su compilación.
2. ¿Cómo se denomina cada uno de los campos de esta instrucción assembler? imprimo: mov ah, 2

Mov es el código de operación AH es el destino y 2 es el dato (origen)

1. ¿Qué es el Código de Operación?
2. ¿Que representa el operando origen o fuente?   
   Representa desde donde se origina o proviene el dato
3. ¿Que representa el operando destino?

Representa donde se va a guardar o trasladar el dato final

1. ¿Cómo se almacena un dato multibyte en big-endian y en little-endian, ejemplifique?  
   En big-endian se almacena de manera que el byte mas significativo quede primero y vaya descendiendo, en cambio el Little endian se ordena del byte menos significativo al mas

FE 2A E2 big-endian

E2 2A FE Little endian

1. ¿Cuáles son los elementos constitutivos de una instrucción de máquina y cuál es el significado de c/u?

Una instrucción de maquina está compuesta por el campo opcode y el data

1. Dibuje y explique el diagrama de estados de un ciclo de instrucción:
2. ¿Cómo se puede clasificar los tipos de instrucciones? Represente en código assembler una instrucción de cada tipo y explique que provoca su ejecución. Ejemplifique para el set de instrucciones x86-64, MIPS, y ARM (en el final o en el parcial solo se toman como referencia las presentadas en el Manual de Microprocesadores de 32 bits)

Instrucciones de:  
Lógicas, transferencia, aritméticas, salto por interrupción, salto incondicional, salto condicional

**Ejemplo**

**Instrucción de transferencia de datos**

|  |  |  |
| --- | --- | --- |
| **ISA x86-64** | **MIPS** | **ARM** |
| **MOV RAX, RBX**  **Carga el contenido del Registro de Cálculo RBX en el Registro RAX** | **lui $1,-7**  **Carga el valor -7 en el registro $1** | **Ldr R0, [R1]**  **Carga en el registro R0 el contenido de la dirección de RAM apuntada por R1** |

1. ¿Cuáles son los aspectos de diseño del repertorio de instrucciones de un procesador? x
2. ¿Qué son las instrucciones de llamada/retorno y para que se utilizan?

CALL y Ret son instrucciones que sirven para invocar y dar el retorno a un procedimiento o subrutina.

1. ¿Cuáles son y que significan los códigos de condición?
2. ¿Qué es una pila y como se implementa?
3. ¿Cuáles son las operaciones básicas de una pila?

Push y POP que consiste en insertar y retirar elementos

1. ¿Qué es el Puntero de Pila, el Puntero Base de la pila y el Límite de la pila?
2. Describa brevemente que es el formato y la longitud de una instrucción en el Manual de Microprocesadores de 32 bits Angulo
3. Describa como es el formato de una instrucción en el Manual de Microprocesadores de 32 bits Angulo
4. Utilice un buscador para encontrar una respuesta formal a: ¿Cuál es la diferencia entre el set de instrucciones x86, el set de instrucciones x86-64 y el set de instrucciones IA-64? ¿Cuál de ellos utiliza un procesador core I5 de Intel?
5. Indique brevemente los 4 tipos de datos presenta el capítulo en el Manual de Microprocesadores de 32 bits Angulo
6. Indique brevemente los 3 grupos en los cuales se clasifican los 7 tipos de datos que pueden manejar el coprocesador matemático (p320/321)
7. Indique brevemente las 3 categorías en que se agrupan las formas de referenciar los operandos. (p321/322/323)
8. Si AX = 3330 y DX = 4879, al ejecutarse ADD AX, DX como queda AX. Justifique (p325)
9. Si AX = 99FF y DX = 4879, al ejecutarse SUB AX, DX como queda AX. Justifique (p326)
10. Si AX = 99FF y DX = 4879, al ejecutarse CMP AX, DX como queda el reg. EFLAGS. Justifique (p327)
11. Explique brevemente que hacen las instrucciones aritmética DEC, INC, MUL e IMUL (p326)
12. Si AX = 3330 y DX = 4879, al ejecutarse AND AX, DX como queda el reg. EFLAGS. Justifique (p328)
13. Si AX = AFE4, al ejecutarse ROL AX, 1 como queda AX. Justifique (p329)
14. Explique brevemente que hacen las instrucciones TEST , SAL y SAR (p329)
15. Si AX = 3330 y DX = 4879, al ejecutarse SHLD AX, DX, 2 como queda AX y DX. Justifique (p330)
16. Explique cómo queda el registro IP al ejecutarse la instrucción JMP FF456781. Justifique (p332)
17. Explique brevemente que hacen las instrucciones de salto JA, JC y JE (p333)
18. Explique brevemente que hacen las instrucciones de transferencia de datos MOV, IN, POP y PUSH (p335)
19. Explique brevemente que hacen las instrucciones de control de señalizadores CLC y CLI (p335/336)
20. Explique brevemente que hacen las instrucciones CALL, RET, INT e IRET (p340)
21. Explique brevemente que hace la instrucción del coprocesador WAIT (p342)
22. Explique brevemente que hace y para que se usa la instrucción NOP (p335/336)

**Ejercicio**

Indicar en cada línea de código el valor del registro o posición de memoria del operando destino

**mov ax, a7de**; *carga en ax el valor del primer operando*

**mov bx, 566c**; *carga en bx el valor del segundo operando*

**and ax, bx**; *realiza la operación* ***and*** *con salida en ax*

**mov [0200], ax;** *guarda el valor existente en ax en la posición de memoria [0200]*

**mov ax, a7de;** *restaura en ax el valor del primer operando*

**and bx, ax;** *realiza la operación* ***and*** *con salida en bx*

**mov [0202], bx;** *guarda el valor existente en bx en la posición de memoria [0202]*

**mov bx, 566c;** *restaura en bx el valor del segundo operando*

**or ax, bx;** *realiza la operación* ***or*** *con salida en ax*

**mov [0204], ax;** *guarda el valor existente en ax en la posición de memoria [0204]*

**mov ax, a7de;** *restaura en ax el valor del primer operando*

**or bx, ax;** *realiza la operación* ***or*** *con salida en bx*

**mov [0206], bx;** *guarda el valor existente en bx en la posición de memoria [0206]*

**mov ax, 566c;** *restaura en bx el valor del segundo operando*

**xor ax, bx;** *realiza la operación* ***xor*** *con salida en ax*

**mov [0208], ax;** *guarda el valor existente en ax en la posición de memoria [0208]*

**mov ax, a7de;** *restaura en ax el valor del primer operando*

**xor bx, ax;** *realiza la operación* ***xor*** *con salida en bx*

**mov [0210], bx;** *guarda el valor existente en bx en la posición de memoria [0210]*

**not bx;** *realiza la operación de* ***negación*** *de bx, con salida en bx*

**mov [0212], bx;** *guarda el valor existente en bx en la posición de memoria [0212]*

**Ejercicios Típicos de Finales**

Dado el siguiente código, conteste las preguntas

Direccion Segmentada Còdigo de Màquina Simbòlico de Màquina

0CED:0100 8A260002 MOV AH,[0200]

0CED:0104 8A260102 MOV BH,[0201]

0CED:0108 00FC ADD AH,BH

0CED:010A 88260202 MOV [0202],AH

Referido a la primera instrucción assembler

1. ¿Cuál es la dirección segmentada que apunta al primer byte en donde esta instrucción esta almacenada?
2. ¿Cuántos bytes ocupa esta línea de código assembler en memoria? 13 (caracteres)
3. ¿Cuántos bytes ocupa la instrucción en código de maquina? 4 (cada 2 digitos)
4. ¿Qué significan los corchetes y que significa el numero 0200? Los corchetes son contenido de y los 0200 son el desplazamiento
5. ¿Cómo clasificaría esta instrucción? De transferencia
6. Indique el nombre del registro destino en su formato de 16 y de 32 bits. AX Y EAX
7. Deduzca y exprese en hexadecimal el código de operación de esta instrucción.8A26
8. ¿Por qué los valores del campo data de la instrucción expresada en assembler y de la instrucción expresada en código de maquina están invertidos de a byte? PORQUE ESTAN EL LITTLE ENDIAN

Comparando las instrucciones 1 y 3 indique el motivo por el cual cambia su código de operación CAMBIA PORQUE UNO ES MOV Y OTRO ES ADD, NO TIENE CAMPO DATA XQ SON REGISTROS

Comparando las instrucciones 1 y 4 indique el motivo por el cual cambia su código de operación SON 2 MOVS, CAMBIA PORQUE CAMBIA DESTINO Y ORIGEN EN EL 1 ES DESDE MEMOPRIA A REGISTRO Y EL OTRO DESDE REGISTRO A MEMORIA 1 ES LECTURA Y EL 4 ESCRITURA

Si tuviese que comprobar el resultado almacenado en [0202] ¿qué dirección o direcciones segmentada/s examinaría la memoria?

Data segment: 0202(dirección segmentada única para leer el valor almacenado)

-------------------------------------------------------------------------------------------------------------------

Para los siguientes registros visibles de un procesador

AX=0800 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=0CD7 ES=0CD7 SS=0CD7 CS=0CD7 IP=0102

indiqué los registros de cálculo (AX BX CX DX), registros punteros (SI DI) y registros de segmento (DS ES SS CS).

Para cada registro de segmento indique en lugar se aloja el respectivo desplazamiento

(Código, datos y de Stack)

¿Por qué digo que la máxima longitud de un segmento es de 64 K cuando observo una dirección segmentada como las presentadas?

0234:457D 0234:0000 0234: FFFF

: FFFF [0000; FFFF] 0;65535 65536 bytes 64K

Exprese en hexadecimal el desplazamiento asociado al primer y al último byte de un segmento de este tipo

Expréselo en binario.

Expréselo en potencias de dos. 0;216-1

-----------------------------------------------------------------------------------------------------------------

¿Cuántos bits debe tener el desplazamiento para permitir el acceso al último byte de un segmento de 4G? 232 32 bits EIP ESP EDATA

¿A qué tipo de segmento le asignaría usted un derecho de acceso “no write”? código

-----------------------------------------------------------------------------------------------------------------

Para el siguiente algoritmo expresado en pseudocódigo indique para cada línea el TIPO de instrucción, la instrucción y el modo de direccionamiento.

Si el contenido inicial de la variable var1 es 20 ¿Cuál es su valor final?

Comienzo:

1. mueva el valor 10 a un registro llamado r1. MOV r1, 10 (inmediato) (de transferencia) r1=10
2. mueva el contenido de la variable var1 a otro registro llamado r2 MOV r2, [0200] r2=20

Calcular:

3. aumente el valor del registro r2 en una unidad INC r2 (implícito) r2=21

4. disminuya el valor de r1 en una unidad DEC r1 r1=0F

5. Si el valor de r1 es igual a cero CMP r1, 0 JZ Calcular

entonces vaya a Calcular

sino, siga con la próxima pseudoinstrucción

6. Mueva el contenido de r2 a la variable var1 MOV [0200], r2

------------------------------------------------------------------------------------------------------------------

Para la línea de código presentada a continuación y considerando el cálculo de la dirección física como Registro de Segmento \* 10 + Desplazamiento

3BB0:0103 MOV AX, [35B0]

¿Si el valor del DS es AB55 cuál es la dirección física del operando involucrado en ésta instrucción? DS: DATA AB55\*10+ 35B0

Si el valor del SS es 3700 y el valor del SP es FFFE cuál será la dirección segmentada asociada al byte vacío de la pila luego de una instrucción PUSH ss:sp 3700:FFFC

Si la instrucción siguiente a la descripta se aloja en la dirección física 3BC07, ¿Cuál es el tamaño de la instrucción y cuantos bytes le corresponden al código de operación?

3BB0\*10+ IP (0103) = 3BC03, el tamaño de la instrucción es 4 bytes

El código de la operación por descarte es 4 bytes

Para el siguiente código en Assembler responder a las siguientes preguntas, teniendo en cuenta el estado inicial de los registros que también se muestra a continuación.

1. ¿Cuál es el tamaño total del código de este programa en la memoria?
2. ¿Qué valor toman los registros AX y BX luego de la ejecución de las instrucciones:

MOV     AX, [0200]

MOV     BX,0030   BX=0030

1. Para la instrucción MOV [0300], AX calcular la dirección efectiva de la instrucción en la memoria
2. ¿En qué instrucciones dentro de este código hay acceso a la memoria?
3. Para la instrucción MOV BX,0030 indicar el tamaño total de la instrucción en memoria ¿Cuántos bits corresponden a la parte DATA, y al COP (OpCode)?

Estado inicial de los registros:

AX=10A1  BX=0F48  CX=95CD  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000

DS=0C98  ES=0C98  SS=0C98  CS=0C98  IP=0100   NV UP EI PL NZ NA PO NC

Código Assembler:

0C98:0100 90                 NOP

0C98:0101 A10002        MOV     AX,[0200]

0C98:0104 BB3000        MOV     BX,0030

0C98:0107 01D8            ADD     AX,BX

0C98:0109 A30003       MOV     [0300],AX

0C98:010C CD20            INT     20

Observe el contenido de los siguientes registros de CPU y responda las consignas.

CX=000A BX=0B01 IP=0008 CS=0032 SS=0090 SP=4532

¿Cuál es el valor del registro destino luego de la ejecución de una instrucción

NOT CH, [FF08]

¿Si este es el estado de los registros de CPU en un momento dado, cuál es el valor de la dirección segmentada de la instrucción que se está ejecutando?

Si consideramos datos enteros signados para BX y CX, ¿Cuál es el valor final de BX luego de la ejecución de la instrucción SUB BX,FFFF?

Considerando el valor actual del registro CX indique cuantas veces la instrucción LOOP decrementa el registro hasta que se sale de la estructura repetitiva.