ASSEMBLY TO MACHINE

Date_____

steps:

1. first find the opcode of the given instruction
2. find d and w, if required in the opcode.
3. find mode, by checking the given instruction
4. find address of mem/reg in the predetermined mode column.
5. write full binary code.
6. convert binary to hexadecimal.

NOTE:

- If the instruction contains the register as both the source and destination, then source and destination will swap while writing binary code.

- If 16-bit hexadecimal displacement, then swap first 8-bits by the next 8-bits. Example: ABCD → CDAB

1- mov bl, al 100010 dw oo rrrmmm + disp

opcode: 100010
d : source = register : 0
w : 0
mod : 11 (reg to reg)
reg : al = 000
bl = 011

1000 1000 1100 0011 = $(88C3)_h$

d: 0 : source = register; 1 : source = memory

w: 0 : instruction 8bit; 1 : instruction 16bit

mode:
- 11 : register to register
- 00 : register/memory (without displacement)
- 01 : register/memory (with 8-bit displacement)
- 10 : register/memory (with 16-bit displacement)

Signature_____ No._____

Date

2. mov ch, bl 100010 dw oo rrrmmm + disp

opcode : 100010
d : 0
w : 0
bl : 011
ch : 101
mode : 11

1000 1000 1101 1101
$= (88DD)_h$

3. mov ax, cx 100010 dw oo rrrmmm + disp

opcode : 100010
d : 0
w : 1
mode : 11
ax : 000
cx : 001

10001001 1100 1000
$= (89C8)_h$

4. mov cx, 0ABEh 1011 w rrr data → reg, imm

opcode : 1011
w : 1
cx : 001

1011 1001 BE0A
$(B9BE0A)_h$

5. mov var 1, dx 100010 dw oo rrrmmm + disp

opcode : 100010
d : 0
w : 1
mode : 00
dx : 010
var 1 : direct address = 110

1000 1001 0001 0110
$= (8916)_h$

reg/mem Date

6- mov ax, [bx] 100010 dw oorrr mmm + disp

opcode : 100010
d : 1
w : 1 1000 1011 0000 0111
mode : 00 = $(8B07)_h$
ax : 000
[bx] : 111

reg, mem →

7. sub bx, [di] 001010 dw oo rrr mmm + disp

opcode : 001010
d : 1
w : 1 0010 1011 0001 1101
mode : 00 = $(2B1D)_h$
bx : 011
[di] : 101

mem, reg →

8. add [bx][di] +1234h, ax 000000 dw oorrr mmm + disp

opcode : 000000
d : 0
w : 1 0000 0001 1000 0001
mode : 10 = $(0181)_h$ + disp
[bx]+[di]+D_{16} = 001
ax : 000 = $(01813412)_h$

mem, reg →

9. add OFABE [bx][di], dx 000000 dw oorrrmmm + disp

opcode : 000000
d : reg : 0
w : 1 → 16 bit instruction
mode : 10
[bx]+[di]+D_{16} : 001
dx : 010

0000 0001 1001 0001
= $(0191BEFA)_h$

Signature No.

reg, mem ✓ Date

10- add ax, [si] 000000 dw oorrrmmm + disp

opcode : 000000

d : 1 → source = memory

w : 1 → 16-bit instruction

mode : 00 → reg/mem with no displacement

ax : 000

[si] : 100

0000 0011 0000 0100

$= (0304)_h$

MACHINE TO ASSEMBLY

Date _____

NOTE:

- If bits are more than 16 then the additional bits are displacement.

1. 8B 6F FE h → displacement of 8 bits
   (8B 6F: 16 bits)

mov opcode:
100010 dw oorrrmmm +disp

8B = 1000 1011
6F = 0110 1111
(0110 1111 = bp)

mov opcode: 100010
- d : 1 → source : memory
- w : 1 → 16 bit instruction
- mode : 01

reg : bp : 101
mem : [bx] + D8 : 111

mov bp, [bx + FE]

~~mov [BX + FE],~~

2. 89 A4 EA 01 h → displacement : 16 bits
   (89 A4: 16 bits)

mov opcode
100010 dw oorrrmmm +disp

89 : 1000 1001
A4 : 1010 0100

mov opcode : 100010
- d : 0 → source = register
- w : 1 → 16 bit instruction
- mode : 10 → reg/mem with D_{16}
- 101 : bp
- 100 : [si] + D_{16}

mov [si + 01EA], bp

Signature _____ No. _____

Date

3. F6 DBh — so, no displacement

16 bit

F6 : 1111 0110
DB : 1101 1011 > NEG INSTRUCTION

Now, match with the opcodes and find the instruction.

neg opcode : 1111011 w 00011 mmm + disp

w : 0 → 8 bit instruction
mode : 11 → reg/reg
reg → 011 : bl (from mode)

neg bl

4. 56h — no displacement

56 : 0101 0110 ⟶ PUSH

Now, match with the opcodes in the table

push opcode : 01010 rrr

reg → 110 : si

push si

can't be dh because push can be used either with 16 bit or 32 bit register