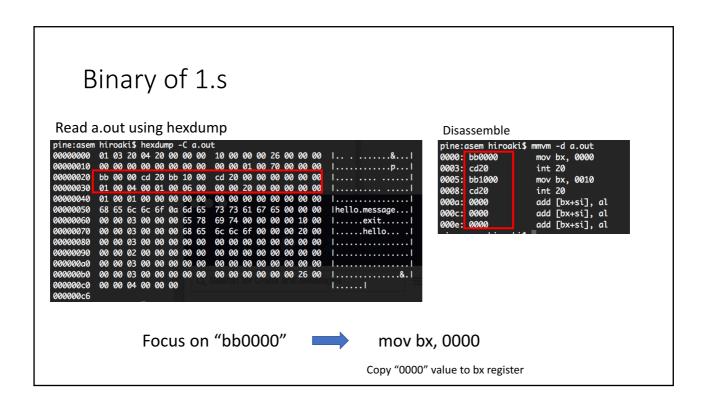
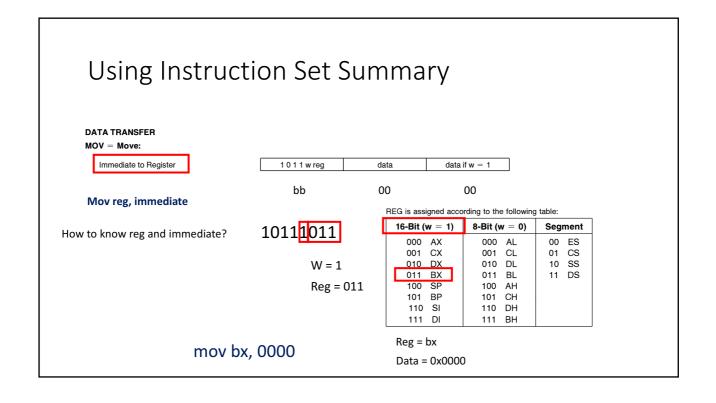
Advanced Operating System and Virtualization

Disassember1 Hiroaki Fukuda

Contents

- Understand the binary of 1.s
- Understand specifications of 8086
 - mod
 - r/m
 - reg
 - d
 - effective address





Compile 1.c and try first 3 instructions

```
mod: reg: r/m: d
```

- Mod
 - 2 bit information to decide displacement(DISP)
- Reg
 - 3 bit information to decide register
- r/m
 - 3 bit information to decide effective address (EA)
- d
 - 1 bit information to decide the direction (from/to)

Let's try 31ed

XOR = Exclusive or:

Reg./Memory and Register to Either

| 001100dw | mod rog r/m |
|----------|-------------|
| ooiiooaw | mod reg r/m |

00110001 11101101

d = 0 w = 1 mod = 11 reg = 101 r/m = 101

if d = 1 then "to" reg; if d = 0 then "from" reg

if mod = 11 then r/m is treated as a REG field

XOR r/m, reg

REG is assigned according to the following table:

| 16-Bit (w = 1) | 8-Bit (w = 0) | Segment |
|----------------|---------------|---------|
| 000 AX | 000 AL | 00 ES |
| 001 CX | 001 CL | 01 CS |
| 010 DX | 010 DL | 10 SS |
| 011 BX | 011 BL | 11 DS |
| 100 SP | 100 AH | |
| 101 BP | 101 CH | |
| 110 SI | 110 DH | |
| 111 DI | 111 BH | |

XOR bp, bp

Let's try by hand

- •89e3
 - Why mov bx, sp

Let's try 8b07

```
76543210
                                                  100010dw
             Register/Memory to/from Register
                                                               mod reg r/m
                             10001011 00000111
                           D = 1 w = 1 \mod = 00 \text{ reg} = 000 \text{ r/m} = 111
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
                                                                 mov reg, r/m
 if mod = 10 then DISP = disp-high; disp-low
 if r/m = 000 then EA = (BX) + (SI) + DISP
 if r/m = 001 then EA = (BX) + (DI) + DISP
 if r/m = 010 then EA = (BP) + (SI) + DISP
                                                              mov ax, [bx]
 if r/m = 011 then EA = (BP) + (DI) + DISP
 if r/m = 100 then EA = (SI) + DISP
 if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP
```

Let's try 1.c

- Implement your own disassembler which can analyze the binary of 1.c
- Compare the result to the outout of mmvm -d