CMPT880 Presentation

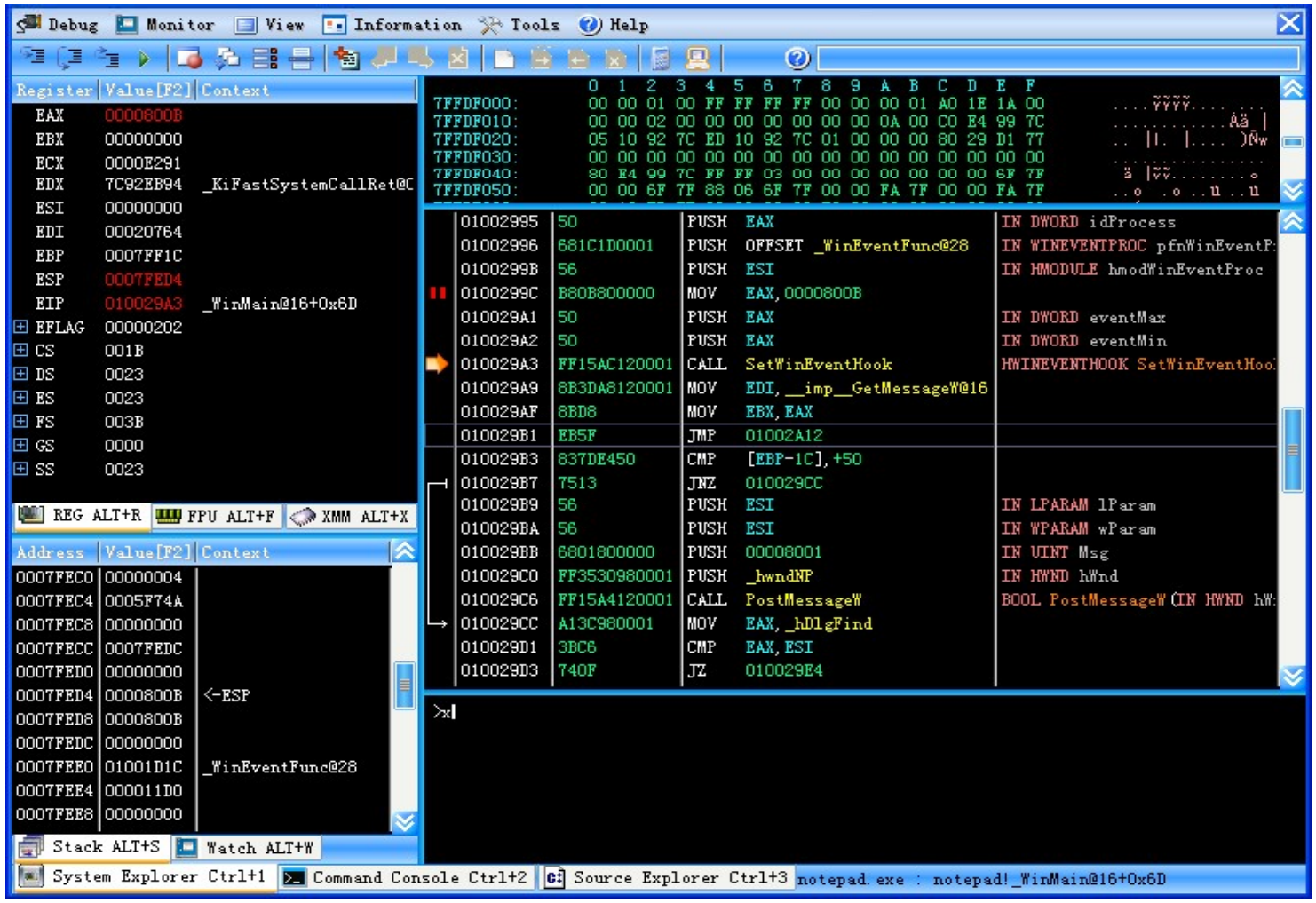
Assembly Programming and Optimization

Jian Zhang
jiz869@mail.usask.ca

- Is assembly language still useful?
- Art of assembly programming
- Resources

Is assembly language still useful?

- *Hack and Crack*

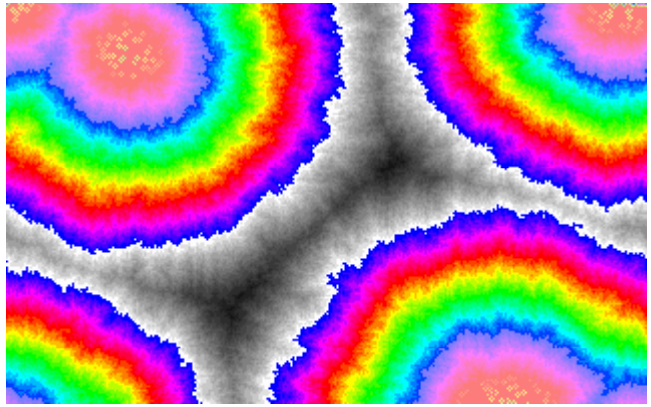


Is assembly language still useful?

- *Just for fun*

```
S equ 0E5h
org 100h
mov  al,13h
int  10h
lds  bx,[bx]
M:
cmp  [bx],cl
adc  [bx],ah
imul bx,byte S
mov  cl,[bx]
add  cl,[bx+di]
add  cl,[bx-321]
add  cl,[bx+si+63]
dec  bx
in   al,60h
dec  al
jnz  M
ret
```

http://www.pouet.net/



Is assembly language still useful?

- Operating system
  - *BOOT/BOOT LOADER of operating system*
  - *Low level MMU operation*
  - *Interrupt management and some device drivers*
- Compiler
  - *Code generator*
- Specific processors
  - *Some processors do not support development tools of high level languages.*

Is assembly language still useful?

- Optimization
  - *Whenever the high level optimization doesn't help any more*
  - *Critical subroutine*
  - *Bitwise operation*
  - *Vector instruction set*

Speed can make profit!

Art of assembly programming

- Instructions can do more than you think

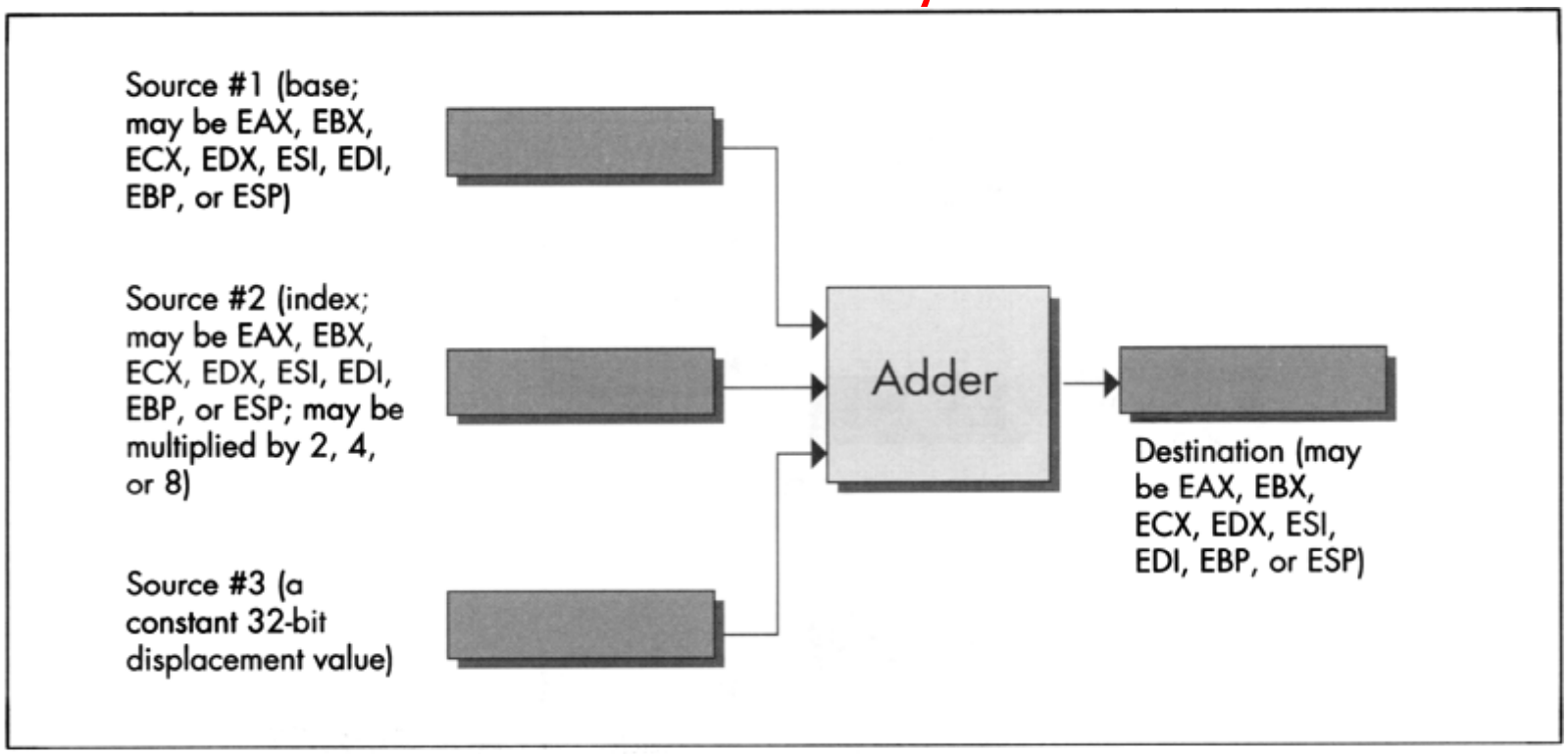
Intel 80x86 instruction set

LEA --Load Effective Address

Opcode	Instruction	Description
8D */r*	LEA *r16,m*	Store effective address for *m* in register *r16*
8D */r*	LEA *r32,m*	Store effective address for *m* in register *r32*

Art of assembly programming

- Instructions can do more than you think



Operation of the 32-bit LEA reg,[Addr].

Art of assembly programming

- Instructions can do more than you think

  So now we can do multiplication and addition with only 1 instruction:

```
lea ebx,[ebx+ebx*4]
```

```
mov edx, ebx
shl ebx, 2
sdd ebx, edx
```

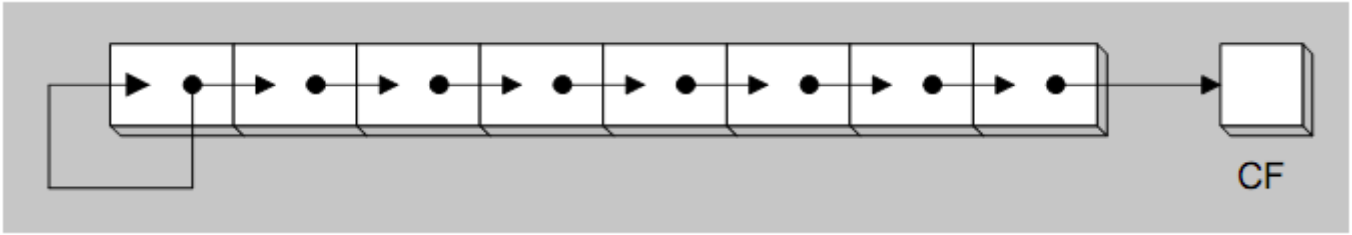
Art of assembly programming

- Instructions can do more than you think

Intel 80x86 instruction set

SAR -- Shift arithmetic right

D0 /7	SAR *r/m8*	Signed divide* *r/m8* by 2, once
D2 /7	SAR *r/m8*,CL	Signed divide* *r/m8* by 2, CL times
C0 /7 *ib*	SAR *r/m8*,*imm8*	Signed divide* *r/m8* by 2, *imm8* times



Art of assembly programming

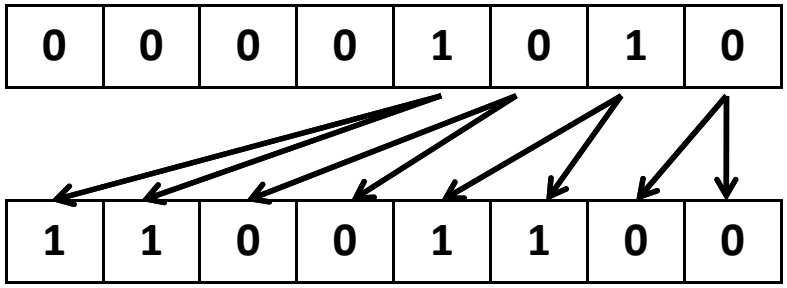
- Instructions can do more than you think

  consider sar as a bit-manipulation instruction rather than as a signed arithmetic instruction.

  The most significant bit is doubled!

  Excellent tool for bit-doubling:

```
shr bl,1
rcr ax,1
sar ax,1
```

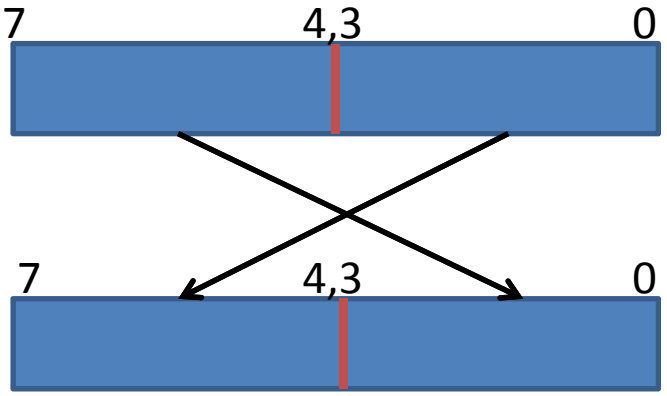


Art of assembly programming

- Do not think in high level language

  Swapping nibbles

```
b = (a >> 4)& 0x0f;
c = (a << 4)& 0xf0;
a = b|c;
```



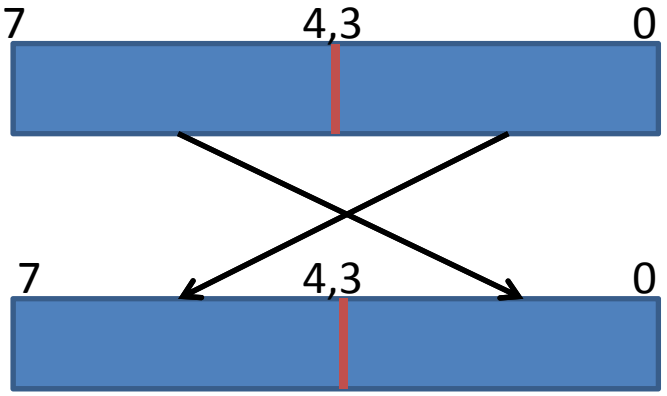

```
movb %al, %dl
shrb $4,  %dl
movb %dl, _b
salb $4,  %al
orb  %al, %dl
```

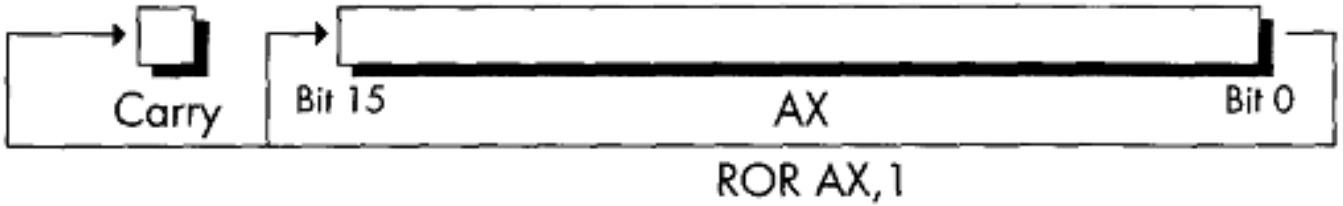
Art of assembly programming

- Do not think in high level language

  Swapping nibbles

```
movb  $4,  %cl
rorb  %cl, %al
```





Art of assembly programming

- Optimization on modern processor
    - *Out of order execution*
        - *Register rename and partial register*
        - *Micro-operations*
        - *Multiple execution units*
        - *Piplelined instructions*

Art of assembly programming

- Optimization on modern processor
    - *Memory access*
        - *Organizing data/code for improving cache*
        - *Alignment of data/code*

Resources

- ***"Zen of Assembly Language" by Michael Abrash***
- ***"Hacker's Delight" by Henry S. Warren***
- **"Code Optimization: Effective Memory Usage" by Kris Kaspersky**
- **"The Art of Computer Programming", Vol 4, by Donald E. Knuth**