

- Exam 2 Review

- SHLD - ~~the registers~~, op1, op2, value or cl registers
 ↓ must be register, not affected

shld ax, bx, 4 ax: CE48 bx: F243

high order bits of bx get shifted into low order bits of ax

ax = E48F bx = F243

overflow flag is undefined when shifting more than 1 bit

value 1 - 4 are words

value 1 = F000

value 2 = 2000

value 3 = C000

value 4 = 5000

mov ax, value 1

shld bx, ax, 4 ; xxxx F

mov ax, value 2

shld bx, ax, 4 ; xx F2

mov ax, value 3

shld bx, ax, 4 ; x F2C

mov ax, value 4

shld bx, ax, 4 ; F2C5

- SHRD $opl, opd, immediate$

shrd ax, bx, 4

ax = FCD2 bx = 43F5
 ax = 5CD2

ax = year bx = day cx = month

mov dx, ax ; dx = year

shrd dx, ax, 7

shrd dx, bx, 5

shrd dx, cx, 4

dx = dddddyyyyyyy ~~~~~

dx = mmmdddddyyyyyy ~~~~~

- Rotate instructions

rol

ror

rdl

ror

- ~~move~~ affect ~~sign~~ or ^{zero} ~~overflow~~, only carry & ~~zero~~ overflow

rol, dest, immediate or cl

rol al, 1

xxxxxxxx

if al = 11010101 then

cl = 1, al = 10101011

ror al, 1

low order bit goes into carry flag and to the high order bit

- rcl - bit shifting goes to CF

CF = 0

al = 11111111

rcl al, 1

CF = 1

al = 11111110

- rcr - rotate right through carry

CF = 0

al = 11101011

rcr al, 1

CF = 1

al = 01110101

- determine if al is even or odd

rcr al, 1

jmp isEven

rcl to rotate the value of CF & AL

- epilogue & prologue

epilogue - what we do at beginning of function

prologue - what we do at end of function

in C & C++, they clean up the stack for us

- to automatically save registers:

func proc uses ebx ecx

- if using uses, recalculate bracket notation because stack changes