Lecture 26:
Advanced Addressing Modes
Today’s Goals

- Use decrement/increment/test with branch instructions to reduce code size and/or execution time
- Use predecrement, preincrement, postdecrement, and postincrement instructions to reduce code size and/or execution time
Decrement/Increment/Test with Branch

• In many programs, we used a variable to track how many times to perform a loop.

• A loop count variable and conditional branches.
  – DEC, DECA, DECB, DEX, DEY
  – INC, INCA, INCB, INX, INY
  – BEQ, BNE,

• This is very common in programming.

• HCS12 has following instructions.
  – DBEQ reg, offset: Decrement by 1 and Branch if EQual
    • reg: A, B, D, X, Y, or SP
  – DBNE reg, offset: Decrement by 1 and Branch if Not Equal
  – IBEQ reg, offset: Increment by 1 and Branch if EEqual
  – IBNE reg, offset: Increment by 1 and Branch if Not Equal
  – TBEQ reg, offset: Test and Branch if EEqual
  – TBNE reg, offset: Test and Branch if Not Equal
Example

- The program that convert a list of 4-byte Big-Endian numbers to an array of 4-byte Little-Endian numbers.
ORG $1000

BEnd DS.W 1
LEnd DS.W 1
Length DS.W 1

ORG $2000
LDX BEnd
LDY LEnd
LDD Length
BEQ Done ; Cycls,bytes
Loop MOVB 0,X,3,Y ; 5,4
        MOVB 1,X,2,Y ; 5,4
        MOVB 2,X,1,Y ; 5,4
        MOVB 3,X,0,Y ; 5,4
        LEAX 4,X ; 2,2
        LEAY 4,Y ; 2,2
        SUBD #$0001 ; 2,3
        BNE Loop ; 3,2
Done SWI

We can save two cycles in a loop!!
Auto Pre/Post Increment/Decrement (IDX)

- Effective Address: Value supplied in index register as with indexed addressing but with no offset

- Format:
  - inc/dec value, (+/-)reg
  - inc/dec value, reg(+/-)

- Example:
  - LDD 2,X+

- Note:
  - Unlike regular indexed, the value in X, Y, SP is changed
  - Pre-version modifies BEFORE the memory access is made
  - Post-version modifies AFTER the memory access is made
  - Inc/Dec value is +/-1 to 8
  - Cannot use an inc/dec value of 0 (not make sense!)
ORG $1000

BEnd DS.W 1
LEnd DS.W 1
Length DS.W 1

ORG $2000
LDX BEnd
LDY LEnd
LEAY 3,Y
LDD Length
BEQ Done ; Cycles,bytes

Loop MOVB 1,X+,1,Y- ; 5,4
MOVB 1,X+,1,Y- ; 5,4
MOVB 1,X+,1,Y- ; 5,4
MOVB 1,X+,7,Y+ ; 5,4
DBNE D,Loop ; 3,3

Done SWI

We can save even four more cycles in a loop from the previous improved implementation!!
Questions?
Wrap-up

What we’ve learned
What to Come