Microprocessors (0630371)
Fall 2010/2011 – Lecture Notes # 19

Conditional Loop Instructions

Objectives

- **LOOPZ and LOOPE**
- **LOOPNZ and LOOPNE**

LOOPZ and LOOPE

- **Syntax**:
  - `LOOPE destination`
  - `LOOPZ destination`
- **Logic**:
  - `ECX ← ECX – 1`
  - `if ECX > 0 and ZF=1, jump to destination`
- Useful when scanning an array for the first element that does not match a given value.

LOOPNZ and LOOPNE

- **LOOPNZ (LOOPNE)** is a conditional loop instruction
- **Syntax:**

```
LOOPNZ destination
LOOPNE destination
```

- **Logic:**

```
ECX ← ECX – 1;
if ECX > 0 and ZF=0, jump to destination
```

- Useful when scanning an array for the first element that matches a given value.

LOOPNZ Example

The following code finds the first positive value in an array:

```
TITLE Scanning for a Positive Value        (Loopnz.asm)
; Scan an array for the first positive value.
; If no value is found, ESI will point to a sentinel
; value (0) stored immediately after the array.
INCLUDE Irvine32.inc
.data
array  SWORD  -3,-6,-1,-10,10,30,40,4
sentinel SWORD  0
.code
main PROC
  mov esi,OFFSET array
  mov ecx,LENGTHOF array
next:
  test WORD PTR [esi],8000h          ; test sign bit
  pushfd                             ; push flags on stack
  add  esi,TYPE array
  popfd                              ; pop flags from stack
```

```
      loopnz next                       ; continue loop
      jnz  quit                         ; none found
      sub  esi,TYPE array               ; ESI points to value
    quit:
      movsx eax,WORD PTR[esi]           ; display the value
      call WriteInt
      call crlf
      exit
    main ENDP
    END main
```

Locate the first nonzero value in the array. If none is found, let ESI point to the sentinel value:

```
        .data
        array SWORD 50 DUP(?)
        sentinel SWORD 0FFFFh
        .code
        mov esi,OFFSET array
        mov ecx,LENGTHOF array
        L1:    cmp WORD PTR [esi],0     ; check for zero
          (fill in your code here)
        quit:
```

Solution

```
        .data
        array SWORD 50 DUP(?)
        sentinel SWORD 0FFFFh
        .code
        mov esi,OFFSET array
        mov ecx,LENGTHOF array
        L1:    cmp WORD PTR [esi],0     ; check for zero
        pushfd  ; push flags on stack
        add esi,TYPE array
        popfd   ; pop flags from stack
        loope next   ; continue loop
        jz quit ; none found
        sub esi,TYPE array    ; ESI points to value
        quit:
```