Chapter 9

Subroutines and TRAPs
- Privileged Instructions
- TRAP Routines
- Subroutines

Privileged Instructions
- There are several instructions that are best executed by a supervisor program (OS) rather than a user program:
  - I/O instructions
  - Interacting with system/device (memory-mapped) registers
  - Resetting the clock
  - Halt
- i.e. instructions where one program can affect the behavior of another.
- Most modern CPUs are designed to enforce at least two modes of operation:
  - User Mode
  - Privileged Mode (aka. supervisor, kernel, monitor mode)
- Only the supervisor program (OS) can execute privileged instructions.
- But how do we ALLOW user programs to access privileged functionality?
- There are two issues to address: Policy and Mechanism

TRAP Instructions
- TRAPs insulate critical tasks from the user
  - with or without privilege enforcement
- The TRAP mechanism:
  - A set of trap service routines or TSRs (part of the CPU OS)
  - We have already seen the basic I/O SRs
  - A table of the starting addresses of these service routines
    - Located in a pre-defined block of memory ...
    - ...called the Trap Vector Table or System Control Block
    - In the LC-3 from x0000 to x00FF (only 8 currently in use)
  - The TRAP instruction
    - which loads the starting address of the TSR into the PC
  - Return link
    - from the end of the TSR back to the original program.

TRAP Example
- Trap Vector Table
  - Or System Control Block
  - In LC-3
    - Specify one of 256 locations (x0000 to x00FF)
    - This location contains the address of the trap service routine.
- TRAP & Interrupts
  - Similar mechanisms
  - A TRAP is an instruction (event internal to a program)
  - An interrupt is external to a program (from an I/O device)
  - Both invoke a supervisor service routine.

LC-3 TRAP Routines
- GETC (TRAP x20)
  - Reads a single character from KBD.
  - Write ASCII code to R0[7:0], clear R0[15:8]
- OUT (TRAP x21)
  - Write R0[7:0] to the monitor.
- PUTS (TRAP x22)
  - Write a string to monitor (address of first character of string in R0)
- IN (TRAP x23)
  - Print a prompt to the monitor and read a single character from KBD.
  - Write ASCII code to R0[7:0], clear R0[15:8], echo character to the monitor.
- HALT (TRAP x25)
  - Print message to monitor & halt execution.
- PUTSP (TRAP x24)
  - Print packed string to monitor (address in R0)
HALT TSR (cont.)

13: Return from the HALT routine
14: (how can this ever happen, if the clock is stopped on line 12??)
15: ...
16: LD R7, SaveR7; Restores registers
17: LD R1, SaveR1; before returning
18: LD R0, SaveR0
19: RET
20: ...;
2A: constants
1C: ASCII NewLine FILL x000A
1D: SaveR0 BLXW 1
1E: SaveR1 BLXW 1
1F: SaveR7 BLXW 1
20: Message STRINGZ “Halting the machine”
21: MCR FILL xFFFE
22: MASK FILL x7FFF
23: ...;
Jump to Subroutine: JSR/JSRR

- A = IR[11] specifies the addressing mode
- JSR: jump to subroutine (PC-Relative), IR[11] = 1
  - R7 ← (PC) i.e. PC is saved to R7
  - PC ← PC + Sext(IR[10:0]) i.e. PC-Relative addressing,
  - using 11 bits => label can be within +1024/-1023 lines of JSR instruction

- JSRR: jump to subroutine (relative base+offset), IR[11] = 0:
  - R7 ← (PC) i.e. PC is saved to R7
  - PC ← (BaseR) + Offset i.e Base+Offset addressing, with offset = 0

Subroutine call example

Calling program

; Calling program
; subroutine multi
; Multiply 2 positive numbers
.ORIG x3000
LD   R1, num1
LD   R2, num2
JSR  multi
ST   R3, prod
HALT

Parameters:
; Input data & result
; num1 .FILL x0006
; num2 .FILL x0003
; prod .BLKW 1

; Loop
ADD   R4, R1, #0
BRz   zero
loop
ADD   R3, R2, R3
ADD   R1, R1, #0
BRp   loop
zero
RET
.END

Notice any undesirable side effects?

Library Routines

- Library
  - A set of routines for a specific domain application.
  - Example: math, graphics, GUI, etc.
  - Defined outside a program.

- Library routine invocation
  - Labels for the routines are defined as external in LC-3:
    - External Label
  - Each library routine contains its own symbol table.
  - A linker resolves the external addresses before creating the executable image.

Practice Problems