Homework 8

2023.07.21 沈韵泓 3200104392

9.16 *a.* How many trap service routines can be implemented in the LC-3? Why?
b. Why must a RET instruction be used to return from a TRAP routine? Why won't a BR (Unconditional Branch) instruction work instead?
c. How many accesses to memory are made during the processing of a TRAP instruction? Assume the TRAP is already in the IR.

a. **256**, for the trap vector in LC-3 is 8-bit, can distinguish at most $2^8 = 256$ service routines.

b. Because `RTI` instruction would save PC and PSR before execute TRAP routine, and restore when finished, while `BR` instruction won't do that.

- Without saving PC, we can't find the return address after finishing TRAP routine.
- Without saving & modifying PSR, the TRAP routine might unable to finish its work due to the lack of supervisor privilege.

c. Only **ONCE**, when visit the Trap Vector Table to load the starting address of corresponding service routine.

9.17 Refer to Figure 9.14, the HALT service routine.
a. What starts the clock after the machine is HALTed? *Hint:* How can the HALT service routine return after bit [15] of the Master Control Register is cleared?
b. Which instruction actually halts the machine?
c. What is the first instruction executed when the machine is started again?
d. Where will the RET of the HALT routine return to?

a. Some **external** mechanism. (The HALT service could **never return** after bit[15] of MCR is cleared, for the clock has stopped, so does the instruction processing)

b. `STI R0, MCR`, which makes `MCR[15] = 0`.

c. `LD R1, SaveR1`

d. The program that executes the HALT instruction (the instruction that is right after HALT).