9.2

a. 256. Because the trap vector is 8 bits wide.

b. After the TRAP routine is executed, program control must be passed back to the code that called the TRAP instruction. This is done by copying the value in R7 into the PC. And the RET provides the function. BRnzp does not restore the PC.

c.1

9.3

a. Some external mechanism start the clock after the machine is HALTed.

b. STI R0,MCR

c. LD R1,SaveR1

d. The RET of the HALT routine will bring program control back to the program that executed the HALT instruction.

9.15

a. TRAP x72

b. Yes, the routine will work, but whatever in R0 before TRAP x72 is executed will be overwritten during the execution of the service routine.

9.17

a. LD R3,NEGENTER

b. STR R0,R1,#0

c. ADD R1,R1,#1

d. STR R2,R1,#0

9.18

a. ADD R1,R1,#1

b. TRAP x25

c. ADD R0,R0,#5

d. BRzp K

10.14

LDI R0,KBDR

LDI R1,NEX;pointer

LD R2,NA

ADD R2,R1,R2

BRz FULL

STR R0,R1,#0

ADD R1,R1,#1

STI R1,NEX

BR DONE

FULL LEA R0,STRI

TRAP x22

DONE RTI

KBDR .FILL xFE02

STA .FILL x4000

NEX .FILL x40FF

NA .FILL xBF01;-x40FF

STRI .STRINGZ "Character cannot be accepted; input buffer full."

10.15

LDI R0,KBDR

LDI R1,NUM;R1<-the number in the queue

LD R2,MAXNUM

ADD R2,R1,R2

BRz FULL ; Buffer is full

ADD R1,R1,#1

STI R1,NUM;update char count

LDI R1,NEX; R1<-pointer

STR R0,R1,#0; store the char

LD R2,NA

ADD R2,R1,R2

BRz CIR; compare the current location with x40FC

;if same, the next empty location turns to x4000

ADD R1,R1,#1; else just increment the pointer

BR STORE

CIR LD R1,STA

STORE STI R1,NEX;update next available empty buffer location pointer

BR DONE

FULL LEA R0,STRI

TRAP x22

DONE RTI

KBDR .FILL xFE02

STA .FILL x4000

NUM .FILL x40FD

MAXNUM .FILL xFF04;-x00FD

NEX .FILL x40FF

NA .FIIL xBF04;-x40FC

STRI .STRINGZ "Character cannot be accepted; input buffer full."