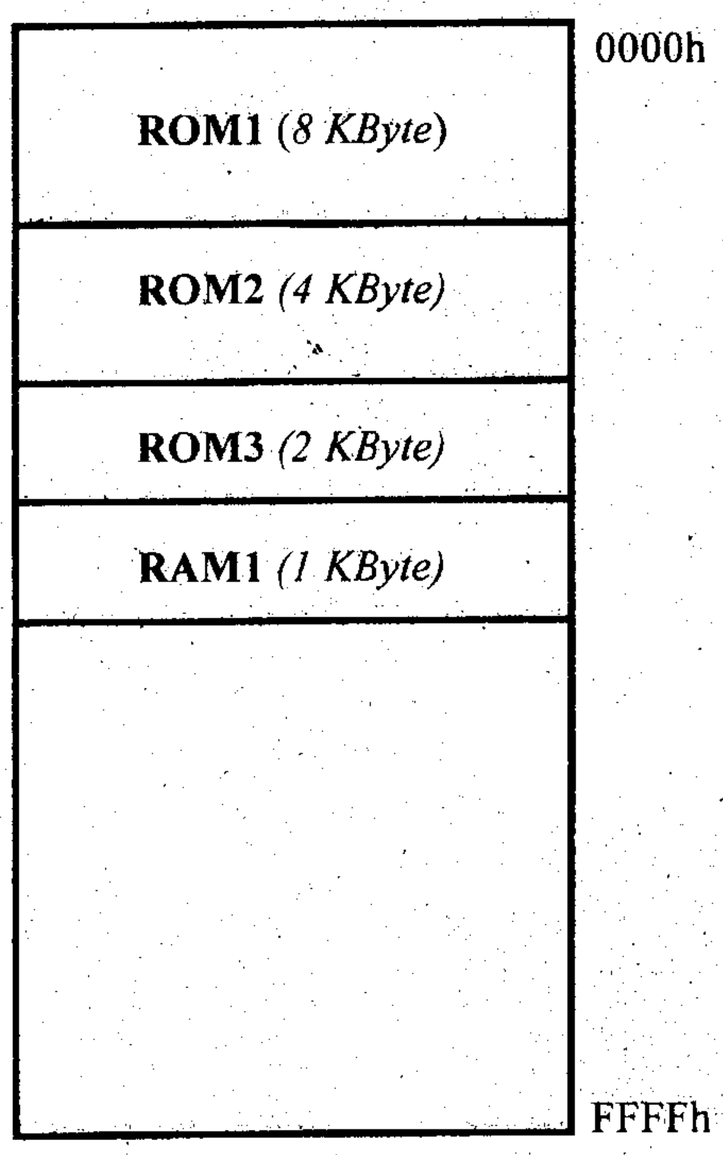
INTRODUCTION TO MICROCOMPUTERS SECOND MIDTERM EXAM

Dr. Salih FADIL **November 28, 2008**

Memory map



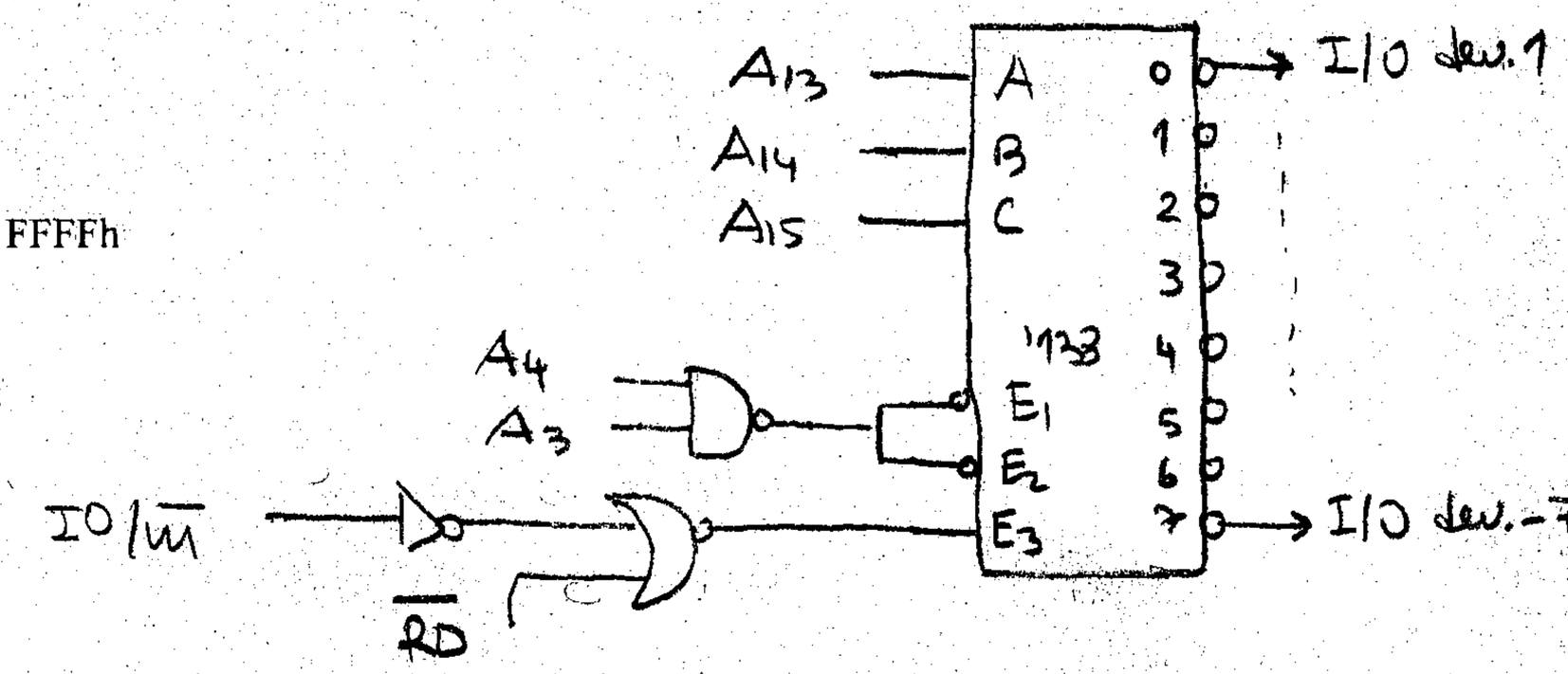
#1) a) Design a PROM memory decoder circuitry that places the memory ICs shown in the memory map to the corresponding address locations. Use a minimum capacity PROM IC. There should not be overlapped addresses in your design. Show connection diagram and programming of the PROM. Assume that the PROM has two chip select inputs. One of them is active-low, the other one is active-high.

b) Consider the following I/O decoder circuitry.

i) What type I/O decoder circuitry is it? Why?

ii) Determine each I/O device's selection address range.

iii) Can data be written to those I/O devices?



#2)

```
                Adjust_H    EQU     number
                Adjust_L    EQU     0FFh
;
                ORG         0000h
                LXI         SP, 0FFFFh
                LXI         D, 0100h
LOOP:           CALL        DELAY
                DCX         D
                MOV         A, E
                ORA         D
                JNZ         LOOP
                HLT
;
DELAY:          PUSH        D
```

```
            MVI     D, Adjust_H
LP_1:       NOP
            CALL    DELAY_1
            DCR     D
            JNZ     LP_1
            POP     D
            RET
;
DELAY_1:    PUSH    D
            MVI     D, Adjust_L
LP_2:       NOP
            DCR     D
            JNZ     LP_2
ZZ:         NOP
            POP     D
            RET
```

a) Consider the above 8085 assembly program that will be run on an 8085 microprocessor based system. The 8085 has 2 MHz crystal. If the elapsed time to execute DELAY subroutine is wanted to be 1 second, determine the value of "*number*" as an hex number.

b) Show the ***active*** part of the stack content when the program reaches the point labeled as ZZ on the assembly program first time. Show the content of the register SP in your diagram also.

#3)

```
            ORG     0000h
            LXI     SP, 0FFFFh
            MVI     A, 00001011b
            SIM
            EI
            MOV     H, A
DUMMY:      NOP
AA:         NOP
            NOP
            NOP
            NOP
BB:         NOP
            NOP
            NOP
            NOP
CC:         NOP
            JMP     DUMMY
            NOP
            ORG     0024h
            JMP     TRAP
;
            ORG     002Ch
            JMP     FIVE_FIVE
            ORG     0034h
```

```
                    JMP     SIX_FIVE
;
                    ORG     003Ch
                    JMP     SEVEN_FIVE
                    NOP
;
TRAP:               MVI     H, 0FFh
LP_TRAP:            DCR     H
                    MOV     A, H
                    CPI     0EEh
                    JNZ     LP_TRAP
DD:                 NOP
                    RIM
                    ANI     00010000b
                    JNZ     FIVE_FIVE
                    MVI     A, 00001111b
                    RIM
EE:                 NOP
                    NOP
                    NOP
                    EI
                    RET

FIVE_FIVE:          MVI     H, 0FFh
LP_FIVE_FIVE:       DCR     H
                    MOV     A, H
                    CPI     0DDh
                    JNZ     LP_FIVE_FIVE
                    IE
                    RET

SIX_FIVE:           MVI     H, 0FFh
LP_SIX_FIVE:        DCR     H
                    MOV     A, H
                    CPI     0CCh
                    JNZ     LP_SIX_FIVE
                    IE
                    RET

SEVEN_FIVE:         MVI     H, 0FFh
LP_SEVEN_FIVE:      DCR     H
                    MOV     A, H
                    CPI     0BBh
                    JNZ     LP_SEVEN_FIVE
;
                    MVI     A, 00001000b
                    RIM
                    IE
                    RET
```

Consider the above assembler program. Assume that the events (or sequence of events) given in the following parts of the question occur just after hard resetting of the microprocessor system under consideration.

a) If an RST 5.5 hardware interrupt request arrives the 8085 at point labeled as AA, what is the content of register H at point CC?

b) If a TRAP hardware interrupt request arrives the 8085 at point labeled as AA and just after that an RST 5.5 hardware interrupt request arrives the 8085 at point labeled as DD, what is the content of register H at point CC?

c) If a TRAP hardware interrupt request arrives the 8085 at point labeled as AA and just after that an RST 5.5 hardware interrupt request arrives the 8085 at point labeled as BB, what is the content of register H at point CC?

d) If an RST 7.5 hardware interrupt request arrives the 8085 at point labeled as AA and just after that an RST 6.5 hardware interrupt request arrives the 8085 at point labeled as BB, what is the content of register H at point CC?

e) If a TRAP hardware interrupt request arrives the 8085 at point labeled as AA and just after that an RST 7.5 hardware interrupt request arrives the 8085 at point labeled as EE, what is the content of register H at point CC?

Please give your short explanations in your answers. ***Answers without explanation are not going to get any point..!***

GOOD LUCK ..... ☺

INTRODUCTION TO MICROCOMPUTERS SECOND MITTERM

EXAM SOLUTION MANUAL

Dr. Salih FADIL

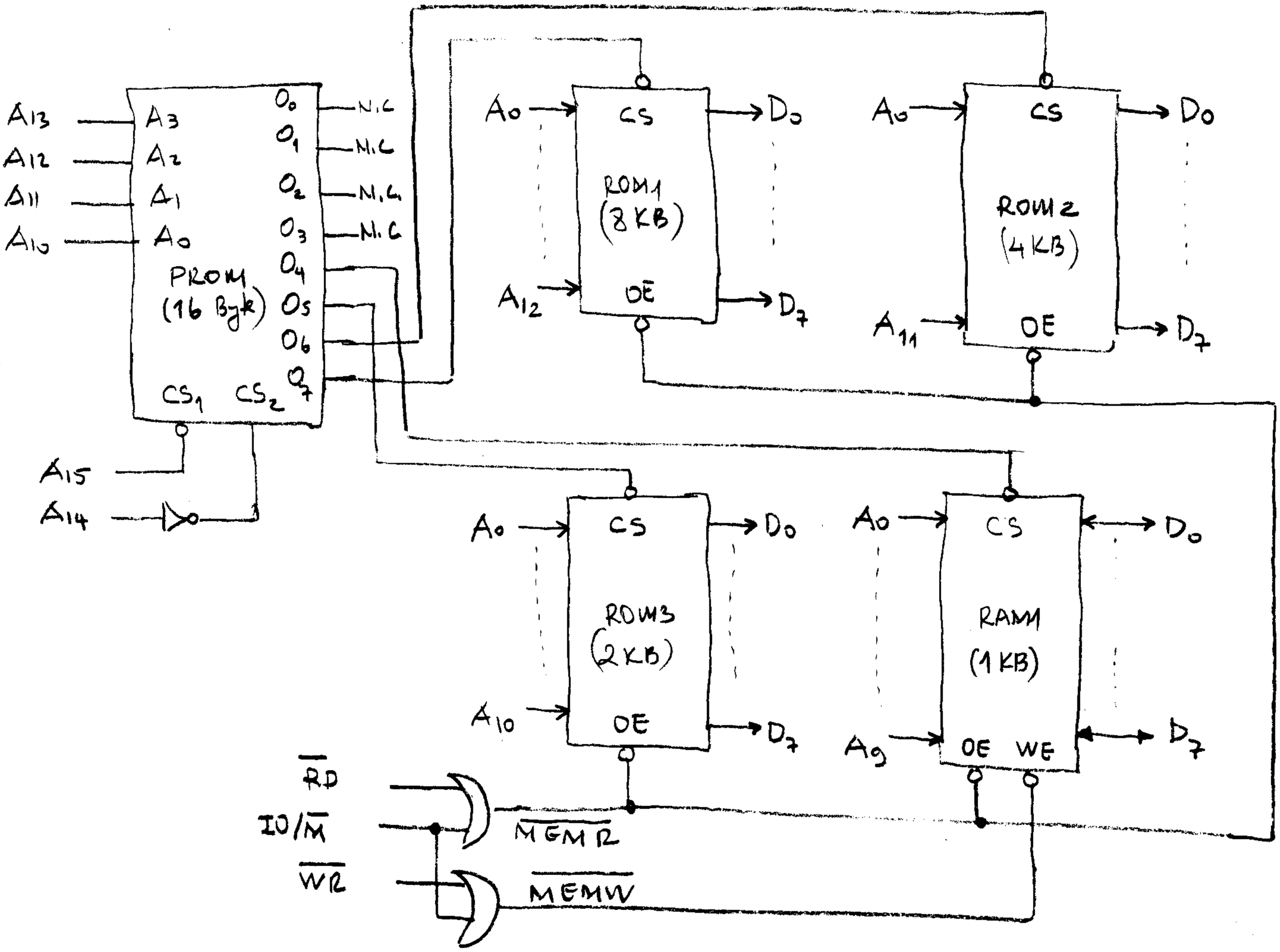
November 28, 2008

#1) a) $8K = 2^3\,2^{10}$, $4K = 2^2\,2^{10}$, $2K = 2^1\,2^{10}$, $1K = 2^{10}$

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	= 0000h	ROM1 (8K)
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	= 1FFFh	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	= 2000h	ROM2 (4K)
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	= 2FFFh	
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	= 3000h	ROM3 (2K)
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	= 37FFh	
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	= 3800h	RAM1 (1K)
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	= 3BFFh	

To CS's of the PROM

A_{13} A_3	A_{12} A_2	A_{11} A_1	A_0 A_0	ROM1 $\overline{CS}$	ROM2 $\overline{CS}$	ROM3 $\overline{CS}$	RAM1 $\overline{CS}$	NC	NC	NC	NC	
				O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0	
0	0	0	0	0	1	1	1	1	1	1	1	for ROM1 eight locations
0	0	0	1	0	1	1	1	1	1	1	1	
⋮				⋮								
0	1	1	1	0	1	1	1	1	1	1	1	
1	0	0	0	1	0	1	1	1	1	1	1	for ROM2 four locations
1	0	0	1	1	0	1	1	1	1	1	1	
1	0	1	0	1	0	1	1	1	1	1	1	
1	0	1	1	1	0	1	1	1	1	1	1	
1	1	0	0	1	1	0	1	1	1	1	1	for ROM3 two locations
1	1	0	1	1	1	0	1	1	1	1	1	
1	1	1	0	1	1	1	0	1	1	1	1	for RAM1 one location
1	1	1	1	1	1	1	1	1	1	1	1	



b) i) It is an isolated I/O decoder since $E_3 = 1$ when $IO/\bar{M} = 1$ and $\overline{RD} = 0$

ii)

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0		
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8		
0	0	0	1	1	0	0	0	= 18h	I/O device-1
0	0	0	1	1	1	1	1	= 1Fh	
0	0	1	—	—	0	0	0	= 38h	I/O device-2
0	0	1	—	—	1	1	1	= 3Fh	
0	1	0	—	—	0	0	0	= 58h	I/O device-3
0	1	0	—	—	1	1	1	= 5Fh	
0	1	1	—	—	0	0	0	= 78h	I/O device-4
0	1	1	—	—	1	1	1	= 7Fh	
1	0	0	—	—	0	0	0	= 98h	I/O device-5
1	0	0	—	—	1	1	1	= 9Fh	
1	0	1	—	—	0	0	0	= B8h	I/O device-6
1	0	1	—	—	1	1	1	= BFh	
1	1	0	—	—	0	0	0	= D8h	I/O device-7
1	1	0	—	—	1	1	1	= DFh	
1	1	1	—	—	0	0	0	= F8h	I/O device-8
1	1	1	—	—	1	1	1	= FFh	

As a result:

I/O Device number	Selected address range
1	18h - 1Fh
2	38h - 3Fh
3	58h - 5Fh
4	78h - 7Fh
5	98h - 9Fh
6	B8h - BFh
7	D8h - DFh
8	F8h - FFh.

(6)

iii) No, it can not be written since $E_3 = 1$ when $IO/\overline{M} = 1$ and $\overline{RD} = 0$. Only data can be read from those I/O devices.

(6)

#2) a)

```
Adjust_H    EQU   number
Adjust-L    EQU   OFFh
            ORG   0000h          ; # of T-states   # of bytes   start. addr
            LXI   SP, OFFFFh     ;                 3            0000h
            LXI   D, 0100h       ;                 3            0003h
LOOP:       CALL  DELAY          ;                 3            0006h
            DCX   D              ;                 1            0009h
            MOV   A, E           ;                 1            000Ah
            ORA   D              ;                 1            000Bh
            JNZ   LOOP           ;                 3            000Ch
            HLT                  ;                 1            000Fh
DELAY:      PUSH  D              ; 12              1            0010h
            MVI   D, Adjust_H    ; 7               2            0011h
LP_1:       NOP                  ; 4               1            0013h
            CALL  DELAY_1        ; 18              3            0014
            DCR   D              ; 4               1            0017h
            JNZ   LP_1           ; 10/7
            POP   D              ; 10
            RET                  ; 10
```

```
DELAY_1:  PUSH   D                ;12
          MVI    D, Adjust_L      ;7
LP_2:     NOP                     ;4
          DCR    D                ;4
          JNZ    LP_2             ;10/7
ZZ:       NOP                     ;4
          POP    D                ;10
          RET                     ;10
```

# of T- States for DELAY_1 Subroutine

Adjust_L = FFh = $(255)_{10}$

$$\underbrace{(12+7)}_{19} + 254\ \underbrace{(4+4+10)}_{18} + \underbrace{(4+4+7)}_{15} + \underbrace{4+10+10}_{24} = 4630$$

# of T-states for DELAY subroutine

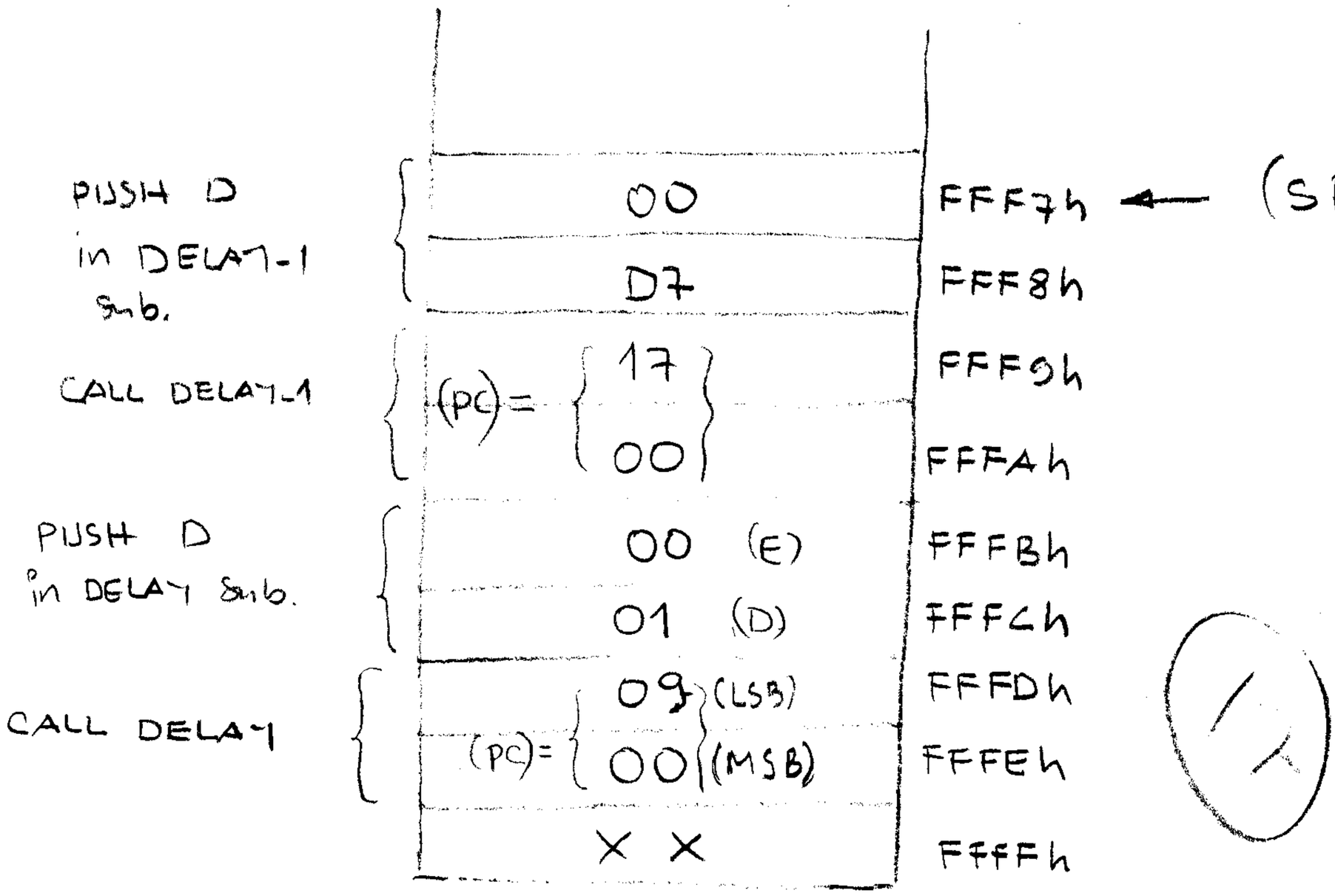
$$\underbrace{12+7}_{19} + (number-1)\ \underbrace{(4+18+4630+4+10)}_{4666} + \underbrace{4663+10+10}_{4683} =$$

$$(number-1)\ 4666 + 4702 = 10^6$$

$$1\ second = \frac{1}{\frac{2}{2}\ 10^6} = 10^6\ T\text{-states.}$$

$$number = \frac{10^6 - 4702}{4666} + 1 = (214.3)_{10} \longrightarrow number = (215)_{10}$$

number = D7h



content of the active stack memory when the program reaches the point labeled as ZZ first time.

(23)

#3) a) At the begining of the assembly program only mask of RST7.5 is removed and interrupts are enabled. Therefore RST5.5 at AA is not recognized by the 8085 and at point CC, (H) = 0Bh.

(7) b) Since TRAP is a NMI, the 8085 finishes the execution of the NOP instruction, pushes the starting address of the next NOP's address into stack. After that it jumps to 0024h, from there it jumps to ISR of TRAP. Inside TRAP's ISR, (H) = EEh, after that pending bit for RST5.5 is checked. Since RST5.5 arrives the 8085 just before this operation, the program jumps to ISR of RST5.5. Inside ISR of RST5.5, (H) = DDh is made. So, (H) = DDh at point labeled as CC.

c) The situation here looks like the one given in part b.

(1) The program finally goes to the ISR for TRAP. (H) = EEh is made. Pending of RST5.5 is cleched. Later on, MVI and RIM instructions are issued. Probably the programmer has written RIM instead of SIM excidently :-). So, the program goes back to the NOP just after the NOP labeled as AA with (H) = EEh. When an RST5.5 interrupt request arrives at point BB, it will not be recognized by the 8085 since its mask is in place. So, at point CC, (H) = EEh

(1) d) When, RST7.5 arrives to the 8085 at AA, the program finally goes to ISR of RST7.5. Inside the ISR, (H) = BBh is made. After that MVI A, 08h, RIM instructions are issued. The programmer (again probably) has written RIM instead of SIM accidently :-) The program returns to the NOP instruction just after the NOP labeled as AA. If an RST6.5 request arrives at point BB, it is not recognized by the 8085, so (H) = BBh at point CC.

(1) e) If a TRAP request arrives to the 8085 at point AA, the program finally reaches to TRAP's ISR. Inside the ISR, register H is loaded with EEh, (H) = EEh, when RST7.5 hardware request reaches to the 8085 at point EE, It is 'latched' as a hardware signal in the RST7.5 latched. So, when the program returns to the loop, it immediately goes to the ISR of RST7.5 (since mask of RST7.5 is been removed). It returns from the subroutine with (H) = BBh. As a result, (H) = BBh at point CC.