Ιωάννης Γεωργακόπουλος - 03111512
Νικόλαος Μαρούσης - 03111025

Εργαστήριο Μικροϋπολογιστών

1η Εργαστηριακή Άσκηση

Άσκηση 1

```
       MVI B,00H    ; Initialize counting flag (0 up, 1 down)

START:              ; Checking the LSB.
       LDA 2000H
       RRC
       JNC START

MAX:                ; Reading the Max time (and resolving a special case).
       RRC          ; Keeping the info on the 4 LSBs
       RRC          ; 3 RRCs because we already did 1 at start
       RRC
       ANI 0FH      ; Clearing the 4 MSBs.
       MOV D,A      ; Special case for when max = timer = 0.
        JNZ CONT    ; Max = 0
       CMP E
       JZ PRINT

CONT:
       CMP E        ; Checking for a special case where the MSB is
       JC FIX       ; less than the current time. (clock >= max)
       JZ FIX
       JMP CHECK

FIX:   MVI B,01H

CHECK:              ; Branch based on counting direction.
       MOV A,B
       CPI 01H      ; If it's 01 we go down.
       MOV A,D      ; Loading timer (DE).
       JZ DOWN

UP:                 ; Count seconds up.
       INR E
```

```
       CMP E
       JNZ PRINT    ; If clock < max  Go to print
       MVI B,01H    ; If clock = max, change flag, then go to print
       JMP PRINT

DOWN:               ; Count seconds down.
       DCR E
       MOV A,E
       CPI 00H      ; If clock = zero, change counting flag
       JNZ PRINT
       MVI B,00H

PRINT:              ; Print the results.
       MOV A,E
       CMA          ; Due to negative logic
       STA 3000H    ; LEDs on
       PUSH B
       LXI B,03E8H  ; 1000ms delay.
       CALL DELB
       POP B
       JMP START

HALT:
       END
```

Άσκηση 2

```
       IN 10H

       MVI A,0DH    ; Activate the RST6.5 interrupt.
       SIM
       EI

       MVI E,00H    ; Initialize plain counter.
       MVI D,00H    ; Initialize interrupt counter.
       LXI B,0064H  ; Set DELB delay to 0,1sec.

LOPO:
       INR E        ; Increase plain counter...
       MOV A,E
       ANI 0FH      ; and trim it to 4 digits.
       RLC
       RLC
       RLC
       RLC          ; Move it over to the 4 MSBs.
       CMA
       STA 3000H    ; Print it.
       DI
       CALL DELB    ; Wait,
       LDA 2000H    ; then check if the switch LSB is set.
       RRC
       JNC LOPO     ; If it's not, keep interrupts disabled and continue.
       EI
       JMP LOPO

INTR_ROUTINE:
       PUSH PSW     ; Push some data down the stack.
       PUSH B
       LXI B,0032H  ; Shorten DELB delay to 50msec, to account for
       CALL DELB    ; RST6.5 signal delay.
       INR D        ; Increment the number of interrupts by one...
       MOV A,D
       ANI 0FH      ; and store as modulo 16.
       MOV D,A
       CALL PRINT   ; Print it to the 7-segment display.
FIN:   RIM          ; Check RST6.5 flag to avoid double-counting the
       ANI 20H      ; interrupt.
       JNZ FIN
       CALL DELB    ; Wait for it,
       EI           ; enable interrupts again,
       POP B        ; restore stacked data...
       POP PSW
       RET          ; and finally return

PRINT:              ; Print the number of interrupts modulo 16
       PUSH H       ; as a single HEX digit.
```

```
        LXI H,0900H  ; We store data in address 0900H.
        MOV M,D      ; Store the iterrupt count as the first digit
        INX H        ; and fill the rest with blanks.
        MVI M,10H
        INX H
        MVI M,10H
        INX H
        MVI M,10H
        INX H
        MVI M,10H
        INX H
        MVI M,10H

        PUSH D
        LXI D,0900H  ; Load the data address for STDM in register D.

        CALL STDM    ; Print the counter.
        CALL DCD

        POP D        ; Restore data and return.
        POP H
        RET

END
```

Άσκηση 3

```
       IN 10H

RD:
       CALL KIND    ; Reading x.
       RAL          ; Multiplying it by 16, by 4 left bitwise shifts.
       RAL
       RAL
       RAL
       MOV B,A      ; Storing x in register B.
       CALL KIND    ; Reading y.
       ADD B        ; (A) = 16*x + y

       LXI H,0905H  ; We store the display data at address 0900H.
       MVI D,00H    ; Initialize hundreds.

HUND:
       CPI 64H      ; If (A)<100 then we move on to decades.
       JC SHUND
       INR D        ; Else we increase hundreds by one,
       SUI 64H      ; subtract 100 from (A)
       JMP HUND     ; and repeat.

SHUND:
       MOV M,D      ; Store hundreds.
       MVI D,00H    ; Initialize decades.
       DCX H

DEC:
       CPI 0AH      ; If (A)<10 then we move on to decades.
       JC UNIT
       INR D        ; Else we increase hundreds by one,
       SUI 0AH      ; subtract 10 from (A)
       JMP DEC      ; and repeat.

UNIT:
       MOV M,D      ; Store decades.
       DCX H
       MOV M,A      ; Store the remaining single digits.
       DCX H

       MVI M,10H    ; Fill the rest with blank characters.
       DCX H
       MVI M,10H
       DCX H
       MVI M,10H
       DCX H
       MVI D,00H

       LXI D,0900H  ; Store the data address in DE before calling STDM.
```

```
        CALL STDM    ; Print the result. :)
        CALL DCD

        JMP RD

END
```

Άσκηση 4

```
       MVI A,0DH    ; Activate the RST6.5 interrupt.
       SIM
       EI

       MVI A,01H    ; The wagon starts from the LSB position.
       MVI D,00H    ; Initialize direction flag. (0 left, 1 right)
       LXI B,01F4H  ; Adjust DELB delay to 0.5 seconds.

       JMP PRINT

START:              ; Check for switch LSB status.
       MOV E,A
       LDA 2000H
       RRC
       MOV A,E
       JNC START    ; If it's not set, wait for it.

CHECK:
       MOV E,A      ; Check the moving direction...
       MOV A,D
       CPI 00H
       MOV A,E
       JNZ GORIGHT  ; and branch accordingly.

GOLEFT:
       CPI 80H      ; When you reach the leftmost position,
       JZ CHANGE    ; it's time to change direction.
       RLC          ; Else just keep going.
       JMP PRINT

GORIGHT:
       CPI 01H
       JZ CHANGE
       RRC
       JMP PRINT

CHANGE:
       CALL CHG     ; Call the routine that changes direction.
       JMP START

PRINT:              ; Print the current state
       DI
       CMA
       STA 3000H
       CMA
       DI
       CALL DELB    ; TODO Interrupts & DELB
       EI
       JMP START
```

```
INTR_ROUTINE:       ; RST6.5 interrupt handler.
       PUSH PSW     ; Push some data down the stack.
       PUSH B
       LXI B,0032H  ; Shorten DELB delay to 50msec, to account for
       CALL DELB    ; RST6.5 signal delay.
       LDA 2000H    ; Check for LSB status.
       RRC
       JNC FIN      ; If it's not set ignore the interrupt.

       CALL CHG     ; Change direction.

FIN:   RIM          ; Check RST6.5 flag to avoid double-counting the
       ANI 20H      ; interrupt.
       JNZ FIN
       CALL DELB    ; Wait for it,
       POP B        ; restore stacked data,
       POP PSW
       EI           ; enable interrupts again...
       RET          ; and finally return

CHG:   PUSH PSW     ; Change directions.
       MOV A,D
       XRI 01H      ; (A) XOR 1 => NOT (A)
       MOV D,A
       POP PSW
       RET

END
```