Ομάδα: Χρίστος Χατζηχριστοφή(03117711)

Δημήτρης Λάμπρος(03117070)

Μάθημα: Συστήματα Μικροϋπολογιστών

Σχολή – Εξάμηνο: ΗΜΜΥ 6ο

Τρίτο σετ ασκήσεων

Άσκηση 1:

```
                        IN 10H
INIT:                   MVI A,0DH           ; enable RST 7,5 and 5,5 and interrupts
                        SIM                 ; interrupt mask
                        LXI B,00FAH         ; 250ms.Initiate E to 60dec.
                        MVI A,3CH           ; Because 60*(250ms+250ms+250ms+250ms) = 60s
                        STA 0A16H           ; store timer counter to 0A16H
                        MVI A,10H           ; fill memory spaces 0A12-0A15 HEX
                        STA 0A12H           ; with 10 (blank print in DCD)
                        STA 0A13H           ; because we want to show our result
                        STA 0A14H           ; to the two rightmost bits which are
                        STA 0A15H           ; 0A10H and 0A11H

INFINITE_LOOP:          EI                  ; enable interrupt
                        JMP INFINITE_LOOP   ; Wait for interrupt

INTR_ROUTINE:           MVI A,3CH           ; Because 60*(250ms+250ms+250ms+250ms) = 60s
                        STA 0A16H           ; store 60 dec in 0A16H
FLICKER:
                        CALL DISPLAY        ; Display TIME
                        MVI A,00H           ; Making sure A is Zero, so flicker is right
                        STA 3000H           ; print complement of A(11111111)
                        CALL DELB           ; delay to see result
                        CMA                 ; complement zero so in outport 00000000 is
                        STA 3000H           ; shown.
                        CALL DELB           ; delay to see result
                        CMA                 ; complement zero so in outport 00000000 is
                        STA 3000H           ; shown.
                        CALL DELB           ; delay to see result
                        STA 3000H           ; shown.
                        CALL DELB           ; delay to see result
                        EI                  ; Restart the timer if interrupted.
                        LDA 0A16H           ; load the counter to reg A
                        DCR A               ; decrease timer counter
                        STA 0A16H           ; store back the updated value of counter
                        CPI 00H             ; Check A with zero.If yes then go back to "INFINITE_LOOP"
                        JZ INFITE_LOOP      ; and wait for an another interrupt
                        JMP FLICKER         ; else keep flickering ;)

DISPLAY:                LDA 0A16H           ; Temporary move of time counter to A
                        MVI H,00H           ; init H to zero -- which is our counter for tens

TENS:                   CPI 0AH             ; compare input with 10
                        JC UNITS            ; if A is less than 10 then we have no tens
                        SUI 0AH             ; Subtract 10 from accumulator(input)
                        INR H               ; tens++
                        JMP TENS            ; loop until no more TENS exist

UNITS:                  STA 0A10H           ; if we reached this point, then inside A are only units
                        MOV A,H             ; moving tens to A so as we store them
                        STA 0A11H           ; Store decades to 2nd rightmost display pos
                        LXI D,0A10H         ; Give Code startin address.
                        CALL STDM           ; Do the printing in the 7segment display
                        CALL DCD            ;
                        RET
END
```

Άσκηση 2:

```
IN 10H
INIT:                   MVI A,0DH           ; enable RST 7,5 and 5,5 and interrupts
                        SIM                 ; set interrupt mask
                        MVI A,10H           ; fill memory spaces 0A10-0A13 HEX
                        STA 0A10H           ; with 10 (blank print in DCD)
                        STA 0A11H           ; so we print the result only on the
                        STA 0A12H           ; two leftmost digits of the
                        STA 0A13H           ; 7segment display

INFINITE_LOOP:          EI                  ; enable interrupt
                        JMP INFINITE_LOOP   ; Wait for interrupt


INTR_ROUTINE:           CALL KIND           ; read input from keyboard

DISPLAY:                STA 0A15H           ; Store LSBs to 2nd leftmost disp pos
                        RLC                 ; make 4 left shifts so as the number in
                        RLC                 ; A goes to the 4 most significant bits
                        RLC                 ; The reason is because we want to
                        RLC                 ; construct the whole number given
                        STA 0A16H           ; Store the shifted number to 0A16
                        CALL KIND           ; call kind to read input from keyboard
                        STA 0A14H           ; Store LSBs to 2nd leftmost disp pos
                        LXI D,0A10H         ; Give Code startin address.
                        CALL STDM           ; Print the number given from keyboard
                        CALL DCD            ; to 7 segment display
                        LDA 0A16H           ; finalise number by loading back the
                        MOV B,A             ; number stored in 0A16, moving it to B
                        LDA 0A14H           ; loading 0A14(second leftmost digit)
                        ADD B               ; and adding it to B
                        MVI C,10H           ; Initializing the 3 areas
                        MVI D,20H           ; by adding numbers to C,D,E
                        MVI E,30H
                        CMP C               ; Here we start comparisons so as
                        JC FIRST_CASE       ; we see in which area the number
                        JZ FIRST_CASE       ; given from keyboard is.
                        CMP D               ; When found, we print the output
                        JC SECOND_CASE      ; and return to INFINITE_LOOP
                        JZ SECOND_CASE      ; where we wait for an other interrupt.
                        CMP E
                        JC THIRD_CASE
                        JZ THIRD_CASE
FOURTH_CASE:            MVI A,08H
                        CMA
                        STA 3000H
                        JMP INFINITE_LOOP
THIRD_CASE:             MVI A,04H
                        CMA
                        STA 3000H
                        JMP INFINITE_LOOP
SECOND_CASE:            MVI A,02H
                        CMA
                        STA 3000H
                        JMP INFINITE_LOOP
FIRST_CASE:             MVI A,01H
                        CMA
                        STA 3000H
                        JMP INFINITE_LOOP

END
```

Άσκηση 3:

a)

```
INR16 MACRO ADDR
PUSH PSW              ; Pushing A,flags so as their contents are saved
PUSH H                ; Pushing H to the stack so as the contents of HL are saved
LHLD ADDR             ; Load ADDR to Pair Register HL
INX H                 ; Increase the content of Pair Register HL
SHLD ADDR             ; Store Pair Register HL to ADDR
POP H                 ; Get back the content of Pair Register HL
POP PSW               ; Get back the content of A and flags
ENDM
```

b)

```
FILL MACRO ADDR, K
                PUSH PSW            ; Pushing A,flags so as their contents are saved
                PUSH D              ; Pushing H to the stack so as the contents of HL are saved
                MOV A,K             ; Move K to A
                LXI D,ADDR          ; Load ADDR to Pair Register DE
                CPI 00H             ; Compare A(which is K) if is equal to zero
                JZ ISMAX            ; if yes then jmp to ISMAX label
                CPI FFH             ; else check if is less than equal to 255
                JC CONTINUE         ; if is equal to 255 CONTINUE
                JZ CONTINUE         ; if is less than 255 CONTINUE
                JMP EXIT            ; else EXIT, invalid number
ISMAX:          STAX D              ; Store A to ADDR(which is stored to DE rp)
                INX D               ; Increase ADDR by one
                DCR A               ; Decrease A(which is K) by one
CONTINUE:       CPI 00H             ; Check if is equal to zero
                JZ EXIT             ; if yes then EXIT
                STAX D              ; else store A to ADDR(which is stored to DE rp)
                INX D               ; Increase ADDR by one
                DCR A               ; Decrease A(which is K) by one
                JMP CONTINUE        ; and continue the loop
EXIT:           PUSH D              ; Get back the content of Pair Register HL
                POP PSW             ; Get back the content of A and flags

ENDM
```

c)

```
; First we move R to A.With the first RAL, content of A(which is R) is being shifted left.
; D7 is going to be the CY flag and previous CY flag is going to be D0. After this action we move A back to R.
; Now we move Q to R. We operate one RAL. Content of A(which now is Q) is being shifted left. D7 is
; going to be the new CY flag and previous CY flag(which is D7 of R) is going to be D0.

RHLR MACRO Q,R
PUSH PSW            ; Pushing A,flags so as their contents are saved
MOV A,R
RAL
MOV R,A
MOV A,Q
RAL
MOV Q,A
POP PSW             ; Get back the content of A and flags
ENDM
```

Άνσκηση 4:

- *Ολοκληρώνεται η τρέχουσα εντολή:*

PC	0900H
SP	1FF0H

- *Αποθήκευση των PCH,PCL στην στοίβα.*

SP	PCH
1FEFH	09

SP	PCL
1FEEH	00

- *Αποθήκευση των A,B,D,H και flags σε δύο διαδοχικές θέσεις.*

SP	Contains
1FEDH	A
1FECH	Flags
1FEBH	B
1FEAH	C
1FE9H	D
1FE8H	E
1FE7H	H
1FE6H	L

- Αναγνώριση συσκευής και προτεραιότητας

- *Μεταβαίνουμε στην ρουτίνα εξυπηρέτησης του Interrupt:*

  $PC \leftarrow 0034H$, αφού έχουμε Hardware διακοπή.

- *Τελικά η στοίβα έχει την εξής μορφή:*

1FE6H	L
1FE7H	H
1FE8H	E
1FE9H	D
1FEAH	C
1FEBH	B
1FECH	FLAGS
1FEDH	A
1FEEH	00
1FEFH	09

- Ανακτάται η κατάσταση του μΕ, δηλαδή γίνονται pop τα στοιχεία απο την στοίβα.

- Επιστρέφει ο έλεγχος στην επόμενη εντολή απο αυτή που έγινε η διακοπή, δηλαδή εκεί που δείχνει ο PC, αφού γίνει pop απο την στοίβα.

Άσκηση 5:

```
                MVI A,0EH               ; enable ONLY RST 5,5 and interrupts
                SIM                     ; interrupt mask

                MVI H,00H               ; Init HL pair
                MVI L,00H               ; to zero which will hold sum
                MVI B,10H               ; B=16 dec
                MVI D,02H               ; Every time D is zero, it means we read one data packet

WAIT_LABEL:     EI                      ; enable interrupt
                MOV A,B                 ; Moving B to A so as we can compare if B reached zero
                CPI 00H                 ; and so we read all data packages.
                JZ TELOS                ; if this is true then we jump to TELOS
                JMP WAIT_LABEL          ; wait for an other interrupt


INTR_ROUTINE:   DCR D                   ; decrease D every time so as we know if we read a whole number
                MOV A,D                 ; Moving D to A to make the comparison
                CPI 00H                 ; If zero then we read a full number
                JZ ADDBITS              ; so we go to ADDBITS to complete data packet
                IN 20H                  ; else we read the half packet(which are MSBs)
                ANI 0FH                 ; keep incoming MSbits(X0 - X3)
                RLC                     ; 4 rotations
                RLC
                RLC
                RLC
                MOV C,A                 ; C has MSBs of Data
                JMP WAIT_LABEL          ; wait for an other interrupt

ADDBITS:        DCR B                   ; B--
                IN 20H                  ; read the second half of the packet(LSBs)
                ANI 0FH                 ; keep incoming bits(X0 - X3)
                ADD C                   ; now we have completed the packet
                MOV E,C                 ; Moving C to E and 0 to D so as
                MVI D,00H               ; in pair register DE the packet is formed (00000000PACKET)
                DAD D                   ; add DE register to HL (HL holds the sum)
                MVI D,02H               ; reset D, to read an other packet
                JMP WAIT_LABEL          ; wait for an other interrupt
TELOS:          DI                      ; disable interrupt
                DAD H                   ; By doing 4 DAD H, we move left the number by one every time
                DAD H                   ; we did a total of 4 DAD so as to divide the total_sum by
                DAD H                   ; 2^(8 - how_many_times_we_used_DAD). At the end, H will have
                DAD H                   ; the integer part of average and L will have the nominator of
                                        ; the fraction average (*16)

END
```