

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Συστήματα Μικρούπολογιστών

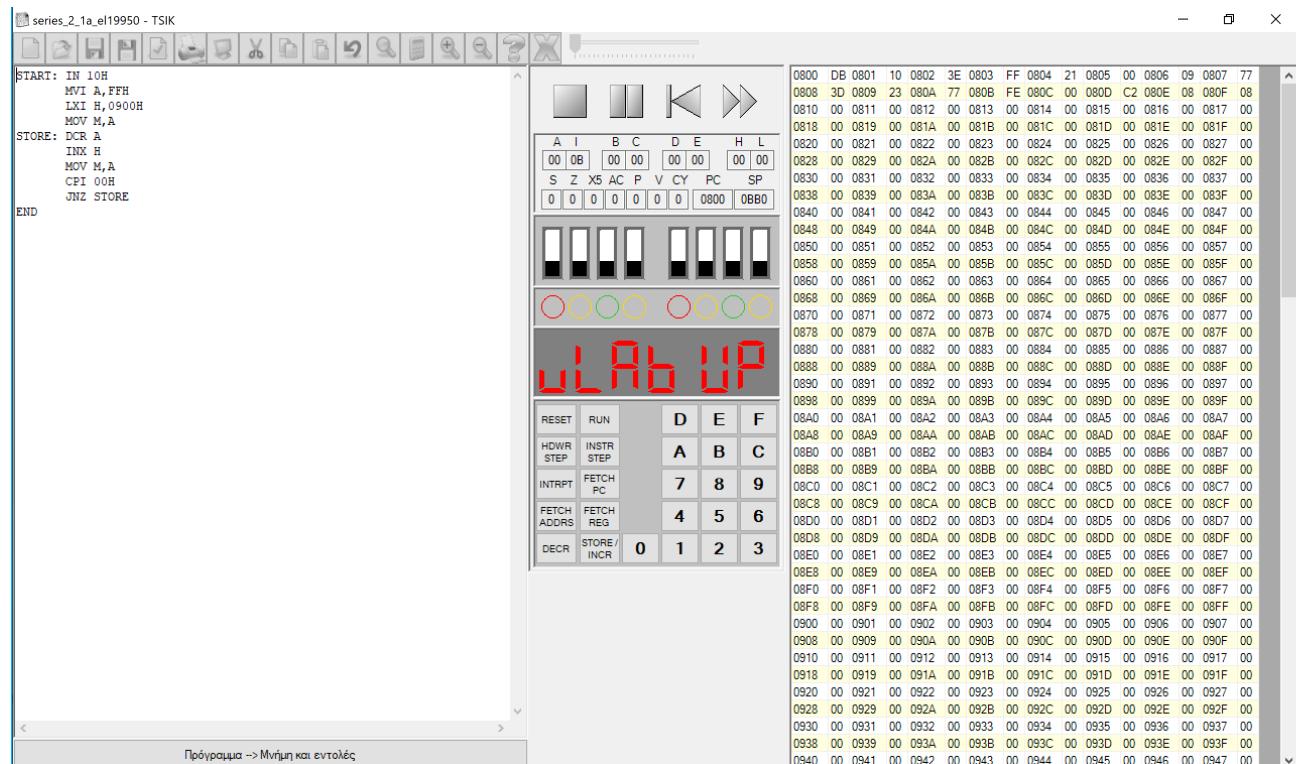
2η Ομάδα Ασκήσεων

Κωνσταντίνος Μπαντής 031 19410
Αργυρώ Τσίπη 031 19950

Άσκηση 1

a)

```
START: IN 10H      // απενεργοποιεί την προστασία μνήμης
       MVI A,FFH    // A = 255
       LXI H,0900H   // 09H = (H) , 00H = (L)
       MOV M,A      // το περιεχόμενο του A μπαίνει στη θέση μνήμης των καταχωρητών HL
STORE: DCR A      // μειώνουμε το A κατά ένα. A = A-1
       INX H        // αυξάνουμε τη θέση των καταχωρητών HL κατά ένα
       MOV M,A      // το περιεχόμενο του A μπαίνει στη θέση μνήμης των καταχωρητών HL
       CPI 00H      // εάν A = 00H τότε Z =1, εάν A<00H τότε CY=1
       JNZ STORE    // εάν Z != 0 τότε πήγαινε στο STORE
END               // τέλος
```



β)

TOTALZ: MVI B,00H

COUNT: MOV A,C

RBC

JC SKIP

INXD

SKIP: MOV C,A

INR B

INR B
MOV A B

CPI 08H

OFFSPRING
INZ GO

ONE COUNT
MOV AC

MOV A,
CPL 00H

CPT 001
17 FINIS

JZ FINIS
DCB C

DCR C
IMP TO

JMP TOTAL
MOV AD

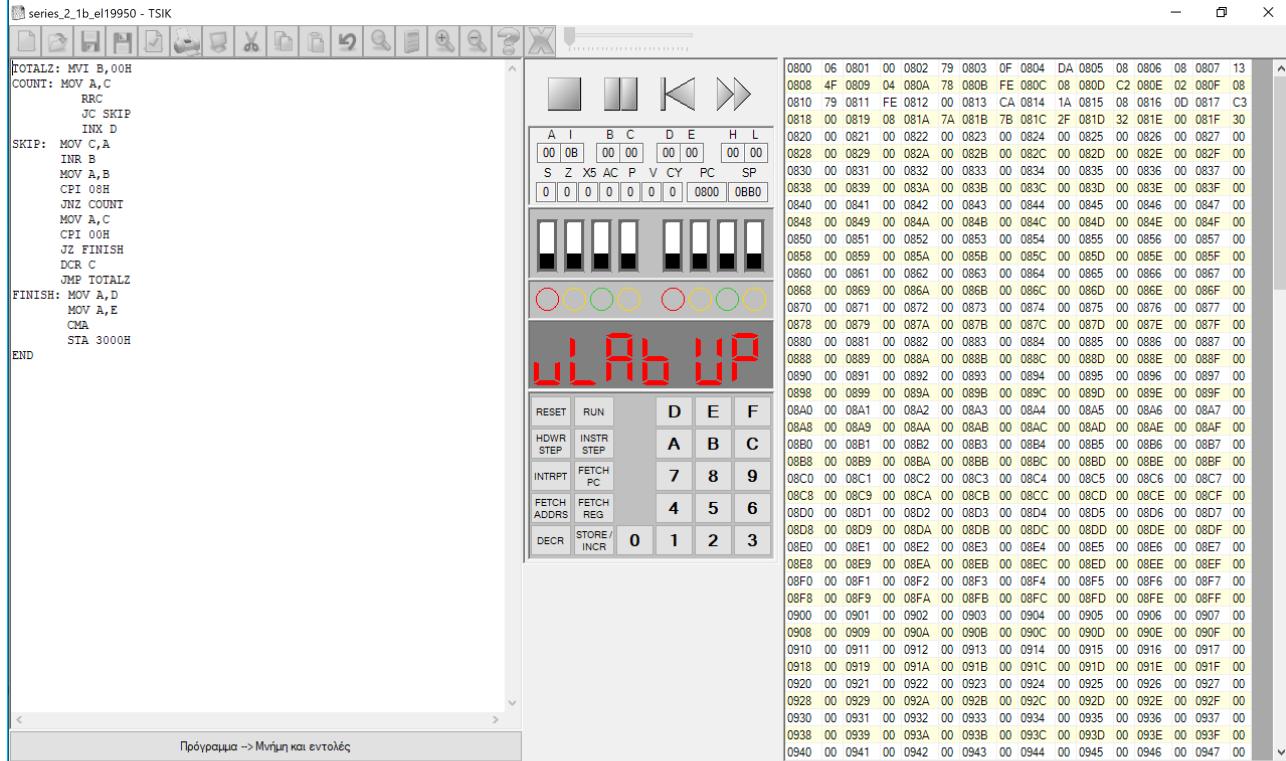
FINISH: MOV A,D
MOV A,E

MOV A,E

CMA
874

STIA 3000H

END



c)

START: MVI C,00H
MVI A,FFH

COMPARE: CPI 20H
JC SKIP
CPI 71H
JNC SKIP
INR C

```
SKIP:  
        CPI 00H  
        JZ FINISH2  
        DCR A  
        JMP COMPARE
```

FINISH2: MOV A,C
CMA
STA 3000H

RST 1

END

The screenshot shows the WLab UP software interface. The top menu bar includes File, Edit, View, Tools, Help, and a Language selection. The left sidebar contains sections for START, COMPARE, SKIP, FINISH2, and END, each listing various assembly instructions like MVI, CPI, JC, JNC, INR, DCR, MOV, STA, and RST. The main workspace is divided into several panels: a top panel with status icons (CPU, RAM, ROM, ROM2, ROM3, ROM4), a central panel with memory dump tables for A, B, C, D, E, F, H, L, S, Z, X5, AC, P, V, CY, PC, and SP, a middle panel with memory dump tables for A, B, C, D, E, F, H, L, S, Z, X5, AC, P, V, CY, PC, and SP, a bottom panel with memory dump tables for A, B, C, D, E, F, H, L, S, Z, X5, AC, P, V, CY, PC, and SP, and a bottom-most panel with control buttons for RESET, RUN, HDWR, INSTR, STEP, STEP, INTRPT, PC, FETCH, ADDRS, FETCH, REG, DECR, STORE, and INCR. The word "WLab UP" is prominently displayed in red in the center of the interface.

Άσκηση 2

START: LXI B,00C8H ;used for 0.2 sec delay
MVI D,00H ;D contains remaining time for light switch
MVI E,FFH ;E is used to print output (lights)

WAIT_FOR_ON: CALL DELB
MOV A,D
ANI 03H ;keep only the last two digits
CPI 03H ;switches lights every 4 repeats
JNZ SKIP_LIGHT_SWITCH_1
MOV A,E ;previous light condition
CMA ;switch
STA 3000H
MOV E,A
MOV A,D
CPI 00H ;if A = 00H then z=1, if A<00H CY=1
JZ TURN_OFF_1 ;turn lights off if A=D=0
DCR D ;decrease D
JMP SKIP_TURN_OFF_1

SKIP_LIGHT_SWITCH_1:
MOV A,FFH
STA 3000H
MVI E,FFH

TURN_OFF_1:
MVI A,FFH
STA 3000H
MVI E,FFH

SKIP_TURN_OFF_1:
LDA 2000H
ANI 01H
CPI 01H ;if A = 01H then z=1, if A<01H CY=1
JNZ WAIT_FOR_ON ;if Z = 0 go to WAIT FOR ON

WAIT_FOR_OFF: CALL DELB
MOV A,D
ANI 03H ;keep only the last two digits
CPI 03H ;switches lights every 4 repeats (0.8s)
JNZ SKIP_LIGHT_SWITCH_2
MOV A,E ;previous light condition
CMA ;switch
STA 3000H
MOV E,A
MOV A,D
CPI 00H ;if A = 00H then z=1, if A<00H CY=1
JZ TURN_OFF_2 ;turn lights off if A=D=0
DCR D ;decrease D
JMP SKIP_TURN_OFF_2

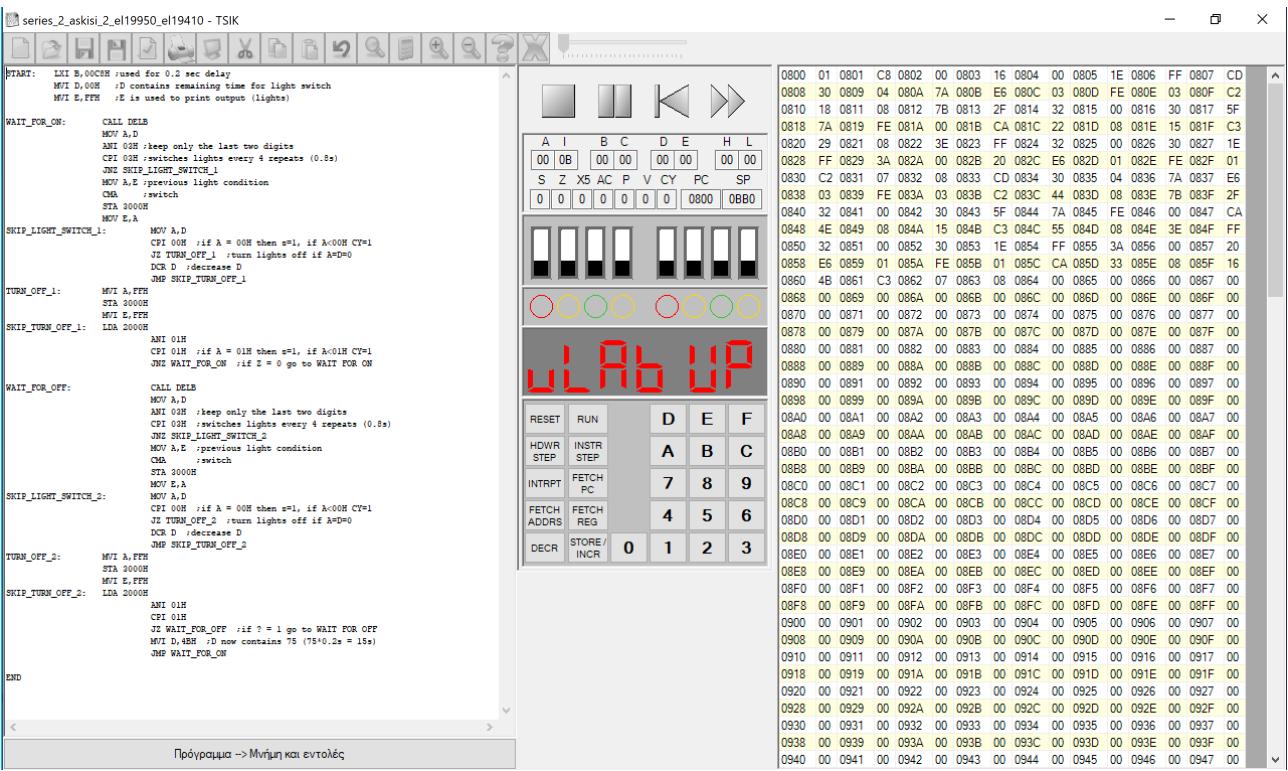
SKIP_LIGHT_SWITCH_2:
MVI A,FFH
STA 3000H
MVI E,FFH

TURN_OFF_2:
MVI A,FFH
STA 3000H
MVI E,FFH

SKIP_TURN_OFF_2:
LDA 2000H
ANI 01H
CPI 01H
JZ WAIT_FOR_OFF ;if Z = 1 go to WAIT FOR OFF
MVI D,4BH ;D now contains 75 (75*0.2s = 15s)

JMP WAIT_FOR_ON

END



Άσκηση 3

a)

START:LDA 2000H

 MVI B,00H
 MVI C,00H

CPI 00H
JZ START

L:
 INR B
 MOV D,A
 MOV A,B

CPI 09H
JNC LAST
MOV A,D
RAL
JC GO
JMP L

GO: CALL DO_C
CALL PUT

LAST: CALL FIX
JMP START

ONE: MOV D,A
MVI A,01H
CPI 05H

MOV A,D
RET

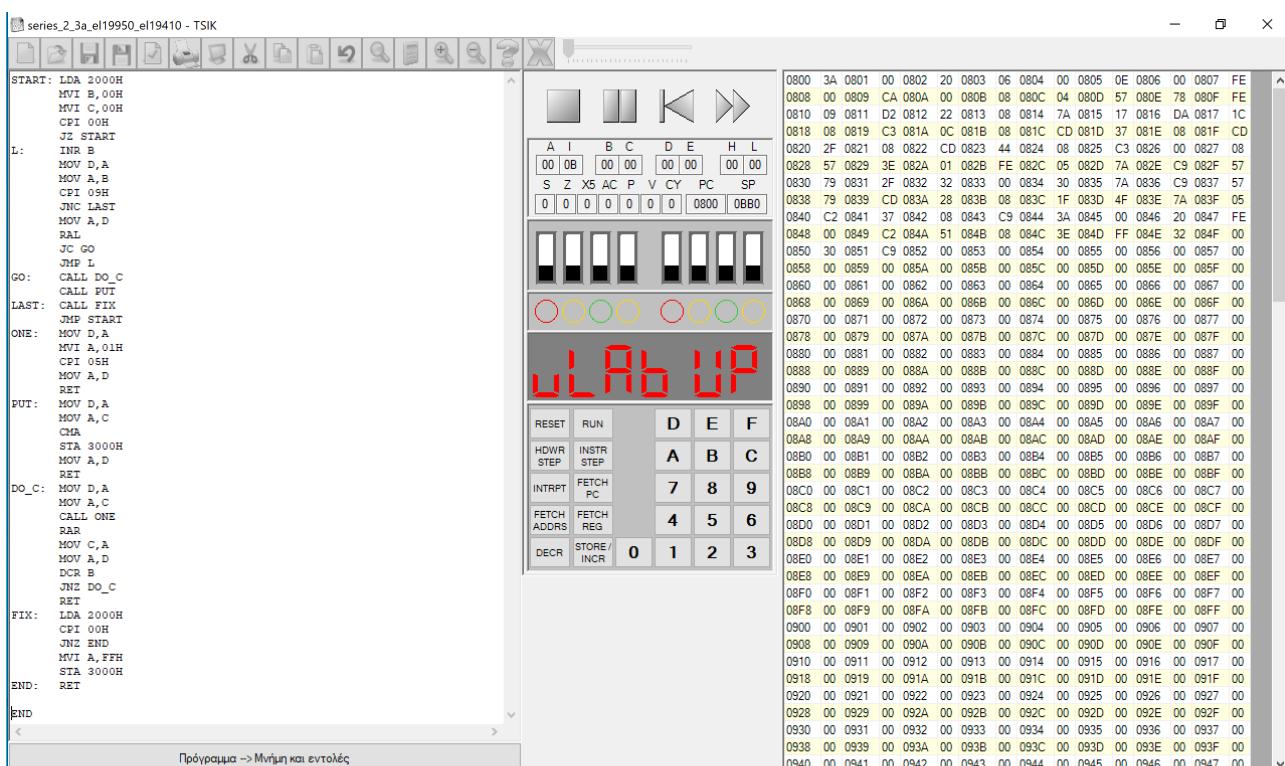
PUT: MOV D,A
MOV A,C
CMA
STA 3000H
MOV A,D
RET

DO_C: MOV D,A
MOV A,C
CALL ONE
RAR
MOV C,A
MOV A,D
DCR B
JNZ DO_C
RET

FIX: LDA 2000H
CPI 00H
JNZ END
MVI A,FFH
STA 3000H

END: RET

END



b)

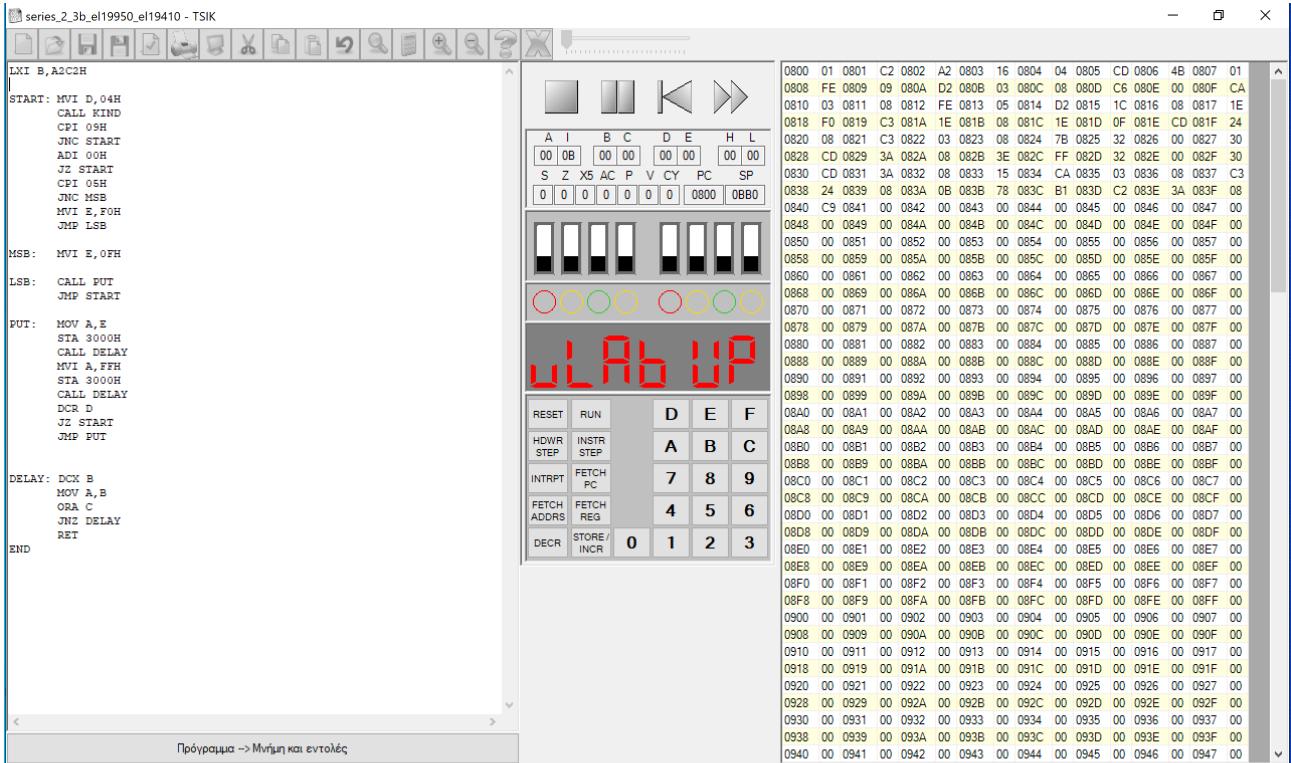
LXI B,A2C2H
START: MVI D,04H
CALL KIND
CPI 09H
JNC START
ADI 00H
JZ START

CPI 05H
JNC MSB
MVI E,F0H
JMP LSB
MSB: MVI E,0FH
LSB: CALL PUT
JMP START

PUT: MOV A,E
STA 3000H
CALL DELAY
MVI A,FFH
STA 3000H
CALL DELAY
DCR D
JZ START
JMP PUT

DELAY: DCX B
MOV A,B
ORA C
JNZ DELAY
RET

END



c)

START: IN 10H
LXI H,0A00H
MVI B,04H
L1: MVI M,10H
INX H
DCR B
JNZ L1

LINE0: MVI A,FEH
STA 2800H
LDA 1800H
ANI 07H

MVI C,86H
CPI 06H
JZ SHOW

MVI C,85H
CPI 05H

JZ SHOW

LINE1: MVI A,FDH
STA 2800H

LDA 1800H
ANI 07H

MVI C,84H
CPI 06H
JZ SHOW
MVI C,80H
CPI 05H
JZ SHOW
MVI C,82H
CPI 03H
JZ SHOW

LINE2: MVI A,FBH
STA 2800H
LDA 1800H
ANI 07H

MVI C,00H
CPI 06H
JZ SHOW
MVI C,83H
CPI 05H
JZ SHOW
MVI C,81H
CPI 03H
JZ SHOW

LINE3: MVI A,F7H
STA 2800H
LDA 1800H
ANI 07H

MVI C,01H
CPI 06H
JZ SHOW
MVI C,02H
CPI 05H
JZ SHOW
MVI C,03H
CPI 03H
JZ SHOW

LINE4: MVI A,EFH
STA 2800H
LDA 1800H
ANI 07H

MVI C,04H
CPI 06H
JZ SHOW
MVI C,05H
CPI 05H
JZ SHOW

MVI C,06H
CPI 03H
JZ SHOW

LINE5: MVI A,DFH
STA 2800H
LDA 1800H
ANI 07H

MVI C,07H
CPI 06H
JZ SHOW
MVI C,08H
CPI 05H
JZ SHOW
MVI C,09H
CPI 03H
JZ SHOW

LINE6: MVI A,BFH
STA 2800H
LDA 1800H
ANI 07H

MVI C,0AH
CPI 06H
JZ SHOW
MVI C,0BH
CPI 05H
JZ SHOW
MVI C,0CH
CPI 03H
JZ SHOW

LINE7: MVI A,7FH
STA 2800H
LDA 1800H
ANI 07H

MVI C,0DH
CPI 06H
JZ SHOW
MVI C,0EH
CPI 05H
JZ SHOW
MVI C,0FH
CPI 03H
JZ SHOW

JMP START

SHOW:LXI H,0A04H
MOV A,C
ANI 0FH

MOV M,A

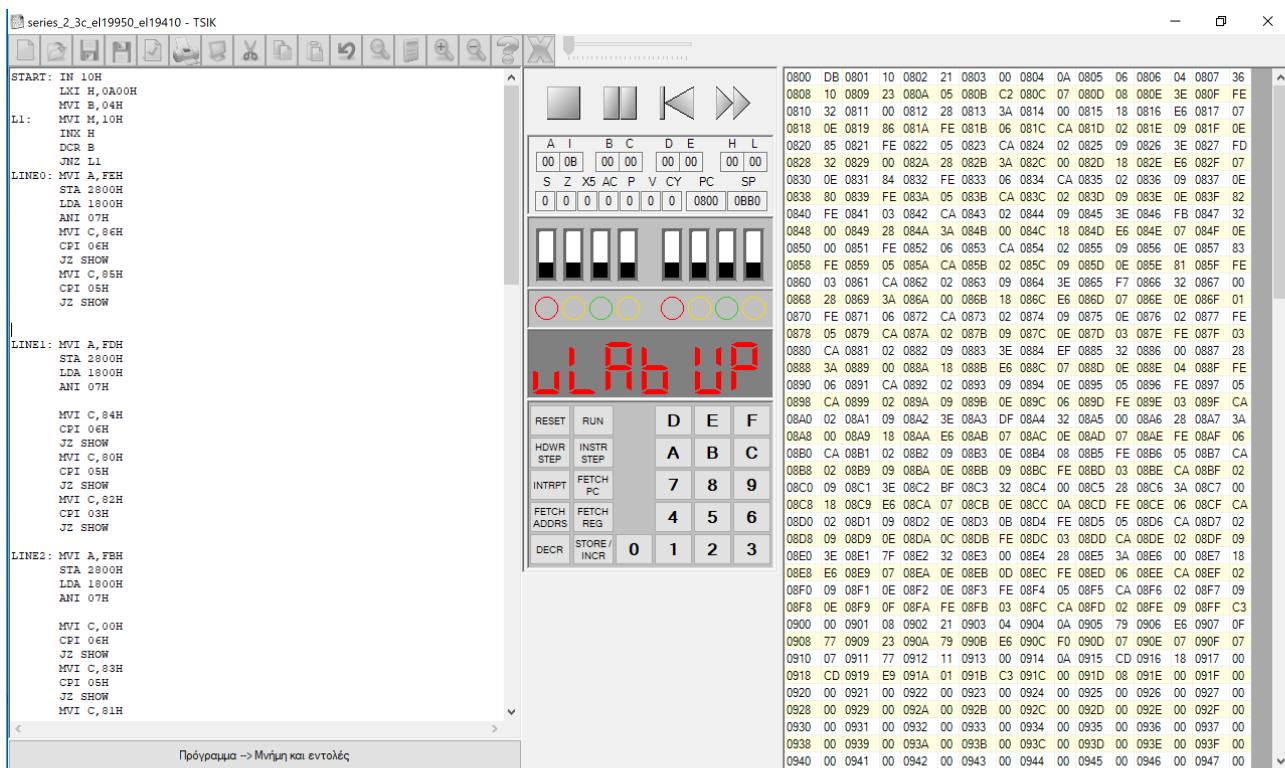
INX H
MOV A,C
ANI F0H
RLC
RLC
RLC
RLC
MOV M,A

LXI D,0A00H

CALL STDM
CALL DCD

JMP START

END



Άσκηση 4

START:
LDA 2000H
MOV B,A ;B contains input
RRC
MOV C,A ;C contains input rotated right once
ANA B ;A contains (B and C)
RRC
MOV D,A
ORI 02H ;A1 or B1
MOV E,A ;E contains A1 OR B1 in correct position
MOV A,D
RRC
RRC
ANI 08H ;A3 and B3
MOV D,A ;D contains A3 AND B3 in correct position
MOV A,C
ORA B ;A contains (B or C)
MOV C,A
ORI 01H ;A0 or B0
MOV B,A ;B contains A0 OR B0 in correct position
MOV A,C
RRC
RRC
ANI 04H ;A contains A2 and B2 in correct position
MOV C,A ;C contains A2 AND B2 in correct position
ANA B
ANA E
ADD E
ADD B
MOV B,A
MOV A,C
RLC
XRA D
ADD B
CMA
STA 3000H
JMP START
END

series_2_askisi_4_el19950_el19410 - TSIK

```

START: LDA 2000H
    MOV B,A ;B contains input
    RRC
    MOV C,A ;C contains input rotated right once
    ANA B ;A contains (B and C)
    RRC
    MOV D,A
    ORI 02H ;A1 or B1
    MOV E,A ;E contains A1 OR B1 in correct position
    MOV B,D
    RRC
    RRC
    ANI 08H ;A3 and B3
    MOV D,A ;D contains A3 AND B3 in correct position

    MOV A,C
    ORI B,A ;B contains (B or C)
    MOV C,A
    ORI 01H ;A0 or B0
    MOV B,A ;B contains A0 OR B0 in correct position
    MOV A,C
    RRC
    RRC
    ANI 04H ;A contains A2 and B2 in correct position
    MOV C,A ;C contains A2 AND B2 in correct position

    ANA B
    ANA E

    ADD E
    ADD B
    MOV B,A
    MOV A,C
    RLC
    XRA D
    ADD B

    GMA
    STA 3000H
    JMP START
END

```

Πρόγραμμα → Μνήμη και εντολές

A	I	B	C	D	E	H	L
00	0B	00	00	00	00	00	00
S	Z	X5	AC	P	V	CY	PC
0	0	0	0	0	0	0	0B00
SP							

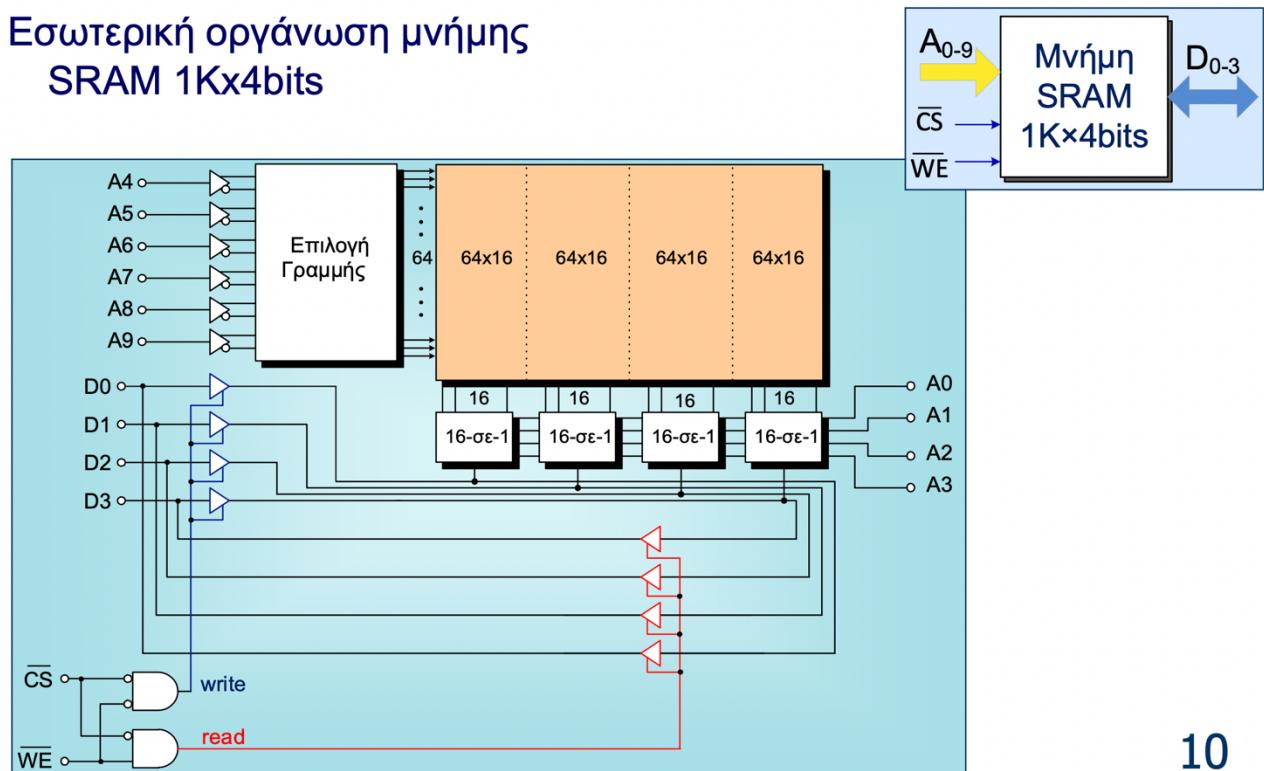
RESET	RUN	D	E	F
HDWR	INSTR	A	B	C
STEP	STEP	7	8	9
INTRPT	FETCH	4	5	6
FETCH	ADDRS	0	1	2
ADDRS	REG	3		
DECR	STORE / INCR			

μLAB UP

A ₀₋₉	Mnήμη SRAM 1Kx4bits	D ₀₋₃
CS		
WE		

Άσκηση 5

Εσωτερική οργάνωση μνήμης SRAM 1Kx4bits



10

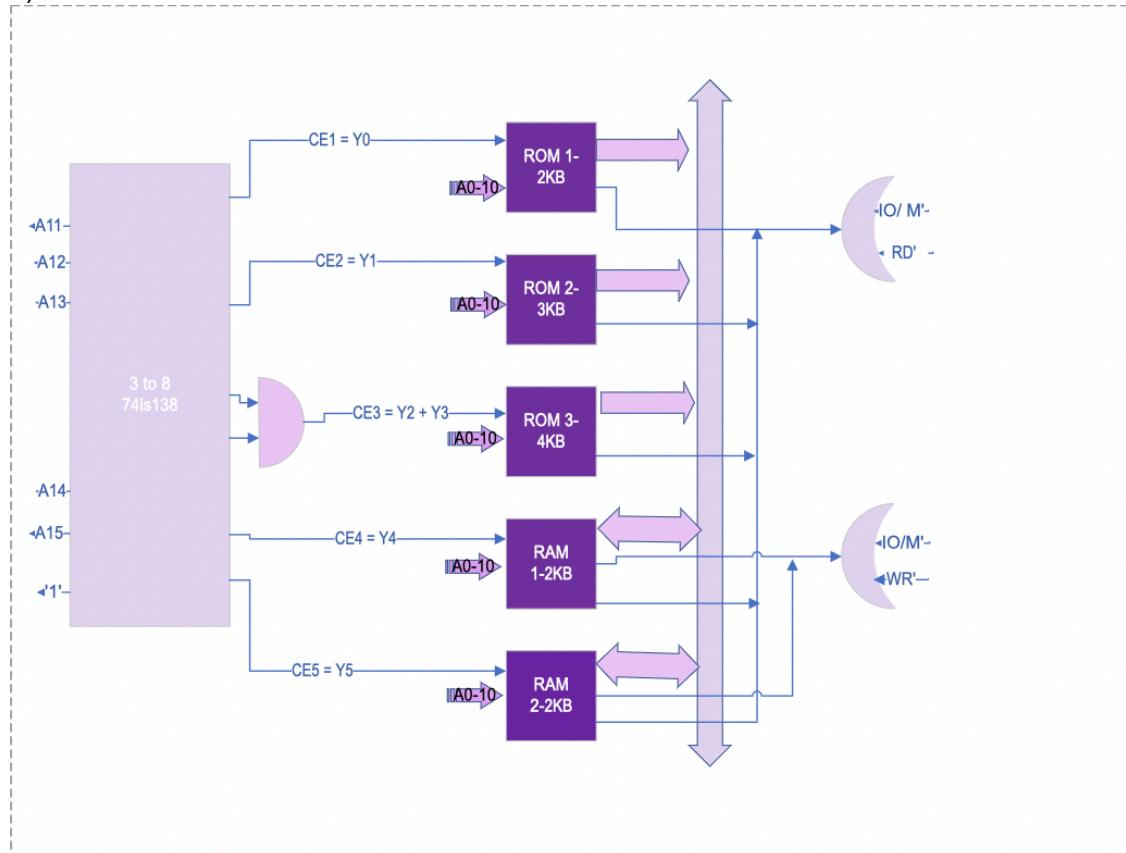
Στις διαφάνειες «Συστήματα Μνημών» σελίδα 10 βλέπουμε την οργάνωση SRAM 256x4 bit. Οι 16 γραμμές των πολυπλεκτών MUX 16x1 επιλέγονται με βάση τις γραμμές διεύθυνσης A4-A7 που

φαίνονται αριστερά της εικόνας. Έπειτα συνδέονται με τις γραμμές δεδομένων D0-D3 και επιλέγουν μία από τις 16 τετράδες μέσω των γραμμών διευθύνσεων A0-A3. Έτσι, είτε εγγράφονται είτε διαβάζονται δεδομένα των θέσεων αυτών μεταφερόμενα στις γραμμές D0-D3. Η εγγραφή και η ανάγνωση καθορίζονται από τα σήματα \overline{CS} , \overline{RD} , \overline{WE} . ‘Όταν το $CS' = 0$, τότε ενεργοποιείται η λειτουργία της μνήμης. Εάν τα σήματα \overline{WE} και \overline{RD} έχουν τις τιμές 0 και 1 αντίστοιχα, τότε θα ενεργοποιηθούν οι απομονωτές με μπλε χρώμα και θα έχουμε εγγραφή στη μνήμη. Ενώ, για να έχουμε διάβασμα από τη μνήμη, τότε τα σήματα \overline{WE} και \overline{RD} θα πρέπει να έχουν τις τιμές 1 και 0 αντίστοιχα.

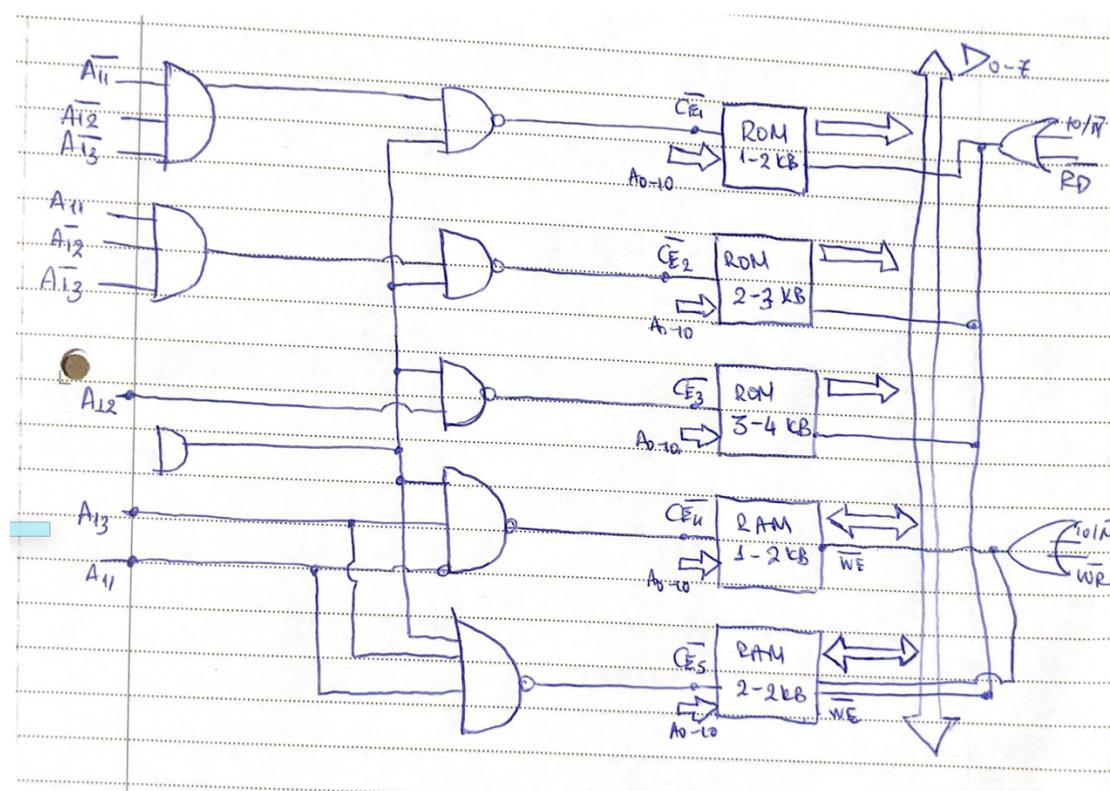
Άσκηση 6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Memory
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A000	ROM1-2K
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	A7FF	ROM1-2K
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	A800	ROM2-4K
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	AFFF	ROM2-4K
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	B000	ROM3-4K
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	BFFF	ROM3-4K
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	C000	RAM1-2K
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	C7FF	RAM1-2K
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	C800	RAM2-2K
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	CFFF	RAM2-2K

A)



B)



Άσκηση 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	Memory
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1000	ROM1-2K
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	17FF	ROM2-2K
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1800	ROM3-4K
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF	ROM2-4K
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2000	ROM3-4K
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	2FFF	ROM3-4K
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3000	RAM1-2K
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	37FF	RAM1-2K
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	3800	RAM2-2K
0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	3FFF	RAM2-2K

