

Συστήματα Μικροϋπολογιστών



1η Ομάδα Ασκήσεων

Ομάδα:

Κατσιάνης Κωνσταντίνος 03120183

Κουρής Γεώργιος - 03120116

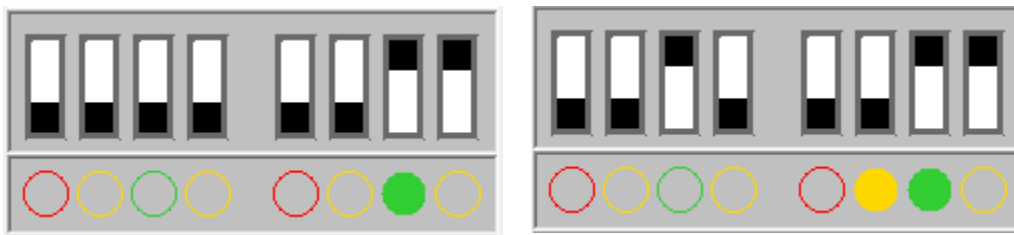
6ο Εξάμηνο - Απρίλιος 2023

Άσκηση 1:

Αρχικά αποθηκεύσαμε στο μLab το πρόγραμμα που δίνεται σε γλώσσα μηχανής στις κατάλληλες διευθύνσεις. Χρησιμοποιήσαμε τις εντολές FETCH_ADDRS και STORE/INCR. Στην συνέχεια χρησιμοποιήσαμε τον πίνακα 2 του παραρτήματος 2 των σημειώσεων κατά τη διαδικασία της αποκωδικοποίησης (disassembly). Το πρόγραμμα που τρέξαμε, μαζί τη μνήμη και τις εντολές είναι το παρακάτω:

	MVI C, 08H	0800	0E	MVI C, 08H
	LDA 2000H	0801	08	
JUMP_1:		0802	3A	LDA 2000H
	RAL	0803	00	
	JC JUMP_2	0804	20	
	DCR C			JUMP_1:
	JNZ JUMP_1	0805	17	RAL
JUMP_2:		0806	DA	JC JUMP_2
	MOV A, C	0807	0D	
	CMA	0808	08	
	STA 3000H	0809	0D	DCR C
	RST 1	080A	C2	JNZ JUMP_1
END		080B	05	
		080C	08	
				JUMP_2:
		080D	79	MOV A, C
		080E	2F	CMA
		080F	32	STA 3000H
		0810	00	
		0811	30	
		0812	CF	RST 1

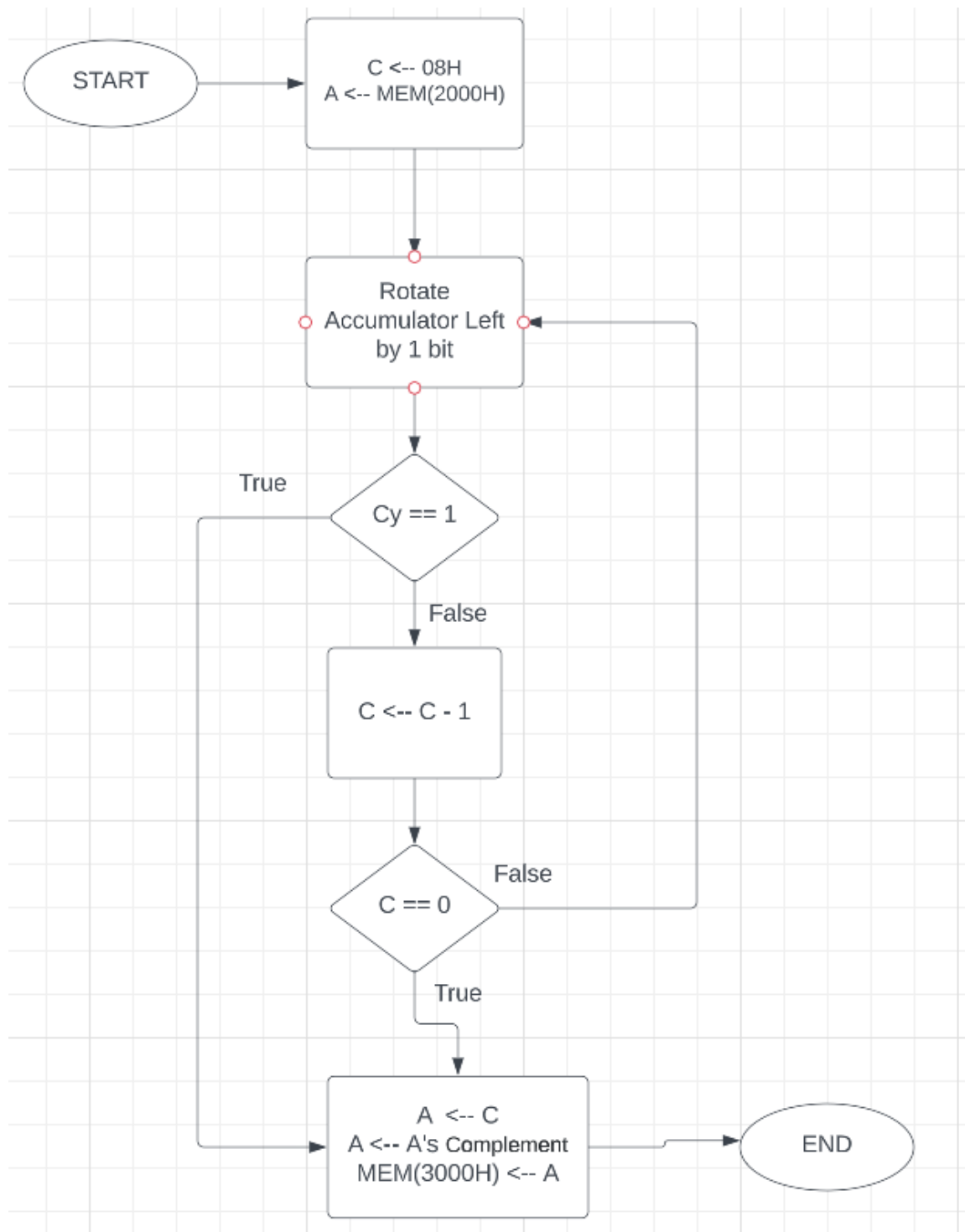
Όσον αφορά τη λειτουργία του, μας δείχνει σε δυαδική μορφή, από δεξιά προς τα αριστερά, το index απο το διακόπτη (και πάλι, μετρώντας από δεξιά προς αριστερά) που είναι πιο αριστερά.



Πρώτο παράδειγμα: Πιο αριστερός αναμμένος διακόπτης είναι ο δεύτερος, άρα ανάβει το δεύτερο λαμπάκι για $2^1 = 2$.

Δεύτερο παράδειγμα: Πιο αριστερός αναμμένος διακόπτης είναι ο έκτος, άρα ανάβουν το δεύτερο και τρίτο λαμπάκι γι $2^1 + 2^2 = 6$.

Το διάγραμμα ροής είναι:



Για να κάνουμε το πρόγραμμα να τρέχει συνεχώς, μπορούμε να αντικαταστήσουμε την εντολή RST 1 με ένα JMP στην διεύθυνση 0800. Χρησιμοποιούμε την ετικέτα START. Ο ανανεωμένος κώδικας:

```
START:      MVI C,08H
            LDA 2000H

JUMP_1:     RAL
            JC  JUMP_2
            DCR C
            JNZ JUMP_1

JUMP_2:     MOV A,C
            CMA
            STA 3000H
            JMP START

END
```

START:		
0800	0E	MVI C,08H
0801	08	
0802	3A	LDA 2000H
0803	00	
0804	20	
JUMP_1:		
0805	17	RAL
0806	DA	JC JUMP_2
0807	0D	
0808	08	
0809	0D	DCR C
080A	C2	JNZ JUMP_1
080B	05	
080C	08	
JUMP_2:		
080D	79	MOV A,C
080E	2F	CMA
080F	32	STA 3000H
0810	00	
0811	30	
0812	C3	JMP START
0813	00	
0814	08	

Άσκηση 2:

```

        LXI B,01F4H    ;dealy of 500 ms
        MVI E,FEH      ;LSB is the first one that turns on (11111110)
BEGIN:
        LDA 2000H      ;load status from switches to accumulator
        CALL DELB       ;delay of 500ms
        RRC             ;rotate accumulator right
        JNC BEGIN       ;if Cy == 1 FALSE then start again from BEGIN
        RLC             ;rotate accumulator to the left (restore to first state)
        RLC             ;rotate accumulator left
        JNC LEFT        ;if Cy == 1 FALSE then jump to LEFT
        JMP RIGHT       ;jump to RIGHT (happens when Cy == 1 TRUE)
LEFT:
        MOV A,E         ;store in the accumulator the state of the LED lights
        STA 3000H       ;store accumulator content to memory (light the LEDs)
        RLC             ;rotate accumulator (and light) to the left
        MOV E,A         ;store the accumulator content in E
        JMP BEGIN       ;jump to BEGIN
RIGHT:
        MOV A,E         ;store in the accumulator the state of the LED lights
        STA 3000H       ;store accumulator content to memory (light the LEDs)
        RRC             ;rotate accumulator (and light) to the right
        MOV E,A         ;store the accumulator content in E
        JMP BEGIN       ;jump to BEGIN
END      ;never gets reached, lights continue to move
```

Άσκηση 3:

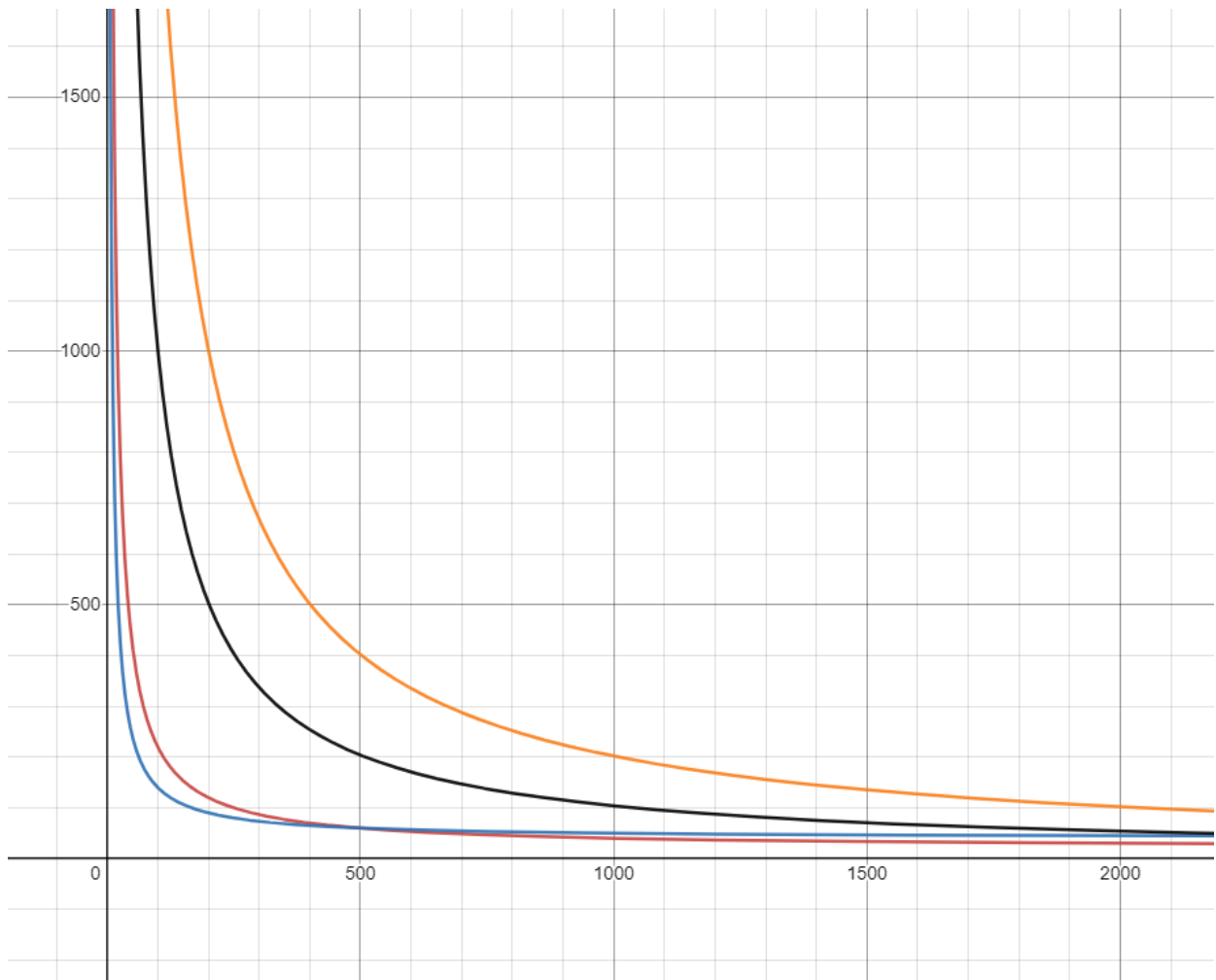
```
BEGIN:
    LDA 2000H ;load status of dip switches to the accumulator
    CPI 64H   ;store in Cy: accumulator >= 100 (64H is 100d)
    JNC LSB_OVERFLOW ;if Cy == 1 (accumulator > 99) jump to LSB_OVERFLOW
    MVI B,FFH ;FFH complement is -1
DECA:
    INR B      ;increase B by 1
    SUI 0AH    ;subtraction of 10 from accumulator
    JNC DECA   ;if Cy == 0 (A-10>0) jump to DECA (continues until A-10<0)
    ADI 0AH    ;add 10 to the accumulator so that it is positive
    MOV E,A    ;store the accumulator content in E (LED lights)
    MOV A,B    ;store the content of B in the accumulator
    RLC        ;rotate accumulator to the left four times
    RLC        ;so that we can save the value of ones in
    RLC        ;the four LSB led lights
    RLC        ;
    ADD E      ;add content of E in the accumulator
    CMA        ;save in the accumulator its complement
    STA 3000H  ;save A to the output (light is ON at 0)
    JMP BEGIN  ;jump to BEGIN
LSB_OVERFLOW:
    CPI C8H    ;store in Cy: accumulator >= 200 (C8H is 200d)
    JNC MSB_OVERFLOW ;if Cy == 1 (accumulator>199) jump to MSB_OVERFLOW
    MVI A,FOH  ;save value 11110000 in the accumulator
    STA 3000H  ;save A to the output (light is ON at 0)
    MVI A,FFH  ;save value 11111111 in the accumulator
    STA 3000H  ;save A to the output (light is ON at 0)
    JMP BEGIN  ;jump to BEGIN
MSB_OVERFLOW:
    MVI A,0FH  ;save value 00001111 in the accumulator
    STA 3000H  ;save A to the output (light is ON at 0)
    MVI A,FFH  ;save value 11111111 in the accumulator
    STA 3000H  ;save A to the output (light is ON at 0)
    JMP BEGIN  ;jump to BEGIN
END
```

Άσκηση 4:





Αρχικά, χρησιμοποιούμε τα δεδομένα της άσκησης για να κατασκευάσουμε τις κατάλληλες συναρτήσεις:

- 1) $f(x) = (20000+20*x)/x$
- 2) $g(x) = (10000+40*x)/x$
- 3) $h(x) = (100000+4*x)/x$
- 4) $k(x) = (200000+2*x)/x$

Οι καμπύλες που παίρνουμε από αυτές τις συναρτήσεις είναι οι ακόλουθες:



Στον οριζόντιο άξονα έχουμε τον αριθμό των τεμαχίων και στο κατακόρυφο το κόστος.

1		$f = \frac{(20000 + 20 \cdot x)}{x} \{x > 0\}$
2		$g = \frac{(10000 + 40 \cdot x)}{x} \{x > 0\}$
3		$h = \frac{(100000 + 4 \cdot x)}{x} \{x > 0\}$
4		$k = \frac{(200000 + 2 \cdot x)}{x} \{x > 0\}$

Για τις γραφικές παραστάσεις έγινε χρήση της ιστοσελίδας desmos.

Στη συνέχεια βρίσκουμε τα σημεία τομής των γραφικών παραστάσεων για να υπολογίσουμε τα τεμάχια στα οποία έχουμε το ίδιο κόστος για κάθε ζεύγος.

$$f(x) = g(x) \Leftrightarrow x = 500, \text{ κόστος} = 60$$

$$f(x) = h(x) \Leftrightarrow x = 5000, \text{ κόστος} = 24$$

$$f(x) = k(x) \Leftrightarrow x = 10000, \text{ κόστος} = 22$$

$$g(x) = h(x) \Leftrightarrow x = 2500, \text{ κόστος} = 44$$

$$g(x) = k(x) \Leftrightarrow x = 5000, \text{ κόστος} = 42$$

$$h(x) = k(x) \Leftrightarrow x = 50000, \text{ κόστος} = 6$$

Χρησιμοποιώντας τις παραπάνω τιμές που βρήκαμε και παρατηρώντας τις μορφές των γραφικών παραστάσεων, βρίσκουμε τις τέσσερις περιοχές αριθμού τεμαχίων που είναι συμφερότερης (χαμηλότερου κόστους) για την κάθε μία τεχνολογία:

- 1) $500 < x < 5000$
- 2) $0 < x < 500$
- 3) $5000 < x < 50000$
- 4) $x > 50000$

Για να εξαφανιστεί η επιλογή της πρώτης τεχνολογίας θα πρέπει το καινούριο κόστος της δεύτερης τεχνολογίας να είναι πάντα μικρότερο της πρώτης. Συγκεκριμένα, θα πρέπει να ισχύει η σχέση:

$$\text{κόστος 2 καινούριο} < \text{κόστος 1} \Leftrightarrow 10000 + (\kappa\kappa + 10) \cdot x < 20000 + 20 \cdot x$$

$$\Leftrightarrow \text{κόστος κατασκευής} = \kappa\kappa < (10000/x) + 10$$

Όταν τα τεμάχια τείνουν στο άπειρο έχουμε $\kappa\kappa < 10$. Φυσικά, δεν μπορούμε να έχουμε άπειρα τεμάχια, οπότε μας αρκεί να έχουμε $\kappa\kappa \leq 10$.