

Συστήματα Μικροϋπολογιστών

2^η ομάδα ασκήσεων

Νασοπούλου Ελένη – 03121087

Άσκηση 1

Ερώτημα 1

Αποθηκεύουμε τους αριθμούς 0-127 με αύξουσα σειρά στις διαδοχικές θέσεις της μνήμης με αρχή τη διεύθυνση 0900H μέσω προγράμματος. Η ζητούμενη λειτουργία έγινε σωστά, όπως φαίνεται παρακάτω στην μνήμη RAM του επεξεργαστή.

08F8	00	08F9	00	08FA	00	08FB	00	08FC	00	08FD	00	08FE	00	08FF	00
0900	00	0901	01	0902	02	0903	03	0904	04	0905	05	0906	06	0907	07
0908	08	0909	09	090A	0A	090B	0B	090C	0C	090D	0D	090E	0E	090F	0F
0910	10	0911	11	0912	12	0913	13	0914	14	0915	15	0916	16	0917	17
0918	18	0919	19	091A	1A	091B	1B	091C	1C	091D	1D	091E	1E	091F	1F
0920	20	0921	21	0922	22	0923	23	0924	24	0925	25	0926	26	0927	27
0928	28	0929	29	092A	2A	092B	2B	092C	2C	092D	2D	092E	2E	092F	2F
0930	30	0931	31	0932	32	0933	33	0934	34	0935	35	0936	36	0937	37
0938	38	0939	39	093A	3A	093B	3B	093C	3C	093D	3D	093E	3E	093F	3F
0940	40	0941	41	0942	42	0943	43	0944	44	0945	45	0946	46	0947	47
0948	48	0949	49	094A	4A	094B	4B	094C	4C	094D	4D	094E	4E	094F	4F
0950	50	0951	51	0952	52	0953	53	0954	54	0955	55	0956	56	0957	57
0958	58	0959	59	095A	5A	095B	5B	095C	5C	095D	5D	095E	5E	095F	5F
0960	60	0961	61	0962	62	0963	63	0964	64	0965	65	0966	66	0967	67
0968	68	0969	69	096A	6A	096B	6B	096C	6C	096D	6D	096E	6E	096F	6F
0970	70	0971	71	0972	72	0973	73	0974	74	0975	75	0976	76	0977	77
0978	78	0979	79	097A	7A	097B	7B	097C	7C	097D	7D	097E	7E	097F	7F
0980	00	0981	00	0982	00	0983	00	0984	00	0985	00	0986	00	0987	00

Ερώτημα 2

Υπολογίζουμε το πλήθος των δυαδικών ψηφίων ‘1’ των παραπάνω δεδομένων μέσω προγράμματος και αποθηκεύουμε το αποτέλεσμα στον διπλό καταχωρητή BC, το οποίο είναι ίσο με 01C0H, δηλαδή 448.

Το αποτέλεσμα αυτό μπορεί να επαληθευτεί και με πράξεις με το ακόλουθο σκεπτικό. Οι αριθμοί από 0-127 μπορούν να αναπαρασταθούν ως εξής: 00000000XXXXXX. Συνεπώς μας ενδιαφέρουν μόνο τα τελευταία 7 bits κάθε αριθμού. Ο συνολικός αριθμός δυαδικών ψηφίων είναι 7×128 . Έτσι, λόγω συμμετρίας των δυαδικών αναπαραστάσεων, ο συνολικός αριθμός ψηφίων ‘1’ είναι $(7 \times 128) / 2 = 448$.

Ερώτημα 3

Υπολογίζουμε μέσω προγράμματος το πλήθος από τους αριθμούς 0-127 που είναι μεταξύ των αριθμών 10H και 60H περιλαμβανομένων ($10H \leq xH \leq 60H$) και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή D, το οποίο είναι 51H, δηλαδή 81.

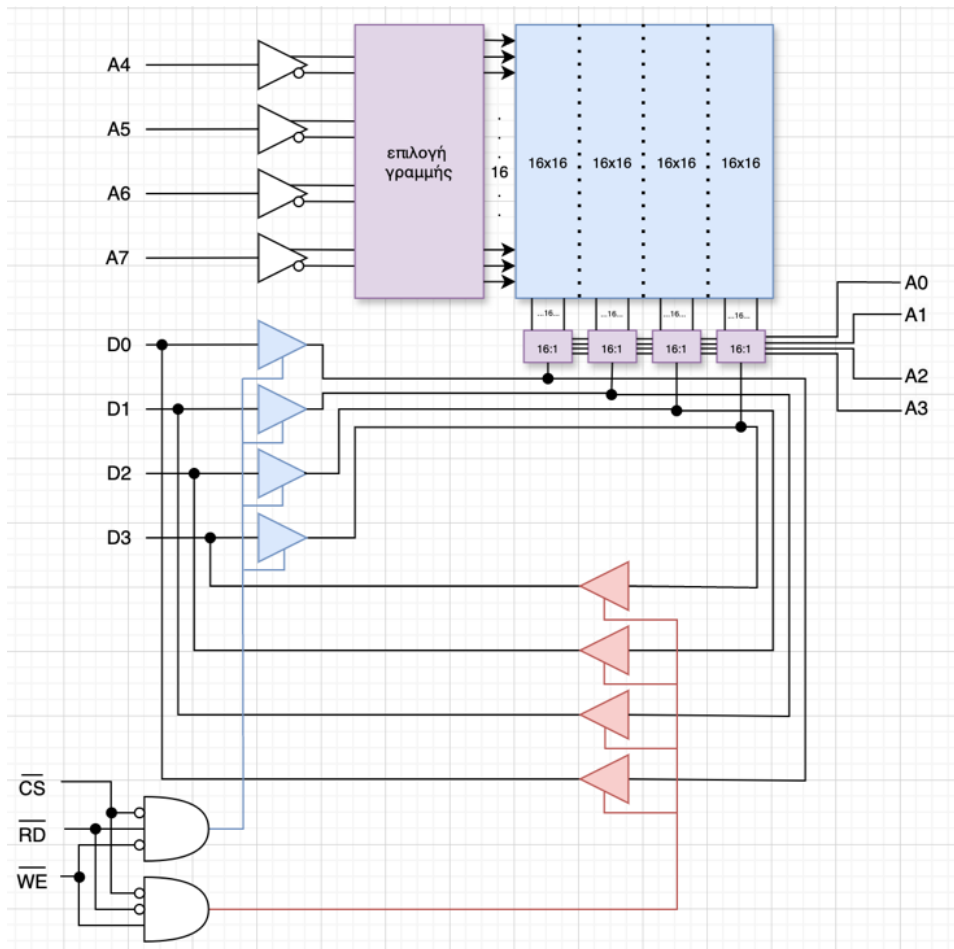
Το αποτέλεσμα αυτό μπορεί να επαληθευτεί και με πράξεις, καθώς $96 - 16 + 1 = 81$.

Το περιεχόμενο του διπλού καταχωρητή BC και του καταχωρητή D φαίνονται παρακάτω:

A	I	B	C	D	E	H	L
7F	0B	01	C0	51	00	09	7F

Άσκηση 5

Παρακάτω φαίνεται η εσωτερική οργάνωση μια μνήμη SRAM 256x4 bit.



Έχουμε $256 \times 4 \text{ bit} = 16 \times 16 \times 4 = 2^4 \times 2^4 \times 4$.

Συνεπώς, προκειμένου να ορίσουμε την διεύθυνση εγγραφής/ ανάγνωσης θα χρειαστούμε 4 bit για την επιλογή στήλης (A0-A3) και 4 bit για την επιλογή γραμμής (A4-A7).

Η λειτουργία των τριών σημάτων είναι η εξής:

Λειτουργία/ Σήμα	\overline{CS}	\overline{RD}	\overline{WE}	=> AND1	=> AND2
WRITE	0	1	0	1	0
READ	0	0	1	0	1

Συνεπώς, στην **λειτουργία εγγραφής**, η έξοδος της 1^{ης} πύλης AND είναι 1 και της 2^{ης} πύλης AND είναι 0. Αυτό έχει ως αποτέλεσμα τα buffer που συνδέονται με την AND1 (που απεικονίζονται με μπλε χρώμα) να είναι ανοιχτά, ενώ αυτά που συνδέονται με την πύλη AND2 (απεικονίζονται με κόκκινο χρώμα) να είναι κλειστά. Έτσι, μπορεί να γίνει η εγγραφή του input στη μνήμη, καθώς τα D0-D3 φέρουν την 4 bit λέξη που επιθυμούμε.

Από την άλλη πλευρά, στην **λειτουργία ανάγνωσης**, η έξοδος της 1^{ης} πύλης AND είναι 0 και της 2^{ης} πύλης AND είναι 1. Αυτό έχει ως αποτέλεσμα τα buffer που συνδέονται με την AND1 (που απεικονίζονται με μπλε χρώμα) να είναι κλειστά, ενώ αυτά που συνδέονται με την πύλη AND2 (απεικονίζονται με κόκκινο χρώμα) να είναι ανοιχτά. Έτσι, μπορεί να γίνει η ανάγνωση της επιθυμητής 4 bit λέξης από την μνήμη και η προώθηση της ως output μέσω των D0-D3.

Άσκηση 6

Τα ολοκληρωμένα που έχουμε στη διάθεσή μας φαίνονται στον παρακάτω πίνακα:

Memory	Size (bits)	Size (bytes)	
ROM1	2K x 8	2^{11}	$A_0 - A_{10}$
ROM2	2K x 8	2^{11}	$A_0 - A_{10}$
ROM3	4K x 8	2^{12}	$A_0 - A_{11}$
SRAM1	2K x 8	2^{11}	$A_0 - A_{10}$
SRAM2	2K x 8	2^{11}	$A_0 - A_{10}$

Χάρτης μνήμης του ζητούμενου συστήματος μνήμης:

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	0000H	ROM1 - 2k
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FFH	
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800H	ROM2 - 2k
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFFH	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000H	ROM3 - 4k
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	17FFH	
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1800H	
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	SRAM1 - 2k
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	27FFH	
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	2800H	SRAM2 - 2k
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	2FFFH	

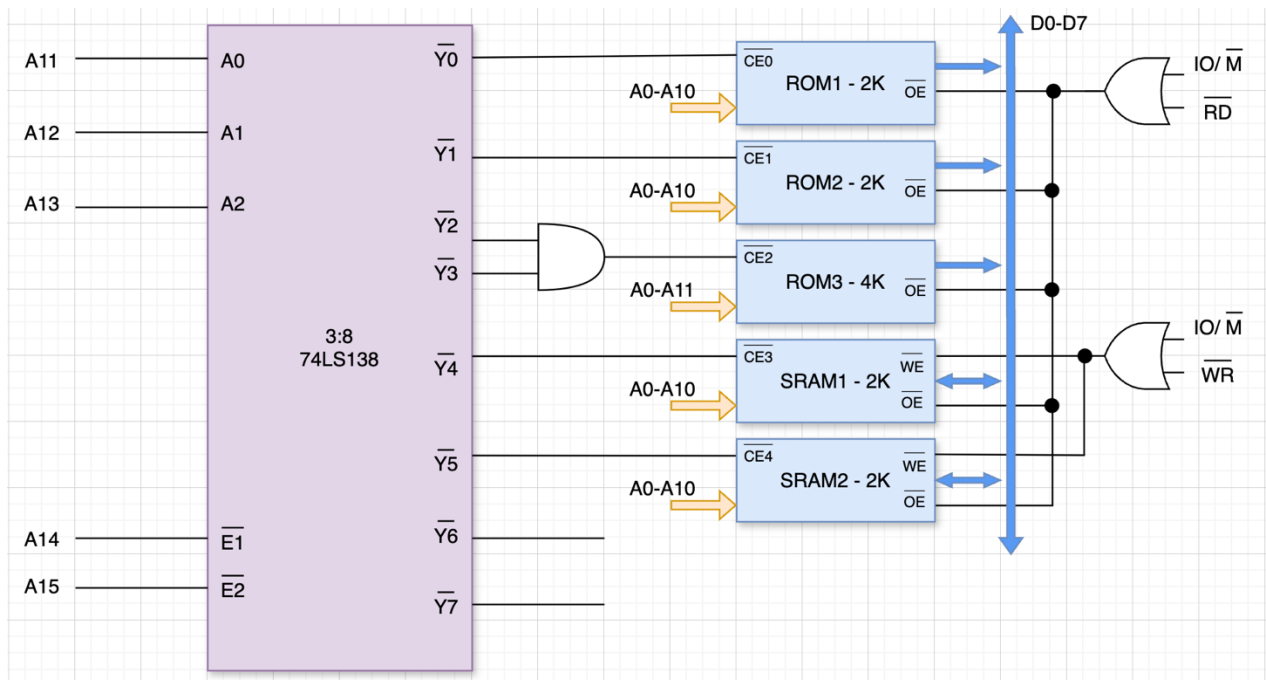
Παρατηρώντας τον χάρτη μνήμης διαπιστώνουμε πως:

- $A_{15}A_{14} = 00$ πάντα => θα χρησιμοποιηθούν ως επέκταση
- $A_{13}A_{12}A_{11}$ καθορίζουν το ολοκληρωμένο

Memory	$A_{13}A_{12}A_{11}$	\overline{CE}
ROM1	000	\overline{Y}_0
ROM2	001	\overline{Y}_1
ROM3	010 / 011	$\overline{Y}_2 * \overline{Y}_3$
SRAM1	100	\overline{Y}_4
SRAM2	101	\overline{Y}_5

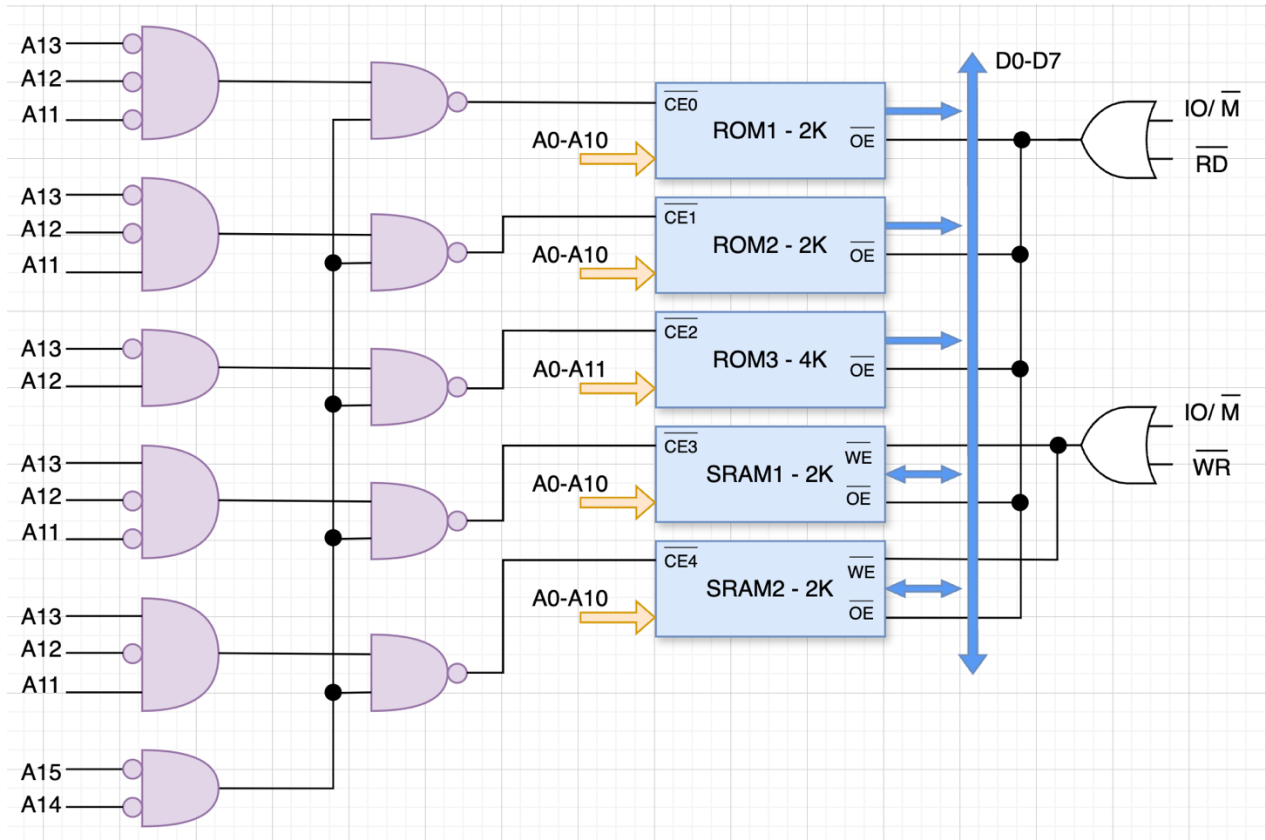
Ερώτημα 1

Υλοποίηση με αποκωδικοποιητή 3:8 (74LS138) και λογικές πύλες:



Ερώτημα 2

Υλοποίηση μόνο με λογικές πύλες:



Άσκηση 7

Τα ολοκληρωμένα που έχουμε στη διάθεσή μας φαίνονται στον παρακάτω πίνακα:

Memory	Size (bytes)	
ROM1	$16K = 2^{14}$	$A_0 - A_{13}$
SRAM1	$4K = 2^{12}$	$A_0 - A_{11}$
SRAM2	$4K = 2^{12}$	$A_0 - A_{11}$
SRAM3	$4K = 2^{12}$	$A_0 - A_{11}$

Χάρτης μνήμης του ζητούμενου συστήματος μνήμης:

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	0000H	ROM1 - 8k
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFFH	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000H	
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	SRAM1 - 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	SRAM2 - 4k
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000H	SRAM3 - 4k
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	4FFFH	
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	5000H	ROM2 - 8k
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	5FFFH	
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000H	
0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	6FFFH	

Παρατηρώντας τον χάρτη μνήμης διαπιστώνουμε πως:

- $A_{15} = 0$ πάντα \Rightarrow θα χρησιμοποιηθεί ως επιτροπή
- $A_{14}A_{13}A_{12}$ καθορίζουν το ολοκληρωμένο

Memory	$A_{14}A_{13}A_{12}$	\overline{CE}
ROM	000/001/101/110	$\bar{Y}_0 * \bar{Y}_1 * \bar{Y}_5 * \bar{Y}_6$
SRAM1	010	\bar{Y}_2
SRAM2	011	\bar{Y}_3
SRAM3	100	\bar{Y}_4

