

Θέματα στα Συστήματα Μικροϋπολογιστών—Εξέταση Εργαστηρίου

- 1) Σε μια θέση μνήμης βάλε μια τιμή. Βάλε την ίδια τιμή και στις 63 επόμενες θέσεις μνήμης. Μετά διάβασε αυτά που έβαλες εκεί με ένα loop και δες αν είναι σωστά. (Ουσιαστικά αυτό είναι ένα τεστ για να δεις αν λειτουργεί καλά η μνήμη. Το δες αν είναι σωστά, σημαίνει "σύγκρινε τις τιμές στη μνήμη με την τιμή που έχεις σε ένα καταχωρητή".)
- 2) Να γίνει το $4x+y$ με x, y 8bit και το αποτέλεσμα σε 16bit.
- 3) Αφαίρεση του x από το y σε μια ομάδα από θέσεις μνήμης.
- 4) Να βρέθει η ισοτιμία ενός byte και ανάλογα να βάλεις ένα bit 0 ή 1 στην πρώτη θέση.
- 5) Σε ένα μπλοκ μνήμης 128 θέσεων να διαγραφεί το 30ο το 50ο και το 70ο αρχείο
- 6) Έστω ότι υπάρχει σε κάποια διεύθυνση μνήμης ένας πίνακας θετικών αριθμών με 8 στοιχεία (από 8 bit το καθένα). Να υπολογίσετε το άθροισμα των 2ου, 4ου, 6ου και 8ου.
- 7) Έχουμε έναν πίνακα 128 θέσεων (128 byte). Να βρεθεί ο αριθμός των περιττών στοιχείων.
- 8) Να προσθεσετε 2 αριθμούς των 64bit
- 9) Πρόσθεση δύο αριθμών πχ $A89B69 + F9A9BC$
- 10) Να προσθέσετε 16 καταχωρητές. (Ένας τρόπος είναι να βάλεις όλους τους αριθμούς στους καταχωρητές 16 ως 31 και να κρατάς το άθροισμα σε 2 από αυτούς και ο δεύτερος (που έπαιρνε και δέκα) είναι να θεωρήσεις ότι οι καταχωρητές 0-15 είναι στη διεύθυνση \$00 και με `ld r16, x+` να φορτώνεις το περιεχόμενο καθενός και να το προσθέτεις στον r16, με βρόχο δλδ.)
- 11) Σε ένα μπλοκ 64 θέσεων RAM να απαριθμηθούν οι λέξεις των οποίων το bit 4 είναι 1 και το αποτέλεσμα να εμφανισθεί στο Port εξόδου.
- 12) Να γίνει πρόσθεση 2 αριθμών των 64bit χρησιμοποιώντας δεικτοδότημένη προσπέλαση
- 13) Χρησιμοποιήστε τη δεικτοδοτημένη προσπέλαση για να εμφανίσετε το περιεχόμενο των καταχωρητών R0 έως R7 στα LED 0-7 με διαδοχικές πιέσεις του SW7.
- 14) Να υπολογισθεί το bit άρτιας ισοτιμίας 7-bitου αριθμού στον R16.
- 15) Έστω πίνακας n bytes στην SRAM, να βρεθούν πόσα στοιχεία είναι αρνητικά και πόσα περιττά
- 16) Να αποσπάσετε τα 4 MSB από ένα byte, τα 4 LSB από ένα άλλο byte, και να τα τοποθετήσετε σε έναν register.

16) Δίνονται 2 πίνακες με n στοιχεία ($n \leq 256$). Να συγκριθούν 1-1 τα στοιχεία τους και αν είναι ίσα το αντίστοιχο στοιχείο ενός τρίτου πίνακα να παίρνει την τιμή 0, αλλιώς να παίρνει την τιμή 0xFF.

17) Έχουμε 32 θέσεις (bytes) συνεχόμενα (στην SRAM). Να βρείτε πόσους άσσους έχουν όλα τα bit 8 επί 32.

18) να γίνει ο εξής έλεγχος μιας SRAM για 256 θέσεις:

σε κάθε θέση αποθηκεύεται πρώτα ο 0xAA και γίνεται έλεγχος αν όντως αποθηκεύτηκε σωστά.

Αν ναι στην ίδια θέση μνήμης αποθηκεύεται ο 0x55 και γίνεται ο ίδιος έλεγχος.

Σε περίπτωση που αποτύχει ο έλεγχος τερματίζεται η διαδικασία και η σημαία V γίνεται 1.

19) Να απαριθμηθεί ο αριθμός των γραμμών σε μια γραμματοσειρά η οποία τερματίζει με το carriage return (0x0d). Τα κενά διαστήματα (0x20) δεν απαριθμούνται.

20) να βρείτε το λογικό AND μεταξύ των στοιχείων 2 πινάκων και το αποτέλεσμα να αποθηκευτεί σε έναν τρίτο πίνακα. (Οι πίνακες είναι στην SRAM.)

21) Να απαριθμηθεί ο αριθμός των bytes της σελίδας 1 (0x0100 -> 0x01FF) της static ram οι οποίοι είναι περιττοί και το αποτέλεσμα να εμφανισθεί στα leds

22) Να διαβαστούν οι ενδείξεις στα SW0-7. Να συγκριθεί με το περιεχόμενο του R0.

Αν είναι μεγαλύτερο να προστεθούν οι R1, R2

Αν είναι μεγαλύτερο να αφαιρεθούν

Αν είναι ίσα να γίνει λογικό AND και να εμφανιστεί το αποτέλεσμα στα led

23) Να δείτε αν αληθεύει η εκφραση $((r1+r2) \text{ OR } r3) \text{ AND } r4$ και αν ναι να αναψει το 1ο LED

24) Έχετε ένα block 32 byte στην SRAM, να βρείτε πόσα από τα bit σε αυτό το block είναι 0. (Για κάθε Byte κάνεις 8 ror ή rol και ελέγχεις το carry.)

25) Να βρεθεί με πλήρη ακρίβεια το αποτέλεσμα $4 * R0 + R1/2$ όπου $r0, r1$ καταχωρητές που ως περιεχόμενο έχουν προσημασμένους αριθμούς.

26) Να γράψετε προγράμμα που να περιμένει μέχρι να πατηθεί το SW0. Μετά παίρνει την είσοδο SW1-SW7, και την βγάζει στα LED1-7 καθώς και το LED0 να δείχνει την άρτια ισοτιμία του παραπάνω αριθμού

27) Στους καταχωρητές R0-R15 να γίνουν set τα bit 3 και bit 5 (τα υπόλοιπα να μένουν ως έχουν).

Λύση:

```
.cseg
.org 0
.def counter = r17
.def temp = r16
```

ldi ZH, 0x00

ldi ZL, 0x00

ldi counter, 16

loop:

ld temp, Z

sbr temp, 3

sbr temp, 5

st Z+, temp

dec counter

brne loop

ret

Προσοχή στο ότι ξεκινάμε προσπέλαση μνήμης από το 0x0000 όπου βρίσκονται οι καταχωρητές εργασίας και όχι από το 0x0100 όπου είναι η SRAM. Επίσης, χρησιμοποιούμε σαν temp έναν από τους R16-R31 για να δουλέψει σε αυτόν η εντολή sbr (set bits in register).

28) Να γίνει η απαρίθμηση των μονάδων στις λέξεις 30 έως 40 της μνήμης προγράμματος.

29) Να γίνει απαρίθμηση των μονάδων που βρίσκονται στις 16 πρώτες θέσεις της μνήμης προγράμματος.

30) Στα 32 bytes SRAM [ξεκινώντας από τη θέση 256], το bit 0 αντιστοιχεί στο bit αρτίας ισοτιμίας. Δηλαδή για κάθε byte ελέγχεις αν το πλήθος των 1 είναι αρτίο. Αν όχι, τότε βάζεις έναν ασσο στο bit 0 ώστε να γίνει αρτίο το πλήθος.

31) Να γράψετε πρόγραμμα με το οποίο υπολογίζουμε την συνάρτηση $f = (1 \& 3)^{(4|5)}$ όπου οι δείκτες 1,3,4,5 είναι τα αντίστοιχα bits σε έναν καταχωρητή. Στην προκειμένη περίπτωση είναι ο R16