ΕΡΓΑΣΤΗΡΙΟ ΒΑΣΙΚΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Δούρου Βασιλική Ευαγγελία- Α.Μ.: 1072633

Πεσκελίδης Παύλος- Α.Μ.: 1072483

**ΕΡΓΑΣΙΑ 3**

1. Ο πίνακας συμπληρώνεται όπως ακολουθεί:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **N** | **C** | **Z** | **V** |
| **@11** | 0 | 0 | 0 | 0 |
| **@12** | 0 | 0 | 0 | 0 |
| **@13** | 0 | 0 | 0 | 0 |
| **@19** | 0 | 1 | 0 | 1 |
| **@20** | 1 | 0 | 0 | 0 |
| **@21** | 0 | 1 | 0 | 0 |

**Σχόλια:**

**@11:** Στην γραμμή 11 προσθέτουμε με την εντολή ADDS το περιεχόμενο του καταχωρητή R0 (94 στο δεκαδικό) με τον εαυτό του και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2. Με το S στο τέλος της εντολής ανανεώνεται το περιεχόμενο του καταχωρητή κατάστασης, όμως, η πράξη δεν δημιουργεί κρατούμενο (C=0), δεν έχει ως αποτέλεσμα αρνητικό αριθμό (N=0), δεν προκαλεί υπερχείλιση (V=0) καθώς από την πρόσθεση δύο θετικών αριθμών και το αποτέλεσμα μας είναι θετικό (0x000000BC), και, τέλος, το αποτέλεσμα μας είναι διάφορο του μηδενός (άρα και Z=0).

**@12:** Στην γραμμή 12 προσθέτουμε με την εντολή ADDS το περιεχόμενο του καταχωρητή R1 (0x0000002F) με τον εαυτό του και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2. Με το S στο τέλος της εντολής ανανεώνουμε και πάλι το περιεχόμενο του καταχωρητή κατάστασης, όμως, η πράξη δεν δημιουργεί κάποιο κρατούμενο (C=0), δεν έχει ως αποτέλεσμα (0x0000005E) αρνητικό αριθμό (Ν=0), δεν προκαλεί πάλι κάποια υπερχείλιση καθώς η πρόσθεση δύο θετικών αριθμών έχει ως αποτέλεσμα θετικό (V=0), και το αποτέλεσμα είναι διάφορο του μηδενός (Z=0).

**@13:** Στη γραμμή 13 προσθέτουμε με την ADDS τα περιεχόμενα των καταχωρητών R0 και R1 και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R2 (0x0000008D). Με το επίθεμα S στην εντολή ανανεώνεται το περιεχόμενο του καταχωρητή κατάστασης, όμως η πράξη δεν δημιουργεί κάποιο κρατούμενο (C=0), το αποτέλεσμα είναι θετικός αριθμός (και άρα Ν=0), δεν προκύπτει υπερχείλιση αφού πάλι από την πρόσθεση δύο θετικών αριθμών το αποτέλεσμα έχει το ίδιο πρόσημο και είναι και αυτό θετικός (V=0), και το αποτέλεσμα είναι μη μηδενικό (Ζ=0).

**@19:** Στη γραμμή 19 αφαιρούμε με την SUBS από το περιεχόμενο του καταχωρητή R0 (0x80000000) το περιεχόμενο του καταχωρητή R2 (0x00000001) και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R3 (0x7FFFFFFF). Με το επίθεμα S στην εντολή ανανεώνουμε τον καταχωρητή κατάστασης. Άρα, C=1 αφού η αφαίρεση μας δεν δημιούργησε δανεικό, N=0 αφού το αποτέλεσμα μας δεν είναι αρνητικός αριθμός, καθώς το πιο σημαντικό ψηφίο είναι 7 και για να είναι αρνητικός θα έπρεπε να είναι μεγαλύτερο ή ίσο του 8, Ζ=0 αφού ο αριθμός μας δεν είναι μηδενικός, και V=1 αφού από την αφαίρεση ενός θετικού αριθμού από έναν αρνητικό (το R0 έχει ως πιο σημαντικό ψηφίο το 8, άρα είναι αρνητικός) το αποτέλεσμα είναι θετικός αριθμός και έτσι προκαλείται υπερχείλιση.

**@20:** Στη γραμμή 20 αφαιρούμε με την SUBS από το περιεχόμενο του καταχωρητή R0 (0x80000000) το περιεχόμενο του καταχωρητή R1 (0x80000080) και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R3 (0xFFFFFF80). Με το επίθεμα S στην εντολή ανανεώνουμε ξανά τον καταχωρητή κατάστασης. Άρα, C=0 αφού η αφαίρεση μας δημιούργησε δανεικό αφού (R1>R0), Ν=1 αφού ο αριθμός που προέκυψε είναι αρνητικός, καθώς έχει ως πιο σημαντικό ψηφίο το F, Ζ=0 αφού το αποτέλεσμα μας είναι μη μηδενικός αριθμός, και V=0 αφού η αφαίρεση γίνεται μεταξύ δύο αρνητικών αριθμών και έτσι δεν υπάρχει υπερχείλιση, καθώς αυτή παρατηρείται στην αφαίρεση μόνο μεταξύ αριθμών με διαφορετικό πρόσημο.

**@21:** Στη γραμμή 21 αφαιρούμε με την εντολή RSBS από το περιεχόμενο του καταχωρητή R1 (0x80000080) το περιεχόμενο του καταχωρητή R0 (0x80000000) και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή R3 (0x00000080). Με το επίθεμα S στην εντολή ανανεώνουμε το περιεχόμενο του καταχωρητή κατάστασης. Άρα, C=1, αφού στην αφαίρεση μας δεν δημιουργήθηκε δανεικό (R1>R0), Ν=0 αφού το αποτέλεσμα μας είναι θετικός αριθμός, καθώς έχει ως πιο σημαντικό ψηφίο το 0, αφού από έναν αρνητικό αριθμό αφαιρούμε έναν ακόμη μικρότερο αρνητικό αριθμό, Ζ=0 αφού ο αριθμός που προκύπτει είναι μη μηδενικός, και V=0 αφού η αφαίρεση έγινε μεταξύ δύο αρνητικών αριθμών και έτσι δεν υπάρχει υπερχείλιση.

Ο πίνακας με τα περιεχόμενα των καταχωρητών μετά από κάθε πρόσθεση και αφαίρεση είναι ο ακόλουθος:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Γραμμή** | **R0** | **R1** | **R2** | **R3** |
| **11** | 0x0000005E | 0x0000002F | 0x000000BC | - |
| **12** | 0x0000005E | 0x0000002F | 0x0000005E | - |
| **13** | 0x0000005E | 0x0000002F | 0x0000008D | - |
| **19** | 0x80000000 | 0x80000080 | 0x00000001 | 0x7FFFFFFF |
| **20** | 0x80000000 | 0x80000080 | 0x00000001 | 0xFFFFFF80 |
| **21** | 0x80000000 | 0x80000080 | 0x00000001 | 0x00000080 |

Το φαινόμενο αποκοπής δυαδικών ψηφίων του αποτελέσματος λόγω περιορισμένου εύρους καταχωρητών εμφανίζεται στην πράξη στη γραμμή 19, δηλαδή στην πράξη που παρατηρείται και η υπερχείλιση, καθώς στην αφαίρεση ενός θετικού από έναν αρνητικό το αποτέλεσμα θα έπρεπε να ήταν και αυτό ένας αρνητικός αριθμός. Όμως, στη δική μας περίπτωση το πιο σημαντικό ψηφίο του αποτελέσματος είναι το 7, γεγονός που δηλώνει ότι είναι θετικός καθώς στο δεκαεξαδικό σύστημα αν το πιο σημαντικό ψηφίο είναι μεγαλύτερο ή ίσο του 8 έχουμε αρνητικό αριθμό.

Στις υπόλοιπες προσθέσεις και αφαιρέσεις των οποίων τα αποτελέσματα αποθηκεύονται στους R2, R3 δεν παρατηρείται κάποιο τέτοιο φαινόμενο, αφού τα αποτελέσματα επαληθεύονται με αυτά που θα έπρεπε να έχουν βγει.

1. Ο κώδικας του ερωτήματος είναι ο ακόλουθος:

.arm

.text

.global main

main:

STMDB R13!, {R0-R12, R14}

LDR R0, =Stor

LDR R1, =Stor +6

MOV R2, #0

Loop:

STRB R2, [R0]

ADD R2, R2, #1

ADD R0, R0, #1

CMP R0, R1

BLO Loop

LDMIA R13!, {R0-R12}

MOV PC, R14

.data

Stor:

.byte 0,0,0,0,0,0

Συγκεκριμένα, στον κώδικα, στην αρχή φορτώνουμε στον καταχωρητή R0 τη διεύθυνση της ετικέτας Stor, ενώ στον R1 τη διεύθυνση της θέσης μνήμης Stor+6, ώστε να σταματήσουμε τη δομή επανάληψης, που φαίνεται στη συνέχεια, όταν το R0 αποκτήσει αυτή την τιμή. Έπειτα, μεταφέρουμε στον καταχωρητή R2 την τιμή 0 στο δεκαδικό. Στη συνέχεια, ακολουθεί μία δομή επανάληψης, στην οποία μεταφέρουμε το περιεχόμενο του καταχωρητή R2 ως byte στη θέση μνήμης που έχει ως περιεχόμενο ο καταχωρητής R0,αυξάνουμε κατά 1 την τιμή που έχει ως περιεχόμενο ο R2 και αυξάνουμε και κατά 1 την τιμή που έχει ο R0.

1. Ο κώδικας του ερωτήματος είναι ο ακόλουθος:

.arm

.text

.global main

main:

STMDB R13!, {R0-R12, R14}

LDR R0, =Stor

LDR R1, =Stor +6

MOV R2, #1

MOV R3, #2

STRB R2, [R0]

STRB R3, [R0, #1]!

Loop:

ADD R4, R3, R2

STRB R4, [R0, #1]!

MOV R2, R3

MOV R3, R4

CMP R0, R1

BLO Loop

LDMIA R13!, {R0-R12}

MOV PC, R14

.data

Stor:

.byte 0,0,0,0,0,0

Συγκεκριμένα, στον παραπάνω κώδικα, στην αρχή φορτώνουμε στον καταχωρητή R0 τη διεύθυνση της ετικέτας Stor, ενώ στον R1 τη διεύθυνση της θέσης μνήμης Stor+6, ώστε να σταματήσουμε τη δομή επανάληψης, που φαίνεται στη συνέχεια, όταν το R0 γίνει ίσο με το R1. Έπειτα, μεταφέρουμε στον καταχωρητή R2 την τιμή 1, αφού είναι ο πρώτος όρος της ακολουθίας μας, και στον R3 την τιμή 2, αφού είναι ο δεύτερος όρος. Μετά αποθηκεύουμε το περιεχόμενο του R2 στη διεύθυνση που δείχνει ο καταχωρητής R0, και του R3 στη θέση που δείχνει ο R0 + 1, και ανανεώνουμε και το περιεχόμενο του R0.Στη συνέχεια, ξεκινάει η δομή επανάληψης όπου προσθέτουμε τα περιεχόμενα των R2, R3 και αποθηκεύουμε το αποτέλεσμα στον R4, μεταφέρουμε το περιεχόμενο του R4 στη διεύθυνση που δείχνει το περιεχόμενο του R0 αυξημένο κατά 1 και ανανεώνουμε και το περιεχόμενο του R0, και, τέλος,μεταφέρουμε το περιεχόμενο του R3 στον R2 και του R4 στον R3.