Εργαστηριακή άσκηση εργαστηριού Βασικά Θέματα Αρχιτεκτονικής Υπολογιστών 1

Τμήμα:Α

Ομάδα:

Σαριδακης Γεώργιος ΑΜ 1072478

Σκαγκος Ιωάννης ΑΜ 1072611

Εντολές: Περιγραφή:

|  |  |
| --- | --- |
| **1 .arm** | Αποτελεί ντιρεκτίβα, σηματοδοτεί την έναρξη του κωδικά, πρέπει να τοποθετείται στην αρχή του προγράμματος ενημερώνει τον as ότι πρέπει να παράγει κώδικα 32 bit. |
| **2 .text** | Αποτελεί ντιρεκτίβα, τοποθετεί τον κώδικα που ακολουθεί σε συγκεκριμένη θέση. |
| **3 .global main** | Αποτελεί ντιρεκτίβα,είναι καθολικά προσπελάσιμο σύμβολο δηλαδή η main είναι προσπελάσιμη και από άλλα προγράμματα. |
| **5 main:** | Σηματοδοτεί την έναρξη του κυρίου μέρους του προγράμματος. |
| **6 STMDB R13!, {R0-R12, R14}** @Ε0 | Αποθηκεύει τους καταχωρητες που θα χρησιμοποιηθούν στην εξωτερική μνήμη. |
| **8 MOV R0, #0x20** @Ε1 | Η MOV μεταφέρει στον καταχωρητη R0 την τιμή 0x20. |
| **9 MOV R1, R0, LSL #2** @Ε2 | Η MOV μεταφέρει το περιεχομενο του R0 στον R1 μετατοπισμένο αριστερά κατά 2 θέσεις λόγο της εντολής LSL. |
| **10 MVN R2, R1, LSL #1** @Ε3 | Η MVN μεταφέρει το συμπλήρωμα ως προς 1 μετατοπισμένο αριστερά 1 θέση του καταχωρητη R1 στον καταχωρητη R2. |
| **12 LDR R3, =Values** @Ε4 | Εισάγουμε στον R3 την διεύθυνση που έχει η ετικέτα values. |
| **13 LDR R4, [R3], #4** @Ε5 | Η LDR μεταφέρει το περιεχόμενο του καταχωρητη R3 στον καταχωρητη R4 και αποθηκεύει την υπάρχουσα τιμή του R3 + 4 στον R3. |
| **14 LDRB R5, [R3], #2** @Ε6 | Η LDRB μεταφέρει ένα byte από τον R3 στον καταχωρητη R5 και αποθηκεύει 0x00008140 + 2 στον R3 |
| **15 LDRSH R6, [R3], #2** @Ε7 | Η LDRSH μεταφέρει halfword στον R6 και αποθηκεύει στον R3 το περιεχόμενο του + 2. |
| **16 LDR R3, =Stack** @Ε8 | Η LDR μεταφέρει το 32 ψηφίων δεδομένο (σε δεκαεξαδικό) stack από την εξωτερική μνήμη στον καταχωρητη R3. |
| **17 STMIA R3!, {R0-R2, R4-R6}** @Ε9 | Η STMIA μεταφέρει το block R0-R2, R4-R6 από τους καταχωρητες στην εξωτερική μνήμη και λόγο του STMIA R3! Ο R3 γίνεται R3+4\*6 (όπου 6 είναι οι καταχωριτες). |
| **18 LDMDB R3!, {R0-R2}** @Ε10 | Η LDMDB μεταφέρει το block που αποθηκεύτηκε στην προηγούμενη εντολή στην εξωτερική μνήμη στους καταχωρητες R0 R1 R2 και επιπλέον λόγο του DB και του ! η R3 μειώνεται κατά 4\*3 (όπου 3 είναι ο αριθμός των καταχωρητων). |
| **19 LDMDB R3!, {R4-R6}** @Ε11 | Η LDMDB μεταφέρει το block που αποθηκεύτηκε στην Ε9 στην εξωτερική μνήμη στους καταχωρητες R4 R5 R6 και η R3 μειώνεται κατά 4\*3 (όπου 3 είναι ο αριθμός των καταχωρητων) . |
| **21 LDMIA R13!, {R0-R12, PC}** @Ε12 | Επαναφέρει τις αρχικές τιμές στους καταχωρητες που τυχόν χρησιμοποιήθηκαν. |
| **23 .data 24 Values:** | Ορίζουμε ότι τα παρακάτω δεδομένα ϑα τοποθετηθούν στην περιοχή "data" που ορίζει αυτόματα το λειτουργικό σύστημα, για τα δεδομένα των προγραμμάτων. |
| **25 .word 0xCAFEBABA** | Αποτελει ντιρεκτίβα, τοποθετεί στην μνήμη με όνομα values το δεδομένο 0xCAFEBABA. |
| **26 .word 0x82345678** | Αποτελει ντιρεκτίβα, τοποθετεί στην μνήμη με όνομα values το δεδομένο 0x82345678. |
| **27 Stack:** | Τα παρακάτω δεδομένα τοποθετούνται στην μνήμη που σημειώνεται από την ετικέτα stack. |
| **28 .word 0,0,0,0** | Αποτελει ντιρεκτίβα, τοποθετεί words στην μνημης μεγεθους εως 32 bit. |
| **29 .word 0,0,0,0** | Αποτελει ντιρεκτίβα, τοποθετεί words στην μνημη μεγεθους εως 32 bit. |
|  |  |

Περιεχόμενο των καταχωρητών R0-R6, PC.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **R0** | **R1** | **R2** | **R3** | **R4** | **R5** | **R6** | **PC** |
| **E0** | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00008004 |
| **E1** | 0x00000020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00008008 |
| **E2** | 0x00000020 | 0x00000080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x0000800c |
| **E3** | 0x00000020 | 0x00000080 | 0xfffffefff | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00008010 |
| **E4** | 0x00000020 | 0x00000080 | 0xfffffefff | 0x0000813c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00008014 |
| **E5** | 0x00000020 | 0x00000080 | 0xfffffefff | 0x00008140 | 0xcafebaba | 0x00000000 | 0x00000000 | 0x00008018 |
| **E6** | 0x00000020 | 0x00000080 | 0xfffffefff | 0x00008142 | 0xcafebaba | 0x00000078 | 0x00000000 | 0x0000801c |
| **E7** | 0x00000020 | 0x00000080 | 0xfffffefff | 0x00008144 | 0xcafebaba | 0x00000078 | 0xffff8234 | 0x00008020 |
| **E8** | 0x00000020 | 0x00000080 | 0xfffffefff | 0x00008144 | 0xcafebaba | 0x00000078 | 0xffff8234 | 0x00008024 |
| **E9** | 0x00000020 | 0x00000080 | 0xfffffefff | 0x0000815c | 0xcafebaba | 0x00000078 | 0xffff8234 | 0x00008028 |
| **E10** | 0xcafebaba | 0x00000078 | 0xffff8234 | 0x00008150 | 0xcafebaba | 0x00000078 | 0xffff8234 | 0x0000802c |
| **E11** | 0xcafebaba | 0x00000078 | 0xffff8234 | 0x00008144 | 0x00000020 | 0x00000080 | 0xfffffefff | 0x00008030 |