Αναφορά Τρίτης Εργαστηριακής Άσκησης

|  |  |  |
| --- | --- | --- |
| **Main Memory** | | |
| Κώδικας εντολής | Θέση | Περιεχόμενο |
|  | m17 | 01 |
|  | m18 | 01 |
|  | m19 | 00 |
| W | m1a | 00 |
| X | m1b | 10 |
| Y | m1c | 22 |
| Z | m1d | 28 |
|  | m1e | 00 |
|  | m1f | 00 |
|  | m20 | 00 |
| LR🡪 | m21 | 00 |
|  | m22 | 00 |
|  | m23 | 00 |
|  | m24 | 00 |
|  | m25 | 00 |
|  | m26 | 00 |
|  | m27 | 00 |
| BR🡪 | m28 | 00 |
| SP 🡪 | m29 | 00 |

|  |  |  |
| --- | --- | --- |
| **Main Memory** | | |
| Κώδικας εντολής | Θέση | Περιεχόμενο |
| **LOADBR #K** | m00 | 00 |
| 28 | m01 | 28 |
| **LOADSP #K** | m02 | 01 |
| 29 | m03 | 29 |
| **LOADLR #K** | m04 | 02 |
| 21 | m05 | 21 |
| **PUSH $K** | m06 | 03 |
| 1D | m07 | 1D |
| **PUSH $K** | m08 | 03 |
| 1B | m09 | 1C |
| **PUSH $K** | m0a | 03 |
| 1C | m0b | 1B |
| **SUB** | m0c | 06 |
| **ADD** | m0d | 05 |
| **POP $K** | m0e | 04 |
| 1A | m0f | 02 |
| **Halt** | m10 | 07 |
|  | m11 | 05 |
|  | m12 | 07 |
|  | m13 | 07 |
|  | m14 | 06 |
|  | m15 | 01 |
|  | m16 | 06 |

Σπυριδάκης Ορέστης Νεκτάριος (ΑΜ:1067541)

|  |  |  |
| --- | --- | --- |
| **Mapper** | | |
| Κώδικας εντολής | Opcode/Θέση | Περιεχόμενα |
| LOADBR #K | **00000000** | 02 |
| LOADSP #K | **00000001** | 06 |
| LOADLR #K | **00000010** | 0a |
| PUSH $K | **00000011** | 0e |
| POP $K | **00000100** | 16 |
| ADD | **00000101** | 22 |
| SUB | **00000110** | 2b |
| HALT | **00000111** | 33 |

Λουδάρος Ιωάννης (ΑΜ:1067400)

Τμήμα Εργαστηρίου: Β05

Δηλωμένοι Registers

Program Counter : 1001

Base Register : 0011

Stack Pointer : 0001

Limit Register : 0111

X : 1011

Temp : 0101

Αποτέλεσμα

Μετά την εκτέλεση του προγράμματος, στην θέση 1a της κύριας μνήμης είναι αποθηκευμένος ο W= 16

**Micromemory**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **LOADBR #K** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** |
|  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  |
| PC+1->PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m02 |
| MDR+0->BR | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 0011 | xx | X | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m03 |
| PC+1->PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m04 |
| NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | X | Χ | 1 | 0 | 0 | 0 | Χ | Χ | Χ | Χ | m05 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BOOTSTRAP** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** |
|  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  |
| SW+0->PC,MAR | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 1001 | xx | x | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | m00 |
| NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 0 | x | x | x | x | m01 |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **LOADSP #K** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** | |  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  | | PC+1->PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m06 | | MDR+0->SP | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 0001 | xx | X | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m07 | | PC+1->PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m08 | | NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | X | Χ | 1 | 0 | 0 | 0 | Χ | Χ | Χ | Χ | m09 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **LOADLR #K** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** | |  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  | | PC+1->PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m0a | | MDR+0->LR | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 0111 | xx | X | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m0b | | PC+1->PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m0c | | NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | X | Χ | 1 | 0 | 0 | 0 | Χ | Χ | Χ | Χ | m0d |   Στις παραπάνω εντολές αρχικοποιούμε τους απαραίτητους δείκτες για την δομή της στίβας.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **PUSH $K** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** | |  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  | | SP-LR->NOP ,MSTATUS | xxxxx | 000 | xxx | 001 | 001 | 001 | 0111 | 0001 | xx | X | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | m0e | | BRmacroZ[setZ],PC+1->PC , MAR(+5) | 01111 | 011 | 011 | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m0f | | MDR+0->NOP,MAR | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | X | x | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | m10 | | MDR+0->X | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | xxxx | xx | X | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m11 | | SP-1->SP,MAR | xxxxx | 000 | xxx | 001 | 001 | 011 | 0001 | 0001 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m12 | | X+0->NOP,MWE | xxxxx | 000 | xxx | 100 | 000 | 001 | 1011 | xxxx | xx | x | x | 0 | 0 | 0 | 1 | x | 1 | x | x | m13 | | PC+1->PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m14 | | NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | X | Χ | 1 | 0 | 0 | 0 | Χ | Χ | Χ | Χ | m15 |   Αύξηση του PC κατά 1, και αν Ζ=1 (SP-LR=0, δηλαδή η στοίβα είναι γεμάτη) μην κάνεις τίποτα και πήγαινε στην επόμενη μακροεντολή. Στον MDR υπάρχει το Κ, το οποίο στέλνουμε στον MAR. Άρα πλέον στον MDR υπάρχει το περιεχόμενο της θέσης μνήμης με διεύθυνση, το οποίο αποθηκεύ- ουμε στον Χ. Αφαιρoύμε ένα από τον SP ώστε να τον ”ανεβάσουμε” κατά μία θέση και διευθυνσιοδοτούμε την κύρια μνήμη σε αυτή τη θέση. Γράφουμε το περιεχόμενο του καταχωρητή Χ στην θέση μνήμης με διεύθυνση Κ.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **POP $K** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** | |  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  | | BR-SP->NOP ,MSTATUS | xxxxx | 000 | xxx | 001 | 001 | 001 | 0001 | 0011 | xx | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | m16 | | BRmacroN[setN],0+1--> NOP(+9) | 01001 | 011 | 010 | 111 | 000 | 001 | xxxx | xxxx | 01 | X | x | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | m17 | | SP+0🡪NOP,MAR | xxxxx | 000 | xxx | 011 | 000 | 001 | xxxx | 0001 | xx | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | m18 | | MDR+0->X | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | xxxx | xx | X | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m19 | | PC+1🡪PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m1a | | MDR+0 🡪NOP,MAR | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | X | x | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | m1b | | X+0🡪NOP,MWE | xxxxx | 000 | xxx | 100 | 000 | 001 | 1011 | xxxx | xx | x | x | 0 | 0 | 0 | 1 | x | 1 | x | x | m1c | | SP+1🡪SP | xxxxx | 000 | xxx | 101 | 000 | 011 | 0001 | 0001 | 01 | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | m1d | | PC+1🡪PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m1e | | NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | X | Χ | 1 | 0 | 0 | 0 | Χ | Χ | Χ | Χ | m1f | | [setZ]:PC+2🡪PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 10 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m20 | | NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | X | Χ | 1 | 0 | 0 | 0 | Χ | Χ | Χ | Χ | m21 |   Αφού γίνει ο αρχικός έλεγχος για τον αν είναι άδεια η στίβα (SP>BR) , είτε αγνοούμε την εντολή (στην περίπτωση που όντως είναι άδεια), είτε στέλνουμε το κορυφαίο στοιχείο της στην διεύθηνση Κ.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **ADD** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** | |  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  | | SP+1🡪X | xxxxx | 000 | xxx | 101 | 000 | 011 | 0001 | 1011 | 01 | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | m22 | | BR-X🡪 NOP,  MSTATUS | xxxxx | 000 | xxx | 001 | 001 | 001 | 1011 | 0011 | xx | X | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | m23 | | BRmacroN[setN],0+1🡪  NOP(+5) | 00101 | 011 | 010 | 111 | 000 | 011 | xxxx | xxxx | 01 | X | x | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | m24 | | SP+0🡪NOP,MAR | xxxxx | 000 | xxx | 011 | 000 | 001 | xxxx | 0001 | xx | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | m25 | | MDR+0🡪X | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 1011 | xx | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m26 | | SP+1🡪SP,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0001 | 0001 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m27 | | MDR+X🡪NOP,MWE | xxxxx | 000 | xxx | 101 | 000 | 001 | 1011 | xxxx | xx | x | x | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m28 | | PC+1🡪PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m29 | | NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | X | Χ | 1 | 0 | 0 | 0 | Χ | Χ | Χ | Χ | m2a | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SUB** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** |
|  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  |
| SP+1🡪X | xxxxx | 000 | xxx | 101 | 000 | 011 | 0001 | 1011 | 01 | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | m2b |
| BR-X🡪 NOP,  MSTATUS | xxxxx | 000 | xxx | 001 | 001 | 001 | 1011 | 0011 | xx | X | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | m2c |
| BRmacroN[setN],0+1🡪  NOP(+5) | 00101 | 011 | 010 | 111 | 000 | 001 | xxxx | xxxx | 01 | X | x | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | m2d |
| SP+0🡪NOP,MAR | xxxxx | 000 | xxx | 011 | 000 | 001 | xxxx | 0001 | xx | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | m2e |
| MDR+0🡪X | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | xxxx | xx | X | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m2f |
| SP+1🡪SP,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0001 | 0001 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m30 |
| MDR-X🡪NOP,MWE | xxxxx | 000 | xxx | 101 | 010 | 001 | 1011 | xxxx | xx | x | x | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m31 |
| PC+1🡪PC,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | X | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m32 |
| NEXT(PC) | xxxxx | 000 | xxx | xxx | xxx | 001 | xxxx | xxxx | xx | X | Χ | 1 | 0 | 0 | 0 | Χ | Χ | Χ | Χ | m33 |

σ

Το **ζητούμενο μακροπρόγραμμα** θα είναι της μορφής:

LOADBR

LOADSP

LOADLR

PUSH

PUSH

PUSH

SUB

ADD

POP

HALT

Δηλαδή, αφού αρχικοποιηθούν οι δείκτες της στίβας και μπουν σε αυτή τα 3 στοιχεία που χρειάζονται για την πράξη, γίνεται πρώτη η αφαίρεση και μετά η πρόσθεση, ύστερα αποθηκεύεται το αποτέλεσμα.

Τόσο η ADD όσο και η SUB πραγματοποιούν αρχικά έναν έλεγχο για το αν υπάρχουν 2 στοιχεία στην στίβα για να πραγματοποιηθεί η αντίστοιχη πράξη. Αν δεν υπάρχουν οι εντολές αγνοούνται, διαφορετικά προστίθεται το κορυφαίο με το επόμενο στοιχείο, είτε αφαιρείται το δεύτερο από το πρώτο αντίστοιχα. Τα στοιχεία αφαιρούνται από την στίβα και στην κορυφή της μένει μόνο το αποτέλεσμα.