Αναφορά Πέμπτης Εργαστηριακής Άσκησης

|  |  |  |
| --- | --- | --- |
| **Main Memory** | | |
| Κώδικας εντολής | Θέση | Περιεχόμενο |
| **SHL B** | m17 | 05 |
| 0B | m18 | 0B |
| **ADD A, B** | m19 | 04 |
| 0A | m1a | 0A |
| 0B | m1b | 0B |
| **STORE A, $33** | m1c | 01 |
| 0A | m1d | 0A |
| 33 | m1e | 33 |
| **HALT** | m1f | 10 |
| FF | m20 | FF |
| FF | m21 | FF |
| FF | m22 | FF |
| FF | m23 | FF |
| FF | m24 | FF |
| FF | m25 | FF |
| FF | m26 | FF |
| FF | m27 | FF |
| FF | m28 | FF |
| FF | m29 | FF |
| X | m30 | 03 |
| Y | m31 | 02 |
| Z | m32 | 01 |
| W | m33 | 00 |

|  |  |  |
| --- | --- | --- |
| **Main Memory** | | |
| Κώδικας εντολής | Θέση | Περιεχόμενο |
| **LOAD A, $30** | m00 | 00 |
| 0A | m01 | 0A |
| 30 | m02 | 30 |
| **ADD A, $31** | m03 | 02 |
| 0A | m04 | 0A |
| 31 | m05 | 31 |
| **SUB A, $32** | m06 | 03 |
| 0A | m07 | 0A |
| 32 | m08 | 32 |
| **DEC A** | m09 | 06 |
| 0A | m0a | 0A |
| **SHL A** | m0b | 05 |
| 0A | m0c | 0A |
| **SHL A** | m0d | 05 |
| 0A | m0e | 0A |
| **LOAD B, $30** | m0f | 00 |
| 0B | m10 | 0B |
| 30 | m11 | 30 |
| **ADD B, $31** | m12 | 02 |
| 0B | m13 | 0B |
| 31 | m14 | 31 |
| **DEC B** | m15 | 06 |
| 0B | m16 | 0B |

Σπυριδάκης Ορέστης Νεκτάριος (ΑΜ:1067541)

|  |  |  |
| --- | --- | --- |
| **Mapper** | | |
| Κώδικας εντολής | Opcode/Θέση | Περιεχόμενα |
| LOAD R, $K | **00000000** | 02 |
| STORE R, $K | **00000001** | 0a |
| ADD R1 , $K | **00000010** | 12 |
| SUB R1 , $K | **00000011** | 1a |
| ADD R1 , R2 | **00000100** | 22 |
| SHL R | **00000101** | 2b |
| DEC R | **00000110** | 30 |
| halt | **00010000** | 01 |

Λουδάρος Ιωάννης (ΑΜ:1067400)

Τμήμα Εργαστηρίου: Β05

Δηλωμένοι Registers

Program Counter : 0010

X : 1000

Α : 1010

Β : 1011

Αποτέλεσμα

Μετά την εκτέλεση του προγράμματος, στην θέση 33 της κύριας μνήμης είναι αποθηκευμένος ο W= 14

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | 1000 | 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 |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **LOAD R, $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+2->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 10 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m06 | | MDR+0->NOP ,MAR | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | m07 | | MDR+0->X | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 1000 | xx | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m08 | | PC -1->NOP , MAR | xxxxx | 000 | xxx | 101 | 001 | 001 | 0010 | xxxx | 01 | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m09 | | MDR+0->NOP | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m0a | | X+0->RAMF , FEEDREG (SELB =0) | xxxxx | 000 | xxx | 100 | 000 | 011 | 1000 | xxxx | xx | x | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m0b | | PC+1->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m0c | | next(pc) | xxxxx | 000 | xxx | xxx | xxx | xxx | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 0 | x | x | x | x | m0d |   Πρώτα παίρνουμε το έντελο και το αποθηκεύουμε προσωρινά στον καταχωριτή Χ, ύστερα παίρνουμε την διεύθυνση του καταχωριτή που θέλουμε να το αποθηκεύσουμε. Με την βοήθεια του feedback register αποθηκεύουμε το περιεχόμενο του X στο αποτέλεσμα της προηγούμενης πράξης, δηλαδή στον R.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **STORE R, $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 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m0e | | MDR+0->X | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 1000 | xx | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m0f | | PC+1->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m10 | | MDR+0->NOP ,MAR | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | m11 | | X+0->NOP | xxxxx | 000 | xxx | 100 | 000 | 001 | 1000 | xxxx | xx | x | x | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m12 | | R+0->NOP ,MWE~,SELB | xxxxx | 000 | xxx | 011 | 000 | 001 | xxxx | xxxx | xx | x | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m13 | | PC+1->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m14 | | next(pc) | xxxxx | 000 | xxx | xxx | xxx | xxx | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 0 | x | x | x | x | m15 |   Αποθηκεύουμε το έντελο από τον καταχωρητή R στον Χ, ύστερα διευθυνσιοδοτούμε την μνήμη με το Κ και με την βοήθεια του feedback register γράφουμε τελικά το περιεχόμενο του R στην θέση μνήμης Κ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | Στις παραπάνω εντολές αποθηκεύουμε προσωρινά το Κ στον Χ, ύστερα χρησιμοποιούμε τον feedback reg (selb=0) ώστε να προσθέσουμε είτε να αφαιρέσουμε τα περιεχόμενα των R και X και να επιστρέψουμε το αποτέλεςμα στον Κ |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ADD R1, $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+2->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 10 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m16 |
| MDR+0->NOP ,MAR | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | m17 |
| MDR+0->X | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 1000 | xx | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m18 |
| PC -1->NOP , MAR | xxxxx | 000 | xxx | 101 | 001 | 001 | 0010 | xxxx | 01 | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m19 |
| MDR+0->NOP | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m1a |
| R-X->R,SELB | xxxxx | 000 | xxx | 001 | 000 | 011 | 1000 | xxxx | xx | x | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m1b |
| PC+1->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m1c |
| next(pc) | xxxxx | 000 | xxx | xxx | xxx | xxx | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 0 | x | x | x | x | m1d |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **SUB R, $K** | BRA | BIN | CON | I | I | I | APORT | BPORT | DDATA | SH~ | SELB | MWE~ | MARCLK | MSTATUS | LDS~ | PCE~ | CARRYE~ | MDE~ | DDATAE~ | **ADDRESS** |
|  | (4:0) | (2:0) | (2:0) | (2:0) | (5:3) | (8:6) | (3:0) | (3:0) | (1:0) |  |  |  |  |  |  |  |  |  |  |  |
| PC+2->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 10 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m1e |
| MDR+0->NOP ,MAR | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | m1f |
| MDR+0->X X=($K) | xxxxx | 000 | xxx | 111 | 000 | 011 | xxxx | 1000 | xx | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m20 |
| PC -1->NOP , MAR | xxxxx | 000 | xxx | 101 | 000 | 001 | 0010 | xxxx | 01 | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m21 |
| MDR+0->NOP | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m22 |
| R+X->R,SELB | xxxxx | 000 | xxx | 001 | 001 | 011 | 1000 | xxxx | xx | x | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m23 |
| PC+1->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 1001 | 1001 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m24 |
| next(pc) | xxxxx | 000 | xxx | xxx | xxx | xxx | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 0 | x | x | x | x | m25 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ADD R1, R2** | 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+2->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 10 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m26 |
| MDR+0->NOP | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m27 |
| R2+0->QREG , SELB | xxxxx | 000 | xxx | 011 | 000 | 000 | xxxx | xxxx | xx | x | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m28 |
| Q+0->X | xxxxx | 000 | xxx | 010 | 000 | 011 | xxxx | 1000 | xx | x | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m29 |
| PC -1->NOP , MAR | xxxxx | 000 | xxx | 101 | 001 | 001 | 0010 | xxxx | 01 | x | x | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m2a |
| MDR+0->NOP | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m2b |
| R1+X->R1 | xxxxx | 000 | xxx | 001 | 000 | 011 | 1000 | xxxx | xx | x | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m2c |
| PC+1->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m2d |
| next(pc) | xxxxx | 000 | xxx | xxx | xxx | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 0 | x | x | x | x | x | m2e |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SHL R** | 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 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m2f |
| MDR+0->NOP | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m30 |
| R+0->R,RAMU ,SH~ | xxxxx | 000 | xxx | 011 | 000 | 111 | xxxx | xxxx | xx | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m31 |
| PC+1->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m32 |
| next(pc) | xxxxx | 000 | xxx | xxx | xxx | xxx | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 0 | x | x | x | x | m33 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DEC R** | 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 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m34 |
| MDR+0->NOP | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m35 |
| R+0->QREG , SELB | xxxxx | 000 | xxx | 011 | 000 | 000 | xxxx | xxxx | xx | x | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | m36 |
| MDR+0->NOP | xxxxx | 000 | xxx | 111 | 000 | 001 | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | m37 |
| Q-1->R,SELB | xxxxx | 000 | xxx | 110 | 001 | 011 | xxxx | xxxx | 01 | x | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | m38 |
| PC+1->PC ,MAR | xxxxx | 000 | xxx | 101 | 000 | 011 | 0010 | 0010 | 01 | x | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | m39 |
| next(pc) | xxxxx | 000 | xxx | xxx | xxx | xxx | xxxx | xxxx | xx | x | x | 1 | 0 | 0 | 0 | x | x | x | x | m3a |

**Επιλεγμένες τιμές**

Όπως φαίνεται στον πίνακα της κύριας μνήμης, για να ελέγξουμε την λειτουργία του προγράμματος μας επιλέξαμε τις τιμές X=3, Y=2, Z=1

Συνεπώς αναμένουμε να γίνουν οι πρώτες 6 εντολές και να έχουμε αποθηκευμένο στον καταχωρητή 0Α το 4\*(3+2-2) ,ύστερα να εκτελεστούν οι επόμενες 4 και να έχουμε στον καταχωρηή 0Β το 2\*(3+2-1) και τελικά να τους προσθέσουμε και να προκύψει το τελικό αποτέλεσμα που θα αποθηκεύσουμε στο W με διεύθυνση 33.

w=4\*(3+2-1-1)+2\*(3+2-1)=12+8=0x14

**Μακροπρόγραμμα**

Για να υλοποιηθεί η w=4\*(X+Y-Z-1)+2\*(X+Y-1) σχεδιάσαμε το παρακάτω μακροπρόγραμμα:

LOAD A, $30

ADD A. $31

SUB , $32

DEC A

SHL A

SHL A

LOAD B, $30

ADD B, $31

DEC B

SHL B

ADD A,B

STORE A, $33

HALT