|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Assembly operation | Opcode  6 | Rs  5 | Rt  5 | Rd  5 | Sa  5 | Function  6 | Hex code |  |
| 1 | Xor $1 , $0 ,$0 | 000000 | 00000 | 00000 | 00000 | 00000 | 000110(^) |  | Initializez contorul i cu 0 |
| 2 | Lw $2 ,0($0) | 000110 | 00000(0) | 00010(2) | 00000 | 00000 | 000000 |  | A(adr de unde incepe vectorul - offset se gaseste la adresa 4 in memorie |
| 3 | Lw $3 , 4($0) | 000110 | 00000(0) | 00011(3) | 00000 | 00000 | 000100 (4) |  | Dimenisunea vectroului (n) se gaseste la adr 4 |
| 4 | Lw $4,8($0) | 000110 | 00000(0) | 00100 (4) | 00000 | 00000 | 001000(8) |  | La adr 8 am val lui x |
| 5 | Add $5 , $2 , $0 | 000000 | 00010(2) | 00000(0) | 00101(5) | 00000 | 000000(+) |  | Reg5 contine adresa la care se gasesc elem. din vectorul meu |
| 6 | L1: beq $1 , $3 , sar la final ( afara din bucla – 11 octeti(16-6 = 10) | 000011 | 00001(1) | 00011(3) | 00000 | 00000 | 001011(unde sar) |  | Sar la adr cand $1=$3 (cand am terminat de parcurs vectorul) |
| 7 | Lw $6 , 0($5) | 000110 | 000101(5) | 00110 (6) | 00000 | 00000 | 000000(0) |  | Copie a elem. A[i] |
| 8 | Slt $7 ,$6 ,$4 | 000101 | 000110(6) | 00100(4) | 00111(7) | 00000 | 000001(-) |  | In reg7 pun 0/1 in urma comp reg6 cu reg4 |
| 9 | Beq $7 , $0 ,2 (pozitia 12, 2 octeti dupa) | 000011 | 00111(7) | 00000(0) | 00000 | 00000 | 000010(adr unde sar)  pas 12 |  | Veriifc daca A[i] > x , daca da sar la adr altffel merg la op urm |
| 10 | Srl $8 , $7 , 1 | 000000 | 00000 | 00111(7) | 01000(8) | 00001(1) | 000010(>>) |  | Daca sunt aici,nu a sarit => <x , acum il /2 |
| 11 | Jump 14 (la primul ADDI de la pas 14 ) | 001001 | 000000 | 00000 | 00000 | 00000 | 001101 |  | Sar la adresa 14 |
| 12 | Sll $9 , $7 ,1 | 000000 | 00000 | 00111(7) | 01001(9) | 00001(1) | 000011(<<) |  | Elem>x => \*2 |
| 13 | Add $8 , $9 , $6 | 000000 | 01001(9) | 00110(6) | 01000(8) | 00000 | 000000(+) |  | +1 => am facut inmultire cu 3 |
| 14 | Addi $1 ,$1 ,1 | 000001 | 00001(1) | 00001(1) | 00000 | 00000 | 000001(1) |  |  |
| 15 | Addi $5 , $5 , 4 | 000001 | 00101(5) | 00101(5) | 00000 | 00000 | 000100(4) |  | Merg la urmatorul element |
| 16 | Jump 6(L1 de la pas 6) | 001001 | 000000 | 00000 | 00000 | 00000 | 000110 |  | Refac bucla pana cand i=n(poz 6 din memoria de instructiuni) |
|  |  |  |  |  |  |  |  |  |  |