***Ejercicio Nº 1:***

|  |  |
| --- | --- |
| B = C +D+A^2+(A\_D),   1. PUSH C 2. PUSH D 3. PUSH A 4. DUP 5. MPY 6. PUSH A 7. PUSH D 8. MPY 9. ADD 10. ADD 11. ADD 12. POP B | Cantidad de instrucciones: 12  Profundidad máxima alcanzada de la pila: 5 |
| B)   1. LDA RO,C 2. LD R1,(R0) 3. LDA R0,D 4. LD R2(R0) 5. LDA R0,A 6. LD R3,(R0) 7. MUL R4,R3,R3 8. MUL R5,R3,R2 9. ADD R6,R5,R4 10. ADD R6,R6,R2 11. ADD R6,R6,R1 12. LDA R7,B 13. ST (R7),R6     Cantidad instrucciones: 13  Accesos a memoria: 4 |  |
| C)   1. MOV R0,[D] 2. ADD R0,[C] 3. MOV R1,[A] 4. MOV R2,R1 5. MUL R1,R1 6. MUL R3,R2 7. ADD R1,R3 8. ADD R1,R0 9. MOV [B],R1   Cantidad de instrucciones: 9  Accesos a memoria: 4 |  |

***Ejercicio Nº 2:***

|  |  |
| --- | --- |
| X = (0 01111101 0000000111) (2^-2)  Y = (0 01111010 0000001001) (2^-5)  Shift y 2 lugares: 0.0010000001001  ***Multiplico***  x 1. 0000000111  y 0.0010000001001 .  0,0100000100001000111111  GR S = 1 = (0|0|0|1|1|1|1|1|1)  Normalizamos: 1,0000010000|0  G R; S=1  Redondeo hacia los pares: 1,0000010000 x 2^-4  Resultado final = 0 01111011 0000010000  En decimal: 0,063476563  Error = 0,00793503969907761 - 0,063476563 = 0,05554152330092239 |  |

***Ejercicio Nº 3:***

|  |  |
| --- | --- |
| X = (0 01111100 0010000000) (2^-3)  Y = (0 01111001 0000110011) (2^-6)  Shift y 3 lugares: 0.0010000110011 x 2^-3  ***Sumo***  x 1. 0010000000  y 0.0010000110011.  1.0100000110011  GR S = 1  Está normalizado.  Redondeo +infi: R ^S = 1. Tenemos que sumar 1.  Resultado = 1.0100000111 x 2^-3 = 0,157104492  Resultado final = 0 01111100 0100000111  En decimal: 0,157104492  Error = 0,157028198242187- 0,157104492= -0,000076293757813 |  |

***Ejercicio Nº 4:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (1) mov R6, #xxxx  (2) mov R6, R  (3) mov R6, (R)  (4) mov R6, xxxx  (5) mov R6, (xxxx)  (6) mov R6, (R2)xxxx  (7) mov R6, @300(R) | #xxxx Inmediato  R Registro  (R) Registro indirecto  xxxx Absoluto  (xxxx) Memoria indirecto  (R)xxxx Base  @xxxx(R) Pre-indexado indirecto | ***Paso*** | ***Valor*** | ***Accesos*** |
| ***1*** | ***xxxx = 100*** | ***0*** |
| ***2*** | ***R = R1*** | ***0*** |
| ***3*** | ***R = R4*** | ***1*** |
| ***4*** | ***xxxx = 400*** | ***1*** |
| ***5*** | ***xxxx = 200*** | ***2*** |
| ***6*** | ***xxxx = 200*** | ***1*** |
| ***7*** | ***R = R3*** | ***2*** |

***Ejercicio Nº 5:***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Programa*** | ***Inciso A: Ensamblado*** | |  | | | | | | |
| Programa A:  LDA R0, FFh  LOAD R1, 0(R0)  ADD R2, RF, RF  JZ R1, lbl2  lbl1: ADD R2, R2, R1  DEC R1  JG R1, lbl1  lbl2: STORE R2, 0(R0)  HLT  Pseudocódigo 1  if (R1 <= 4) R2++;  else R2--;  Pseudocódigo 2  R3 = 0;  for (R4 = 0; R4 < 10; R4++)  R3 += R4; | 00:  02:  04:  06:  08:  0A:  0C: 0E:  10: | 80FF  6100  02FF  9106  0221  E1XX  A1F7  7020  FXXX |  | | | | | | |
| ***Inciso B:*** | | | | | | |
| ***R0*** | ***R1*** | ***R2*** | ***PC*** |  |  |  |
| -- | -- | -- | **00** |  |  |  |
| **FF** | -- | -- | **02** |  |  |  |
| FF | **03** | -- | **04** |  |  |  |
| FF | 03 | **00** | **06** |  |  |  |
| FF | 03 | 00 | **08** |  |  |  |
| FF | 03 | **03** | **0A** |  |  |  |
| FF | **02** | 03 | **0C** |  |  |  |
| FF | 02 | 03 | **0E-08** |  |  |  |
| FF | 02 | **05** | **0A** |  |  |  |
| FF | **01** | 05 | **0C** |  |  |  |
| FF | 01 | 05 | **0E-08** |  |  |  |
| FF | 01 | **06** | **0A** |  |  |  |
| FF | **00** | 06 | **0C** |  |  |  |
| FF | 00 | 06 | **0E** |  |  |  |
| FF | 00 | 06 | **10** |  |  |  |
| FF | 00 | 06 | **--** |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | | | | | | |
| ***Inciso C:*** | | | | | | |
| LDA R4, 04 ; R4 = 4  SUB R3, R1, R4 ; R3 = R1 – R4  JG R3, false ; si (R1-4)>0 🡪 R1>4  INC R2  JMP fin  false: DEC R2  fin: HLT | | | | | | |
| ***Inciso D:*** | | | | | | |
| XOR R3, R3, R3 ; R3=0  LDA R4, 09 ; R4 = 09  loop: ADD R3, R3, R4 ; R3+=R4  DEC R4 ; R4--;  JG R4, loop ; Loop mientras R4>0  HLT | | | | | | |