***Ejercicio Nº 1:***

|  |  |  |
| --- | --- | --- |
| ***B = (C+D) + A\*A + (A\*D)*** | | |
| ***Inciso a) PILA*** | ***Inciso b) RISC*** | ***Inciso c) INTEL*** |
| 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   Profundidad Máxima: 5 | 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   Cant. Accesos: 4 | 1. MOV R0, [C] 2. MOV R1, [D] 3. ADD R0, R1 4. MOV R2, [A] 5. MOV R3, R2 6. MUL R2, R3 7. MUL R3, R1 8. ADD R0, R2 9. ADD R0, R3 10. MOV [B], R0   Cant. Accesos: 4 |

***Ejercicio Nº 2:***

|  |  |
| --- | --- |
| ***Multiplicación de X\*Y*** | |
| X = 0 01111101 0000000111 | (+) 2^(-2) \* 1.0068359375  0,2517089843 |
| Y = 0 01111010 0000001001 | (+) 2^(-5) \* 1.0087890625  0,0315246582 |
| ***Cálculo de exponente*** | ***Cálculo de mantisa*** |
| Se deben sumar los exponentes, y compensar el exceso.  01111101  01111010  11110111  Como no hay carry de salida, debo restar el exceso 127.  11110111  01111111  01111000 🡪( 120 - 127 = -7 ) | 1.0000000111  X 1.0000001001  10000000111  1000000011100  10000000111000000  1.0000010000|0000111111  GRS = 0 0 1  p0 = 0  El resultado está normalizado; no uso G, y re-computo bits R S = 0 1. |
| ***Redondeo y resultado Final*** | |
| Redondeo:  Como p0 = 0 y R = 0, aplicando el redondeo hacia pares, no se debe realizar ninguna operación extra.  Expresión Final:  X \* Y = 0 01111000 0000010000 | |
| ***Cálculo de error*** | |
| Decimal: 0,0079350396  Norma: 2^(-7) \* 1.015625  Diferencia: | 0,0079350396  0.0079345703  0.0000004693 |

***Ejercicio Nº 3:***

|  |  |
| --- | --- |
| ***Suma de X+Y*** | |
| X = 0 01111100 0010000000 | (+) 2^(-3) \* 1.125  0.1406250000 |
| Y = 0 01111001 0000110011 | (+) 2^(-6) \* 1.0498046875  0.0164031982 |
| ***Algoritmo de implementación en HW*** | |
| - Comparo los exponentes. E1 > E2 entonces no hago swap.  - Los signos no difieren, por lo tanto, no complemento ninguna mantisa.  - Ajusto los exponentes 🡪 Realizar 3-shift derecha de m2.  m2 = 1.0000110011  Shift = 0.0010000110 | 011  - De los bits descartados computo G R S = 0 1 1.  - Realizo la suma de mantisas.  1.0010000000  0.0010000110  1.0100000110  - Los signos son iguales, y no hay carry out, por lo que no se debe complementar el resultado ni normalizar. Resultado final sin redondear.  - Ajusto RS = 0 1 | |
| ***Redondeo y resultado Final*** | |
| Redondeo:  Como R = 0 y S = 1, aplicando el redondeo hacia +inf, se debe sumar 1 LSB (R OR S).  1.0100000110 + 1 = 1.0100000111  Expresión Final:  X + Y = 0 01111100 0100000111 | |
| ***Cálculo de error*** | |
| Decimal: 0,1570281982  Norma: 2^(-3) \* 1.2568359375  Diferencia: | 0,1570281982  0.1571049218  0.0000767236 |

***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 | | | | | | |