***Ejercicio Nº 1:***

En el marco de la norma IEEE 754, considerando la representación en punto flotante de

media precisión: mantisa fraccionaria en signo magnitud con hidden bit, exponente en exceso y base 2

y la siguiente distribución de bits:

**Sig (1b) Exponente (5 bits) Mantisa (10 bits)**

|  |  |  |
| --- | --- | --- |
| Sig (1b) | Exponente (5 bits) | Mantisa (10 bits) |

Dados los números:

*X* = (1 10110 0011111001)

*Y* = (0 00111 1000111100)

Realizar el producto *X* × *Y* aplicando redondeo hacia +∞ y proximidad pares, explicando cada uno de

los pasos involucrados e indicando claramente qué se hace con los bits **G, R y S** del resultado y **con R**

**y S** al redondear. El resultado debe ser expresando según la representación enunciada.

Solución:

X = (-1) \* (22-15) \* 1,0011111001

Y = (1) \* 27-15 \* 1,1000111100

-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sumo exponentes:  +  22 – 15 + 7 ~~– 15~~ + ~~15~~ =  10110 (22)  00111 (7)  11101  01111 (15)  01110 (14) | Multiplicamos mantisas:   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | |  |  |  |  |  |  |  |  |  |  | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | |  |  |  |  |  |  |  |  |  |  | 011 | 011 | 010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |  |  |  |  |  |  |  |  |  | 011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | |  |  |  |  |  |  |  |  | 110 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |  |  | |  |  |  |  |  |  |  | 110 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |  |  |  | |  |  |  |  |  |  | 110 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |  |  |  |  | |  |  |  |  |  | 110 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |  |  |  |  |  | |  |  |  |  | 010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  | |  |  |  | 01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  | |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  | |  | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |  |  |  |  |  |  |  |  |  | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  | | 0, | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0, | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |   G = 1, R = 0, S = 1  Normalizamos:  1,0110001101001011100 => R = 0 S = 1.  Ajustamos Exponente: 14 – 1 = 13   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1, | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |   Redondeo +∞ R = 0 S = 1  +1 LSB     |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1, | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |  |  |  |  | | --- | --- | --- | | Sig (1b) | Exponente (5 bits) | Mantisa (10 bits) | | 1 | 01110 | 1,0110001110 |   Redondeo proximidad pares: R = 0 S = 1 (no hay cambio)   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1, | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |   Resultado:   |  |  |  | | --- | --- | --- | | Sig (1b) | Exponente (5 bits) | Mantisa (10 bits) | | 1 | 01110 | 1,0110001101 | |

***Ejercicio 2:***

En el marco de la norma IEEE 754, considerando la representación en punto flotante de

media precisión: mantisa fraccionaria en signo magnitud con hidden bit, exponente en exceso y base 2

y la siguiente distribución de bits:

|  |  |  |
| --- | --- | --- |
| Sig (1b) | Exponente (5 bits) | Mantisa (10 bits) |

Dados los números:

*X* = (0 01101 0010110101)

*Y* = (1 01110 1101000110)

Realizar la suma *X +* *Y* aplicando redondeo por proximidad, explicando cada uno de

los pasos involucrados e indicando claramente qué se hace con los bits **G, R y S** del resultado y **con R**

**y S** al redondear. El resultado debe ser expresando según la representación enunciada.

Solución:

|  |  |
| --- | --- |
| Igualar Exponentes  X = 1 \* 2-2 \* 1.0010110101  Y = (-1) \* 2-1 \* 1.1101000110 >> 1  Y = (-1) \* 2-2 \* 0.11101000110 | Sumo exponentes:  -2 + -2 = -4 |
| Complementar y  Y 0.11101000110  1.00010111001  + 1 .  1.00010111010 |  |
| Sumar mantisas  X 01.0010110101  Y 11.0001011101 0 .  00.0100010010 0 0 0  G R S  Normalizar: << 2  1.0001001000 0 0 0  Nuevos valores R = 0, S = 0.  Redondeo proximidad unbiased (pares): p0=0 R=0 S=0  No sumar nada.  Resultado final:   |  |  |  | | --- | --- | --- | | 0 | 01101 | 0001001000 | | Es positivo, con lo cual no hay que complementar.  Se debe normalizar. Sumo 2 al exponente, con lo cual el exponente resultante es -2. |

***Ejercicio 3:***

Asumiendo que se cuenta en todos los casos con las instrucciones add y mpy. Encontrar una

secuencia de instrucciones que resulte óptima en tiempo de ejecución (es decir, que minimice la cantidad de

accesos a memoria), y cuya ejecución tenga como resultado la evaluación de la siguiente expresión aritmética:

*B* = (*A* × (C+D) ) + (*A x (C + D))3*

Las etiquetas denotan las *direcciones de memoria* que contienen los valores sobre los que se quiere operar

a) Asumiendo una arquitectura de 0–direcciones (tipo pila), con las instrucciones push y pop para acceder

a memoria y la instrucción **dup** que duplica el tope de la pila. Determinar la cantidad de instrucciones y la profundidad de la pila alcanzada.

b) Asumiendo una arquitectura estilo RISC con operaciones registro a registro, sin limitaciones en cuanto a

los registros disponibles, y las instrucciones ld (load) y st (store) para acceder a memoria, y la instrucción

lda (load address). Las operaciones aritméticas operan con dos operandos (dst/fte , fte). Indicar la cantidad de accesos a memoria requeridos.

c) Asumiendo una arquitectura tipo INTEL con operaciones 1–dirección más registros, sin limitaciones en

cuando a los registros disponibles, que en lugar de load y store cuenta con la instrucción mov para acceder

a memoria y donde las operaciones aritméticas operan con tres operandos (dst, fte, fte). Indicar la cantidad de accesos a memoria realizados.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Pila   Inst 1: PUSH A  Inst 2: PUSH C  Inst 3: PUSH D  Inst 4: ADD  Inst 5: MPY  Inst 6: DUP  Inst 7: DUP  Inst 8: DUP  Inst 9: MPY  Inst 10: MPY  Inst 11: ADD  Inst 12: POP B  Cantidad de instrucciones: 12  Profundidad de la pila alcanzada: 4 | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Pila 1: inst 1 a 3   |  | | --- | | A | | C | | D | | Pila 2. Inst 4   |  | | --- | | A | | C+D | | | Pila 3: Inst 5   |  | | --- | | A\*(C+D) | | Pila 4: Inst 6   |  | | --- | | A\*(C+D) | | A\*(C+D) | | | Pila 4: Inst 7 y 8   |  | | --- | | A\*(C+D) | | A\*(C+D) | | A\*(C+D) | | A\*(C+D) | | Pila 5: Inst 9   |  | | --- | | A\*(C+D) | | A\*(C+D) | | (A\*(C+D))^2 | | | Pila 6: Inst 10   |  | | --- | | A\*(C+D) | | (A\*(C+D))^3) | | Pila 7: Inst 11   |  | | --- | | (A\*(C+D))+ (A\*(C+D))^3 | | |
| 1. RISC registro a registro 2. LDA R0,C 3. LD R1,(R0) 4. LDA R0,D 5. LD R2,(R0) 6. LDA R0,A 7. LD R3,(R0) 8. ADD R4,R1,R2 9. MUL R5,R4,R3 10. MUL R6,R5,R5 11. MUL R6,R6,R5 12. ADD R7,R6,R5 13. LDA R8, B 14. ST (R8),R7   Cantidad de accesos a memoria: 4 | R1 <- C  R2 <- D  R3 <- A  R4 <- C+D  R5 <- A\*(C+D)  R6 <- (A\*(C+D))^3  R7 <- (A\*(C+D)) + (A\*(C+D))^3  B <- R7 |
| 1. INTEL 1 dir + reg   1: mov R0,[C]  2: ADD R0,[D]  3: MUL R0,[A]  4: MOV R1,R0  5: MUL R0,R0  6: MUL R0,R1  7: ADD R0,R1  8: MOV [B],R0 | Cantidad de instrucciones: 8  Accesos a memoria: 4 |