Dado A que representa la configuración en base 4 de un número almacenado en formato binario de punto fijo con signo de 2 bytes y B que representa la configuración en base 8 de un empaquetado de 3 bytes, se pide:

a) Indicar cuáles son los números almacenados en base 10.

b) Siendo C la división entera de su padrón por 10, calcular A-C en BPF c/s de 16 bits indicando la validez del resultado.

c) Si el resultado de la cuenta anterior es válido, representarlo en formato flotante IEEE e indicar su configuración hexadecimal. Si es inválido, hacer lo mismo pero con el número almacenado en B

Indicar claramente su padrón.

Datos:

A = 13021012(4)

B = 1201074 (8)

A = 13021012 (4) -> BPF C/S 2 BYTES

B = 1201074 (8) -> EMPQ 3 BYTES

a)

A = 13021012 (4) -> BPF C/S 2 BYTES

Config a formato

A = 13021012 (4) -> X (2)

Aplicamos regla de potencia.

4 = 2 ELEVADO AL CUADRADO

X = 2 -> bits

13021012 (4) = 0111 0010 0100 0110 (2)

¿Tiene 2 bytes? SI

Formato a numero

0111 0010 0100 0110 (2) -> BPF C/S 2 Bytes

Realizamos sumas sucesivas para pasar a base 10.

0111 0010 0100 0110 (2) = 14662(10)

Configuracion a formato.

B = 1201074 (8) -> X(16)

Primero tenemos que pasar a base 10

1201074 (8) = 4 . 8 ^0 + 7 . 8^1 + 1. 8 ^ 3 + 2 . 8 ^5 + 1.8 ^6 = 328252(10)

328252(10) -> X(16)

328252/16

12 20515 /16

3 1282/16

2 80/16

0 5

502312 = 5023C

5023C(16) -> Tiene 3 bytes? NO. Expandimos

05023C (16) EMPAQUETADO 3 BYTES.

05023C (16) -> 5023(10)

RTAS PUNTO 1: A=14662(10) B= 5023(10)

b)

C = 10917 (10)

A – C = X en BPF C/S 16 BITS

A = 0111 0010 0100 0110 (2) -> BPF C/S 16 bits

10917 (10) = X(2)

10917/16

5 682/16

10 42/16

10 2

2AA5(16) -> 0010 1010 1010 0101 (2) -> TIENE 16 BITS? SI.

C = 0010 1010 1010 0101 (2) -> BPF C/S 16 BITS

NOT:

1101 0101 0101 1010

+1:

1101 0101 0101 1011

A = 0111 0010 0100 0110 (2)

C = 1101 0101 0101 1011 (2)

Debido al NOT + 1 ahora podemos sumar los 2 binarios entre si.

11 11 11 110

+ 0111 0010 0100 0110

1101 0101 0101 1011

---------------------------------

0100 0111 1010 0001

2 ultimos acarreos IGUALES. (Es válido)

**RTA B = VALIDO**

**C)**

0100 0111 1010 0001  
Normalizamos.

0,10001111010001

Como nos desplazamos 14 veces, lo trasladamos al exponente en binario

**0,10001111010001 x 10 ^ 1111 (2) -> BPFLOTANTE**

Ahora dividimos parte por parte.

SIGNO = 0

EXC = 14 + 127 = 141(10) -> 10001101 (2)

MANTISA = 10001111010001

0 10001101 10001111010001

Agregamos los 0 que faltan para completar la precision simple.

**0 10001101 100011110100010000000 RTA.**