

## SOKET

### APLICACIÓN SOCKET SERVIDOR STREAM: "OK"- CHAR BB [256]; INT AA, DES1, DES2; antes de llenar ssl

Aplicación [socket](#) servidor stream

- Recibe datos por el [socket](#), le envía al [socket](#) cliente "ok" y muestra en pantalla lo recibido por el [socket](#).
- Puede conectarse con varios clientes en simultaneo utilizando hijos.

Completar:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>

char bb[256]; // declara vector de 256 posiciones
struct sockaddr_in cc={}; // define estructura
int aa,des1,des2; // define variables globales enteras

int main(int argc , char const * argv[]){ //declara variables locales
    argv[1]="2000"; // ingresa puerto en el vector arg[1]=2000
    add.sin_family = AF_INET; // el primer argumento de la estructura es por conexion IPv4
    add.sin_port = htons(atoi(argv[1])); //define puerto del servidor argv[1]
    add.sin_addr.s_addr = htonl(INADDR_ANY); //asigna en IP de la maquina // INADDR_ANY convierte la IP al protocolo de la computadora
    ssl=socket(AF_INET,SOCK_STREAM,0); //el descriptor que devuelve socket lo guarda en ssl
    bind(ssl,(struct sockaddr *)&add,sizeof(struct sockaddr_in)); //vincula el socket a la direccion del socket
    listen(ssl,1); // el ssl escucha al descriptor del socket y 1 cant de pendientes

    while(1) {
        ss2=accept(ssl,NULL,0); // se bloquea en la espera, NULL (no me interesa guardar la direccion del cliente), ssl lo uso para escribir y leer
        // crea proceso hijo
        if(fork()) {
            //linea en blanco
            while((aa = read(ss2,ff,sizeof(ff)) > 0) { //lee el socket ss2
                write(ss2,"ok", 2); //escribe lo leído del socket
                write(STDOUT_FILENO,ff,aa); //escribe por pantalla lo leído del socket
            }
        }
    }

    return 0;
}
```

## APLICACIÓN SOCKET SERVIDOR STREAM: "OK"- CHAR BB [256]; INT AA, DES1, DES2; antes de llenar argv [1]

Aplicación socket servidor stream

Recibe datos por el socket, le envía al socket cliente "ok" y muestra en pantalla lo recibido por el socket.

Completar:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```
char bb[256];
struct sockaddr_in cc={};
int aa,des1,des2;
```

// declara vector de 256 posiciones  
// define estructura  
//declara variables globales enteras

```
int main(int argc , char const * argv[]){
    argv[1]="2000";
```

//declara variables locales dentro del int  
//ingresa el puerto en arg[1]=2000

```
    //línea en blanco
```



```
    cc.sin_family = AF_INET;
    cc.sin_addr.s_addr = htonl(INADDR_ANY);
    cc.sin_port = htons(atoi(argv[1]));
    des1=socket(AF_INET,SOCK_STREAM,0);
```

//el primer argumento de la estructura es por conexión IPv4  
//asigna una IP de la máquina INADDR\_ANY convierte IP a protocolo de computadora  
//define puerto al servidor argv[1]

```
    bind(des1,(struct sockaddr *)&cc,sizeof(struct sockaddr_in));
```

// vincula el socket a la dirección del socket

```
    listen(des1,1);
```



//el des1 escucha al descriptor del socket y el 1 es la cantidad de pendientes

```
    while(1){
        des2=accept(des1, NULL, 0);
```

//se bloquea en espera, NULL(no interesa guardar la dirección del cliente)

```
        while((aa=read(des2, bb, sizeof(bb)))>0) {
```

//lee del socket, escribe en el descriptor del socket el contenido del buffer bb que es donde lo guardo y su tamaño bb

```
            write(des2,"ok",2);
```

//escribe en el socket

```
        }
    }
    return 0;
```

// escribe lo leído del socket por pantalla

```
}
```

```
return 0;
```

```
}
```

## APLICACIÓN SOCKET SERVIDOR STREAM: "OK"- CHAR FF [256]; INT AA, SS1, SS2; antes de llenar add.sin\_port

### Aplicación socket servidor stream

- Recibe datos por el socket, le envía al socket cliente "ok" y muestra en pantalla lo recibido por el socket.
- Puede conectarse con varios clientes en simultaneo utilizando hijos.

Completar:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>

char ff[256];
struct sockaddr_in add={};
int aa, ss1, ss2;

int main(int argc , char const * argv[]){
    argv[1]="2000";
    add.sin_family = AF_INET;
    add.sin_addr.s_addr = htonl(INADDR_ANY);
    add.sin_port = htons(atoi(argv[1]));

    ss1=socket(AF_INET, SOCK_STREAM, 0);
    bind(ss1, (struct sockaddr *)&add, sizeof(struct sockaddr_in));
    listen(ss1, 1);

    while(1) {
        ss2=accept(ss1, NULL, 0);
        if(fork()) {
            //linea en blanco
            while((aa = read(ss2, ff, sizeof(ff))) > 0) {
                write(ss2, "ok", 2);
                write(STDOUT_FILENO, ff, aa);
            }
        }
    }
    return 0;
}
```

// declara vector de 256 posiciones  
//define estructura  
// define variables globales enteras  
//declara variables locales  
//ingresa puerto en el vector argv[1]=2000  
//el primer argumento de la estructura es por conexion IPv4  
// asigna IP de la maquina, INADDR convierte IP a protocolo PC  
//define el puerto del servidor en argv[1]  
//el descriptor que devuelve socket lo guarda en ss1  
//vincula el socket a la direccion del socket  
//ss1 escucha el descriptor del socket y 1 cantidad de pendientes  
//se bloquea en la espera, NULL (no me interesa guardar la descripcion del cliente)  
//crea proceso hijo  
//lee el socket ss2  
//escribe lo leído en el socket  
//escribe por pantalla lo leído en el socket

## APLICACIÓN SOCKET SERVIDOR STREAM: - CHAR BB [256]; INT AA, dd1, dd2, CC; antes de llenar argv [1]

Aplicación [socket](#) servidor stream

- Los datos recibidos por el [socket](#) los reenvía al [socket](#) cliente y los muestra en pantalla.
- Puede conectarse con varios clientes en simultaneo utilizando hijos.

Completar:

```
include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>

char bb[256]; //declara vector de 256 posiciones
struct sockaddr_in st_dir={}; //define estructura
int aa,dd1,dd2,cc; //define variables globales enteras

int main(int argc , char const * argv[]){ //declara variables locales
    argv[1]="2000"; //ingresa puerto en el vector argv[1]=2000
    st_dir.sin_family = AF_INET; //el primer argumento de la estructura es por conexion IPv4
    st_dir.sin_addr.s_addr = htonl(INADDR_ANY); //asigna IP de la maquina, INADDR_ANY convierte IP a protocolo de PC
    st_dir.sin_port = htons(atoi(argv[1])); //define puerto del servidor argv[1]
    dd1=socket(AF_INET, SOCK_STREAM, 0); // el descriptor que devuelve socket lo guarda en dd1
    bind(dd1, (struct sockaddr *)&st_dir,sizeof(struct sockaddr_in)); // vincula el socket a la direccion del socket
    listen(dd1,1); // el dd1 escucha el descriptor del socket 1 cantidad de pendiente
    //línea en blanco

    while(1) {
        dd2=accept(dd1, NULL, 0); //se bloquea en la espera NULL (no me interesa guardar la direccion del cliente) dd2 lo uso para escribir y leer
        cc=fork(); //crea HIJO
        if(cc==0) { //si se crea devuelve un 0
            while((aa = read(dd2,bb,sizeof(bb))) > 0) { //lee el socket dd2
                write(dd2,bb,aa); //escribe lo leído del socket
                write(STDOUT_FILENO, bb, aa); //escribe por pantalla lo leído del socket
            }
        }
    }
    return 0;
}
```

## APLICACIÓN SOCKET CLIENTE STREAM: "IP: 92.168.10.10, PUETO: 2000" - CHAR AA [256], BB [256] antes de llenar cc.sin port

Aplicación [socket](#) cliente stream

- La aplicación cliente se conecta con un servidor (IP: 192.168.10.10, Puerto: 2000)
- Al recibir algo por teclado, envía por el [socket](#) "data\_tx", lee la respuesta el [socket](#) y la muestra en pantalla.

Completar:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>

char aa[256], bb[256]; // declara vector de 256 posiciones
struct sockaddr_in cc={}; //define estructura
int ss,ee; //define variables globales enteras

int main(int argc , char const * argv[]){ //declara variables locales
    argv[1]="2000"; // ingresa puerto en el vector argv[1]=2000
    cc.sin_family= AF_INET; // el primer argumento de la estructura es por conexion IPv4
    cc.sin_port=htons(atoi(argv[1])); // define puerto del servidor argv1
    inet_aton("192.168.10.10", &cc.sin_addr); //convierte la direccion de IP estandar a binario
    ss=socket(AF_INET, SOCK_STREAM, 0); //crea el socket, descriptor del socket en ss
    connect(ss,(struct sockaddr *)&cc,sizeof(cc)); //realiza un connect (conecta socket activo)
    while(1){
        //linea en blanco
        if ((read(STDIN_FILENO,aa,sizeof(aa))) > 0) { //lee de teclado
            write(ss,"data_tx",7); // escribe en el socket
        }
        ee = read(ss,bb,sizeof(bb)); // lee las variables las guarda en ee
        write(STDOUT_FILENO,bb,ee); // imprime por pantalla ee ; bb
    }
    return 0;
}
```

## APLICACIÓN SOCKET CLIENTE STREAM: "IP: 192.168.0.1, PUETO: 2000" - CHAR AA [256], BB [256] antes de llenar cc.sin\_port

Aplicación [socket](#) cliente stream

- La aplicación cliente se conecta con un servidor (IP: 192.168.0.1, Puerto: 2000)
- Al recibir algo por teclado, envía por el [socket](#) "dato tx", lee la respuesta el [socket](#) y la muestra en pantalla.

Completar:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>

char aa[256], bb[256];           // declara 2 vectores char
struct sockaddr_in cc={};       //define estructura
int ss,ee;                     // declara 2 variables globales enteras

int main(int argc , char const * argv[]){ // declara variables locales
    argv[1]="2000";             // ingresa puerto en argv[1]=2000
    cc.sin_family= AF_INET;     // eñ primer argumento de la estructura es por conexion IPv4
    cc.sin_port=htons(atoi(argv[1])); //puerto previamente guardado en argv

    inet_aton("192.168.0.1", &cc.sin_addr); //convierte direccion IP a cadena de caracteres

    ss=socket(AF_INET, SOCK_STREAM, 0); // crea el socket

    connect(ss,(struct sockaddr *)&cc,sizeof(cc)); //realiza un connect, conecta un socket activo

    while(1){
        //línea en blanco ✓

        if ((read(STDIN_FILENO,aa,sizeof(aa))) > 0) { // lee de teclado
            write(ss,"dato tx",7); // escribe en el socket ss ✓
        }

        ee = read(ss,bb,sizeof(bb)); // guarda en ee las variables ss bb
        write(STDOUT_FILENO,bb,ee); // imprime por pantalla
    }

    return 0;
}
```



El siguiente programa es un **SERVIDOR** basado en **SOCKET STREAM** que reenvía por el socket lo que recibió por el socket (eco). Indique al menos 2 errores en el siguiente código: (falta marcar un error mas)

10) El siguiente programa es un servidor basado en socket stream que reenvía por el socket lo que recibe por el socket (eco). Indique al menos 2 errores en el siguiente código.

```
int d, a, b, c;
char fg[256], h[1];
struct sockaddr_in e = {};

int main(int argc, const char *argv[])
{
    argv[1]="2000";

    b = socket(AF_INET, SOCK_STREAM, 0);
    if((b)<0) { exit(-1); }

    e.sin_family = AF_INET;

    a = bind (b, (struct sockaddr *)&e,sizeof(struct sockaddr_in)) ;
    if (a<0) { exit(-1); }

    a = listen(a, 1);
    if (a<0) { exit(-1); }

    while(1)
    {
        c = accept(b, NULL, 0) ;

        if (c >=0) {send (c ,"\n", sizeof("\n"),0 ); }
        if (c >=0)
        {
            while ( ( d = read(c, fg, sizeof (fg))) > 0 )
            {
                write ( c ,fg, d);
            }
        }

        close(c);

        return 0;
    }
}
```

ESTA MAL QUE  
SEAN LAS  
MISMAS  
UNIDADES

```
char bb[256];
struct sockaddr_in struct_direccion={};
int aa,desock,desacc;

int main(int argc , char const * argv[]){

    desock=socket(AF_INET, SOCK_STREAM, 0);
    argv[1]="2000";
    struct_direccion.sin_addr.s_addr = htonl(INADDR_ANY);
    struct_direccion.sin_port = htons(atoi (argv[1]));
    listen(desock,1);
    bind(desock,(struct sockaddr *)&struct_direccion,sizeof(struct
sockaddr_in));
    while(1) {
        desacc=accept(desock, NULL, 0) ;
        while (( aa = read(desacc, bb, sizeof (bb))) > 0) {
            write (desacc ,bb, aa);
            write (STDOUT_FILENO , bb, aa);
        }
    }
    return 0;
}
```

Se Confundió:  
• familia  
• IP  
• Puerto

\* Falta Indicar el Campo de la familia en la estructura.

\* Se llama después de la llamada a bind().

\* Falta cerrar el socket "close()". (LIBERAR conexión)

## PUNTO FIJO

El siguiente número binario representa a un número entero en complemento a dos: 1000 0011

El siguiente número binario representa a un número entero en complemento a dos:

1000 0011

¿Qué número representa en base 10?

Respuesta:  ✓

Punto fijo=8bits

Calculadora PC, en opción BYTE, escribo numero binario 1000 0011 y miro en decimal el numero -125

Two's complement signed integers:

- Range:  $-2^{N-1}$  to  $2^{N-1} - 1$ . un lugar o palabra para el cero

- $n_{10} = -b_{N-1}2^{N-1} + \sum_{i=0}^{N-2} b_i 2^i$

MA's usado por eficiencia de Hardware, porque al operar se puede usar un sumador tanto para sumar como restar

El siguiente número binario representa a un número entero en complemento a dos: 1111 1111

El siguiente número binario representa a un número entero en complemento a dos:

1111 1111

¿Qué número representa en base 10?

Respuesta:  ✓

Punto fijo= 8bits

Calculadora PC, opción BYTE, escribo numero binario 1111 1111 y miro en decimal -1

Two's complement signed integers:

- Range:  $-2^{N-1}$  to  $2^{N-1} - 1$ . un lugar o palabra para el cero

- $n_{10} = -b_{N-1}2^{N-1} + \sum_{i=0}^{N-2} b_i 2^i$

MA's usado por eficiencia de Hardware, porque al operar se puede usar un sumador tanto para sumar como restar



¿Cuántos bits tienen en total un número en Punto Fijo representado en formato Q3.16?

¿Cuántos bits tiene en total un número en punto fijo representado en formato Q3.16?

Respuesta:

20



**Va a tener 3bits parte entera + 16 parte decimal + 1 bit de signo = 20**

¿Cuántos bits tienen en total un número en Punto Fijo representado en formato Q15?

¿Cuántos bits tiene en total un número en punto fijo representado en formato Q15?

Respuesta:

16



**Q15 tiene 15 bits más 1 bit de signo = 16 en total**

**Q3.12 tiene 3 bits parte entera más 1 bit de signo más 12 bits decimal para dar un total de 16bits**

**Q0.31 tiene 31 bits decimales más 1 bit de signo = 32 en total**

se interpreta Q0.15  
1 para signo  
0 enteros  
15 decimales  
total=16 bits

**Si se multiplican 2 números en formato Q15, ¿cuántos bits se necesitan como mínimo para guardar el resultado correctamente?**

- a) 30
- b) 31 (2N - 1)**
- c) 32
- d) 40
- e) Ninguna de las anteriores

**El producto de dos números de Nbits, requiere 2Nbits para contener todos los valores posibles**  
**15x2=30 + 1bit de signo =31**

**Implemente la operación MAC en lenguaje C en forma eficiente para números Q3.12 (16 bits)**

```
int16_t a[K];  
int16_t b[K];  
int32_t c = 0;  
for (i=0; i<K; i++)  
{ c = c + ( (int32_t) a[i] * (int32_t) b[i] ); }
```

**Hacer el programa de multiplicar y acumular**

## Complete el siguiente código en C para que el resultado de la Operación MAC presente el menor error posible para la Precisión Q15.16

Complete el siguiente código en C para que el resultado de la operación MAC presente el menor error posible para la precisión Q15.16:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <limits.h>

int32_t fp2fx(float X, int32_t n){ punto flotante a punto fijo
    int32_t temp;
    temp=(int32_t) round( X * ( 1 < n ) ); ✓ multiplico
    return temp;
}

float fx2fp(int64_t X, int32_t n){ punto fijo a punto flotante le paso el numero y la presicion
    float temp;
    temp = (float) (X) / ( 1 < n ); ✓ divido
    return temp;
}

int main(int argc, char** argv)
{
    float x[5]={1.1,2.2,3.3,4.4,5.5};
    float y[5]={6.6,7.7,8.8,9.9,10.10};

    int32_t N = 5;
    int64_t acum = 0; ✓
    int64_t aux = 0; ✓

    double acum_db;
    double aux_db;

    acum = 0;

    for(int i=0;i<N;i++)
    {
        aux = (int64_t) fp2fx( x[i], 16 ) * (int64_t) fp2fx( y[i], 16); ✓
        acum = (acum + aux);
    }

    acum_db = 0.0;

    for(int i=0;i<N;i++)
    {
        aux_db = (double)x[i] * (double)y[i];
        acum_db = acum_db + aux_db;
    }

    printf("acum = %ld \n", acum );
    printf("acum = %f \n", (float)(acum >> 32) );
    printf("acum_db = %f \n", acum_db );
}
```

**Cuál es el mejor enfoque de REDONDEO cuando se desea implementar la operación MAC entre dos conjuntos grandes de números en PUNTO FIJO.**

### Operación MAC

Basicamente se trata de Multiplicar y Acumular el resultado. La operación MAC representa la suma de convolución en el dominio discreto y resulta útil en DSP.

En "C", nos queda

$$\text{res} = 0$$

for ( $i=0$ ;  $i<n$ ;  $i++$ )

$$\text{res} = \text{res} + (a \times b)$$

¿Pero conviene calcular en cada iteración o al finalizar el bucle?

### Mejor enfoque de Redondeo

Conviene utilizar el método de "Redondeo al valor más cercano" ya que la media aritmética ( $\mu$ ) del error tiende a cero ( $\mu=0$ ).

El redondeo consiste en:

- Tomar el valor a redondear
- Sumarle o restarle a mitad de la precisión
- Truncar

Describe el RANGO, la PRECISION, y la Cantidad de Bits necesarios para el formato Q10.16

c) Describe el rango, la precisión y la cantidad de bits necesarios para el formato Q10.16

### Representación en Punto Fijo Qm.n

$$N = m + n + 1 \rightarrow \text{longitud de la palabra en representación Qm.n}$$

$$N = 10 + 6 + 1$$

donde:

$m$  = cantidad de bits de la Parte Entera

$n$  = cantidad de bits de la Parte Fraccionaria

$1$  = Representa el bit reservado para el signo.

•  $N = 17$  ✓

$$\text{Rango} : -2^{N-1} \text{ a } 2^{N-1} - 1$$

$$\text{Rango} = -2^{17-1} \text{ a } 2^{17-1} - 1 \quad -2^{10} \text{ a } 2^{10} - 2^{-6}$$

•  $\text{Rango} = -65536 \text{ a } +65535$  ✗

$$\text{Precisión} = 2^{-n}$$

$$\text{Precisión} = 2^{-6} =$$

•  $\text{Precisión} = \frac{1}{64} = 0,015625$  ✓

Si la variable ab se representa en Formato Qm.n ¿Qué valor tiene ab en Base 10?

La siguiente expresión es una operación en C:

`ab = 1 << n;`

Si la variable ab se representa en formato Qm,n, ¿qué valor tiene ab en base 10?

- ☐ a. 0.5
- ☐ b. 127

☒ c. 1

1 → luego del 1 agrego n cantidad de ceros → 1,0000 n ceros. Es igual a parte entera 1, parte decimal cero.

si desplazo un 1 hacia la izquierda la misma cantidad de veces que mis numeros decimales  
no queda ningun decimal  
multiplico n veces por 2

$X := (\text{int})(x \cdot (1 \ll n))$  multiplico por la unidad en  
 $X := (\text{int})(x \cdot 2^n)$  en formato P. Fijo

Si la variable ab se representa en Formato Qm,n ¿Qué valor tiene ab en Base 10?

La siguiente expresión es una operación en C:

`ab = 1 << (n-1);`

Si la variable ab se representa en formato Qm,n, ¿qué valor tiene ab en base 10?

☒ a. 0.5

☐ b. 1

☐ c. 127

0,1000...<sup>n</sup>

Parte entera 0, parte decimal  $1 \times 2^{-1} = 0,5$

- Unit:  $z = 1 \ll n = 1 \cdot 2^n$ . agregar n bits a su derecha, desplazamiento a la izquierda es lo mismo que multiplicar por  $2^n$

Example:  $n = 4 \Rightarrow z = 1.0000_2$ .

- One half ( $1/2$ ):  $z = 1 \ll (n-1) = 1 \cdot 2^{(n-1)}$ .

Example:  $n = 4 \Rightarrow z = 0.1000_2$ .

$2^{-1} = 0,5$

Calcule el valor de entero (int) que representa el número 13.4 en Formato Q3.4

Calcule el valor de entero (int) que representa el número 13.4 en formato Q3.4.

Respuesta:  ✓

$$X * 2^n; \text{ siendo } Qm.n$$

$$13.4 * 2^4 = 214.4 \rightarrow \text{parte entera} = 214$$

Calcule el valor de entero (int) que representa el número 13.4 en Formato Q7.8

Calcule el valor de entero (int) que representa el número 13.4 en formato Q7.8.

Respuesta:  ✓

$$x * 2^n; \text{ siendo } Qm.n$$

$$13.4 * 2^8 = 3430.4 \rightarrow \text{parte entera} = 3430$$

Analice el siguiente código C. Si a y b son variables que están representadas en Formato Entero (int) ¿Qué problema se ha producido?

Analice el siguiente código en C:

```
int8_t a, b, c;  
a = 127;  
b = 2;  
c = a + b;
```

son 8 bits, 1 uno para el signo, quedan 7 para el numero entero

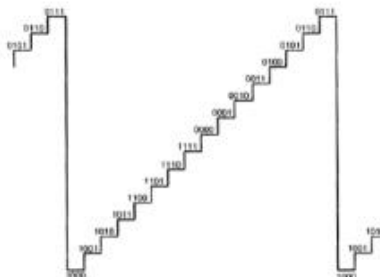
- Range:  $-2^{N-1}$  to  $2^{N-1} - 1$ . un lugar o palabra para el cero
- $n_{10} = -b_{N-1}2^{N-1} + \sum_{i=0}^{N-2} b_i 2^i$

Si a y b son variables que están representadas en formato entero (int), ¿qué problema se ha producido?

- ☒ a. Overflow.
- ☐ b. Underflow.
- ☐ c. Ninguno.

Int8 es 8 bits, entonces va de -128 a 127.  $127+2=129$  entonces hay se pasa la parte entera = overflow

- An **overflow** occurs in a when a result is greater than  $2^{N-1} - 1$  or lesser than  $-2^{N-1}$ . Se produce cuando el resultado sale del rango con el que estoy operando





Analice el siguiente código C. Si a y b son variables que están representadas en Formato Punto Fijo Q3.4 ¿Qué problema se ha producido?

Analice el siguiente código en C:

```
int8_t a, b;  
int16_t c;  
a = 127;  
b = 2;  
c = a + b;
```

Si a y b son variables que están representadas en formato punto fijo Q3.4, ¿qué problema se ha producido?

- ☐ a. Underflow.
- ☒ b. Ninguno.
- ☐ c. Overflow.

a y b 8 bits (-128 a 127) en formato Q3.4,  $127 \times 2^4 = 2032$ ,  $2 \times 2^4 = 32$ ,  $2032 + 32 = 2064$  en binario entra en 16bits (calculadora pc)

How to avoid overflow

1ra forma de evitar el overflows (la mas sencilla) acumulador con mayor cantidad de Bits .... Se tiene que calcular la cantidad de bits necesarios o la cantidad maxima de palabras que puede sumar el acumulador que tengo disponible

Longer word-length accumulator

- Saving the result in a N+1 word avoids overflows.
- The general rule is the sum of s individual m-bit can require an accumulator of as many as  $m + \log_2(s)$  bits. Para calcular cuantos bits hacen falta para sumar "s" numeros de m bits
- **Example:** 256 8-bits words requires an accumulator whose word length is  $8 + \log_2(256) = 16$ .

Analice el siguiente código C. Si a y b son variables que están representadas en Formato Punto Fijo Q3.4 ¿Qué problema se ha producido?

Analice el siguiente código en C:

```
int8_t a, b;  
int16_t c;  
a = 127;  
b = 2;  
c = a * b;
```

Si a y b son variables que están representadas en formato punto fijo Q3.4, ¿qué problema se ha producido?

- ☒ a. Underflow.
- ☐ b. Overflow.
- ☐ c. Ninguno.

a y b 8 bits (-128 a 127) en formato Q3.4.  $127 \times 2^4 = 2032$ ,  $2 \times 2^4 = 32$   $2032 \times 32 = 65024$ , se pasa de los 16 bits, como es multiplicación es un underflow

si los valores de a y b la compu los interpreta como enteros, es el programador quien interpreta el formato Q3.4 .... por eso no hay overflow

Pero si Underflow porque no puedo representar  $2^8$  los numeros que puedo representar santan de  $2^4$  asumiendo que c tienen el mismo formato que a y b



¿Qué cambios introduciría en el siguiente programa para mejorar la precisión del resultado (variable C)? elija todas las opciones que considere correcta – void main (void)

¿Qué cambios introduciría en el siguiente programa para mejorar la precisión del resultado (variable c)? Elija todas las opciones que considere correctas.

```
1 void main(void)
2 {
3     int16_t i;
4     int32_t a, b;
5     int32_t c;
6
7     c = 0;
8
9     for (i = 0; i < 256; i++)
10    {
11        c = c + a * b;
12    }
13 }
14
```

Seleccione una o más de una:

- ☐ a. En la línea 11, se debe hacer un casting de la variable c a int32\_t.
- ☒ b. En la línea 11, se debe hacer un casting de las variables a y b a int64\_t.
- ☒ c. En la línea 5, la variable c debe declararse como int64\_t.
- ☐ d. Se debe borrar la línea 7.
- ☐ e. No hacer nada.

Activar V

Al aumentar tanto a, b y c de 32bit a 64bit voy a tener más precisión en la parte decimal

¿Qué cambios introduciría en el siguiente programa para mejorar la PRECISION del resultado (VARIABLE c)? elija todas las opciones que considere correcta

¿Qué cambios introduciría en el siguiente programa para mejorar la precisión del resultado (variable c)? Elija todas las opciones que considere correctas.

```
1 void main(void)
2 {
3     int16_t i;
4     int32_t a, b;
5     float c;
6
7     c = 0;
8
9     for (i = 0; i < 256; i++)
10    {
11        c = c + a * b;
12    }
13 }
14
```

Seleccione una o más de una:

- ☐ a. En la línea 11, se debe hacer un casting de la variable c a int64\_t.
- ☒ b. En la línea 11, se debe hacer un casting de las variables a y b a float.
- ☐ c. En la línea 11, se debe hacer un casting de las variables a y b a int64\_t.
- ☐ d. No hacer nada.

Corrija los errores que presenta el siguiente código en C, si los hubiera:

Código original	Código corregido
<pre>int16 i; int32 a, b; float c;  c = 0.0; for(i == 0; i &lt; 256; i++ ) {     c += a * b; }</pre>	<pre>int16 i; int32 a, b; float c; a = 1; b = 1; c = 0.0; for(i = 0; i &lt; 256; i++ ) {     c += a * b; }</pre>

En el for i=0 no == que es una asignación

Indique todas las observaciones que considere correctas sobre el siguiente programa- define N 16

Indique todas las observaciones que considere correctas sobre el siguiente programa.

```

1
2 #define N 16
3 int32_t fp2fx (float x, int16_t N);
4
5 main (void)
6 {
7
8     int16_t i;
9     int32_t c;
10    float a[5] = {1.1, 2.2, 3.3, 4.5, 7.6};
11    float b[5] = {1000.1, 2000.2, 3000.3, 4000.5, 7000.6};
12
13    c = 0;
14
15    for (i=0; i<256; i++)
16    {
17        c = c + fp2fx(a[i]) * fp2fx (b[i]);
18    }
19 }
20

```

Seleccione una o más de una:

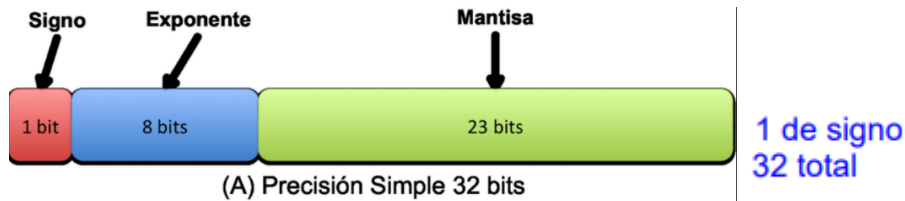
- ☒ a. El bucle for se ejecuta demasiadas veces.
- ☐ b. El programa implementa la operación MAC en punto flotante.
- ☒ c. La función fp2fx() está declarada incorrectamente.
- ☐ d. La variable c debería ser punto flotante (float).
- ☒ e. La variable c debería ser un entero de 64 bits (int64\_t).
- ☒ f. El programa implementa la operación MAC en punto fijo.

## PTO FLOTANTE

### En el Formato PUNTO FLOTANTE PRECISION 32 bits (single)

En el formato punto flotante precisión 32 bits (*single*),

¿Cuántos bits tiene el campo F?	23	✓
¿Cuántos bits tiene el campo E?	8	✓



### En el formato PUNTO FLOTANTE precision 64BITS (double)

En el formato punto flotante precisión 64 bits (*double*),

¿Cuántos bits tiene el campo E?	11	✓
¿Cuántos bits tiene el campo F?	52	✓



### ¿Cuánto vale aproximadamente el Rango Dinámico del formato Punto Flotante de 64 BITS en db?

¿Cuánto vale aproximadamente el rango dinámico del formato punto flotante de 64 bits en dB?

- ☐ a. 1541
- ☐ b. 6154
- ☐ c. 24657
- ☒ d. 12328

$$6,02 \times 2^{11} = 12328$$

en este formato el Exp = 11

Dynamic range for floating-point numbers is defined as,

$$DR_{dB} \approx 6.02 \cdot 2^{b_E}$$

## DSP

Si una señal Senoidal de 5450HZ se muestrea con una frecuencia de muestreo de 4900Hz ¿Qué señal observaría a la salida del conversor ADC?

Si una señal senoidal de 5450 Hz se muestrea con una frecuencia de muestreo de de 4900 Hz, ¿qué señal observaría a la salida del conversor ADC?

No utilice un punto para separar unidades de mil.

Respuesta: 550

Una señal alias de 550 Hz

Si una señal Senoidal de 5450HZ se muestrea con una frecuencia de muestreo de 11000Hz ¿Qué señal observaría a la salida del conversor ADC?

Si una señal senoidal de 5450 Hz se muestrea con una frecuencia de muestreo de de 11000 Hz, ¿qué señal observaría a la salida del conversor ADC?

No utilice un punto para separar unidades de mil.

Respuesta: 5450

$$\Omega_s - \Omega_N \geq \Omega_N, \text{ or,}$$

$$\Omega_s \geq 2\Omega_N$$

Suponga que debe muestrear un Sensor Analógico cuya maxima frecuencia es de 200 HZ. Como Filtro ALIASSING solo puede usar un Filtro Pasa Bajo RC de tercer Orden con frecuencia de Corte de 400 Hz ¿Qué Mínima Frecuencia de Muestreo fijaría para evitar el efecto de ALIASING?

Suponga que debe muestrear un sensor analógico cuya máxima frecuencia es de 200 Hz. Como filtro antialiasing solo puede usar un filtro pasa-bajos RC de tercer orden con frecuencia de corte de 400 Hz. ¿Qué **mínima** frecuencia de muestreo fijaría para evitar el efecto de aliasing?

Respuesta: 800

Suponga que debe muestrear un Sensor Analógico cuya maxima frecuencia es de 200 HZ. Como Filtro ALIASSING solo puede usar un Filtro Pasa Bajo RC de tercer Orden con frecuencia de Corte de 600 Hz ¿Qué Mínima Frecuencia de Muestreo fijaría para evitar el efecto de ALIASING?

Suponga que debe muestrear un sensor analógico cuya máxima frecuencia es de 200 Hz. Como filtro antialiasing solo puede usar un filtro pasa-bajos RC de tercer orden con frecuencia de corte de 600 Hz. ¿Qué **mínima** frecuencia de muestreo fijaría para evitar el efecto de aliasing?

Respuesta: 1200

Suponga que debe digitalizar la salida Analógica de un GIROSCOPIO cuya relación señal a ruido según el fabricante es de 58 db. Determine cuantos bits  $B$  debe tener el ADC para un correcto muestreo de la señal. Redondee el número de bits al Entero Superior (redondeo hacia más Infinito). Considere que  $SNR_{ADC} = 6 \cdot B [dB]$

Suponga que debe digitalizar la salida analógica de un giróscopo cuya relación señal ruido según el fabricante es de 58 dB. Determine cuántos bits  $B$  debe tener el ADC para un correcto muestreo de la señal. Redondee el número de bits al entero superior (Rendondeo hacia más infinito).

Considere que:

$$SNR_{ADC} = 6 \cdot B [dB].$$

Respuesta: 10



$6.02 \cdot B + 1.76 = 58 \text{db} + 3 \text{db} = 9.50 = 10$  lo saco usando la calculadora

Suponga que debe digitalizar la salida Analógica de un ACELEROMETRO cuya relación señal a ruido según el fabricante es de 91 db. Determine cuantos bits  $B$  debe tener el ADC para un correcto muestreo de la señal. Redondee el número de bits al Entero Superior (redondeo hacia más Infinito). Considere que  $SNR_{ADC} = 6 \cdot B [dB]$

Suponga que debe digitalizar la salida analógica de un acelerómetro cuya relación señal ruido según el fabricante es de 91 dB. Determine cuántos bits  $B$  debe tener el ADC para un correcto muestreo de la señal. Redondee el número de bits al entero superior (Rendondeo hacia más infinito).

Considere que:

$$SNR_{ADC} = 6 \cdot B [dB].$$

Respuesta: 16



$6.02 \cdot B + 1.76 = 91 \text{db} + 3 \text{db} = 15.32 = 16$  lo saco usando la calculadora

Suponga que debe digitalizar la salida Analógica de un ACELEROMETRO cuya relación señal a ruido según el fabricante es de 56 db. Determine cuantos bits debería tener el ADC para un correcto muestreo. Justifique

Respuesta: 10



$6.02 \cdot B + 1.76 = 56 \text{db} + 3 \text{db} = 9.50 = 10$  lo saco usando la calculadora

Si una señal analógica tiene una SNR de 42db y es muestreada con un ADC de 10bits ¿aproximadamente cuantos bits se usaran para muestrear ruido?

Si una señal analógica tiene una SNR de 42 dB y es muestreada con un ADC de 10 bits, ¿aproximadamente cuántos bits se usarán para muestrear ruido?

Respuesta: 3



$6.02 \cdot B + 1.76 = 42 \text{ dB}$   $\Rightarrow 6.68 = 7$  lo saco usando la calculadora, no pide correcto muestreo asi que 7 a 10 son 3 adicionales.

$$\begin{aligned} \text{Respuesta: } \text{SNR}_{\text{ADC}} &= 1.76 + 6.02 \cdot B = 61.96 \text{ dB} \\ 61.96 \text{ dB} - 3 \text{ dB} &= 58.96 \text{ dB} \\ (58.96 - 42) / 6 &= 2.83 \sim 3 \text{ bits} \end{aligned}$$

$$\begin{aligned} &= 1.76 + 6.02 \cdot B \quad [\text{dB}] \\ &= 61.96 \text{ dB SNR del ADC} \\ 42 - 62 - 3 \text{ (margen)} &= -17 \text{ dB} \\ \text{un bit se deja de margen de} & \\ \text{seguridad se recortan los 2 exedentes} & \end{aligned}$$

$$-17 \text{ dB} / 6.02 = 2.8 \dots \text{ aprox } 3 \text{ muestrean ruido}$$

Suponga que debe muestrear la salida de una termocupla que presenta una SNR = 80 dB. Indique la cantidad de bits que necesita un ADC para no deteriorar esta señal y qué filtro discreto usaría para reducir el ruido a la salida del ADC.

$$\text{SNR}_{\text{ADC}} = 1.76 + 6.02 \cdot B - 3 \geq 80 \implies B \geq 13.5 \Rightarrow B = 16 \text{ bits}$$

Como se desea eliminar ruido se debe usar un filtro en el dominio del tiempo (MA o LI).

$$6.02 \cdot B + 1.76 = 80 \text{ dB} + 3 \text{ dB} = 83 \text{ dB} \Rightarrow 13.49 = 14 \text{ lo saco usando la calculadora}$$

Suponga que un sensor cardíaco presenta una relación señal a ruido de 88 dB según su fabricante. Suponga que solo tiene a disposición conversores A/D de 12, 16 y 24 bits. Cuál usaría? ¿Debe realizar alguna corrección al valor muestreado y digitalizado?

$$\begin{aligned} \text{SNR}_{\text{ADC}} &= 1.76 + 6.02 \cdot B - 3 \geq 88 \text{ dB} \\ B &\geq (88 - 1.76 + 3) / 6.02 \\ B &\geq 14.8 \Rightarrow 16 \end{aligned}$$

Al utilizar 16 bits para digital una señal que se debe digital con 15, estamos usando un bit para muestrear ruido, por lo que al valor muestreado se le debiera descartar el bit de ruido.

Suponga que tiene un Conversor Analógico/Digital ADC de 12 bit ¿Cuál es la maxima relación señal a ruido que debe presentar la señal analógica de entrada de este ADC sin que el ADC deteriore la calidad de la misma?

$$\text{ADC DE 12 BIT} \rightarrow \text{Si } \text{SNR} = 1.75 + 6.02B = 1.75 + 6.02 \cdot 12 - 3 = 70.99 \text{ dB}$$

## FIR

Si el ORDEN de un FILTRO MOVING AVERAGE es igual a  $N=10$ , ¿Cuánto vale cada elemento de su KERNEL  $h(n)$ ?

Si el orden de un filtro moving average es igual a  $N = 10$ , ¿cuanto vale cada elemento de su kernel  $h[n]$ ?

Respuesta:

$$\frac{1}{N}$$

Si se parte el Diseño de un Filtro Moving Average de Orden  $M=10$  ¿Qué valor tendrá el  $\lambda$  de un filtro Leaky Integrador?



Si se parte del diseño de un filtro moving average de orden  $M = 10$ , ¿qué valor tendrá el  $\lambda$  de un filtro leaky integrator?  
Exprese su respuesta hasta con 2 decimales.

Respuesta:

0.9



Defining  $\lambda = \frac{M-1}{M}$ ,

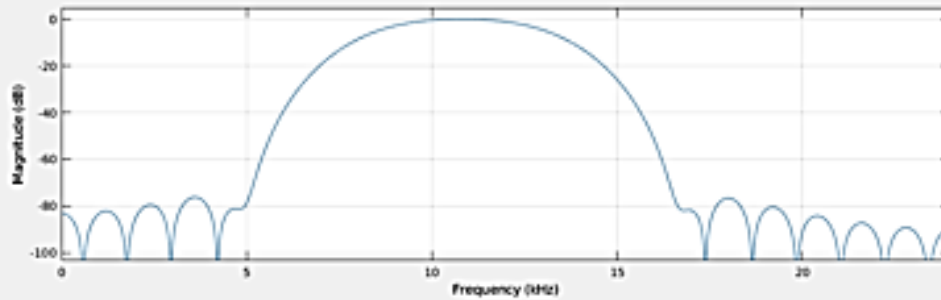
**Diseñe el siguiente FILTRO FIR (BLACKMAN- ORDEN 20- FC: 8400 Y 13200- FS48000) ¿Cuál de las siguientes figuras muestra la respuesta en frecuencia del filtro?**

Diseñe el siguiente filtro FIR:

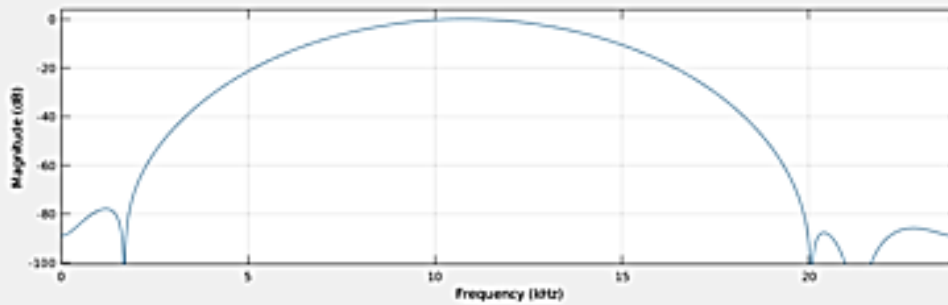
- 1) Filtro pasa banda con ventana de Blackman.
- 2) Orden del filtro: 20.
- 3) Frecuencias de corte: 8400 y 13200 Hz.
- 4) Frecuencia de muestreo: 48000 Hz.

¿Cuál de las siguientes figuras muestra la respuesta en frecuencia del filtro? El número de la figura es la respuesta a este ejercicio.

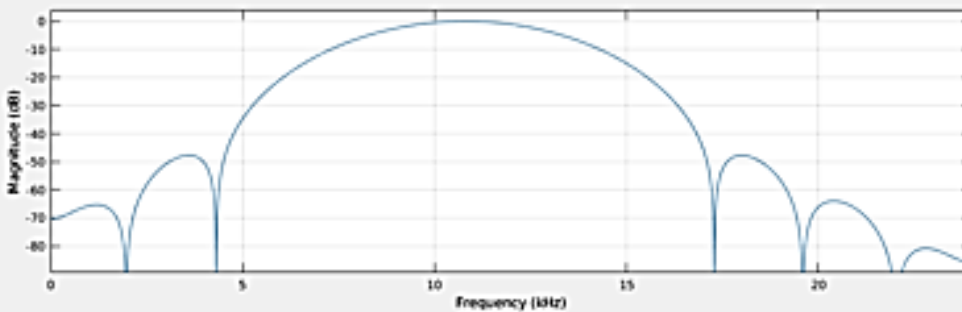
1)



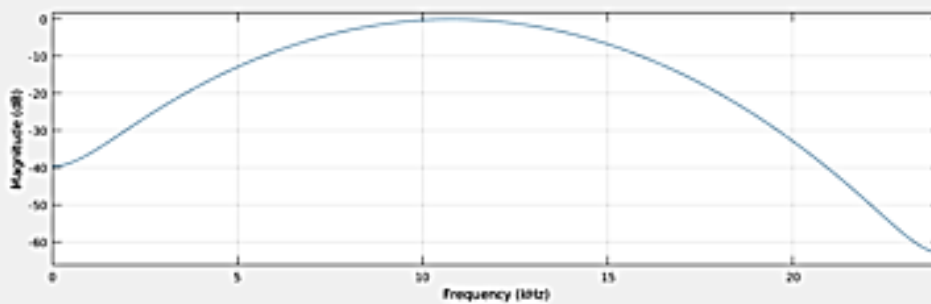
2)



3)



4)



Respuesta:  ✓

Obtenga los Coeficientes (vector coeff) que definen al siguiente Filtro FIR (Pasa Bajo Blackman- Orden: 50- Fc: 2000- Fs: 44100)

Obtenga los coeficientes (vector coeff) que definen al siguiente filtro FIR:

- 1) Filtro pasa bajos con ventana de Blackman.
- 2) Orden del filtro: 50.
- 3) Frecuencia de corte: 2000 Hz.
- 4) Frecuencia de muestreo: 44100 Hz.

Sume todos los elementos del vector coeff ( resp = sum(coeff) ). El valor de la suma (resp) es la respuesta a esta pregunta.  
Use hasta 2 decimales para expresar el resultado.

Respuesta:



## IR

**Obtenga los coeficientes (matriz SOS y vector G) que define el siguiente Filtro IIR: Filtro Pasa Bajo: Butterworth- Orden: 10- Fc: 3400- Fs: 44100**

Obtenga los coeficientes (matriz SOS y vector G) que definen al siguiente filtro IIR:

- 1) Filtro pasa bajos tipo Butterworth.
- 2) Orden del filtro: 10.
- 3) Frecuencia de corte: 3400 Hz.
- 4) Frecuencia de muestreo: 44100 Hz.

Sume todos los elementos de la matriz SOS ( resp = sum(sum(SOS)) ). El valor de la suma (resp) es la respuesta a esta pregunta.  
Use hasta 4 decimales para expresar el resultado. Utilice una coma para los decimales (no un punto).

Respuesta:



**Obtenga los coeficientes (matriz SOS y vector G) que define el siguiente Filtro IIR: Filtro Pasa Bajo: Elliptic- Orden: 20- Fc: 9600- Fs: 48000**

Obtenga los coeficientes (matriz SOS y vector G) que definen al siguiente filtro IIR:

- 1) Filtro pasa bajos tipo Elíptico.
- 2) Orden del filtro: 20.
- 3) Frecuencia de corte: 9600 Hz.
- 4) Frecuencia de muestreo: 48000 Hz.

Sume todos los elementos de la matriz SOS ( resp = sum(sum(SOS)) ). El valor de la suma (resp) es la respuesta a esta pregunta. Use hasta 4 decimales para expresar el resultado. Utilice una coma para los decimales (no un punto).

Respuesta:  ✓

### Dado el siguiente Sistema (ROC)

Dado el siguiente sistema,

$$Y(z) = \frac{z(z - 1)}{(z + 1)(z + 1/3)}$$

Si el ROC está dado por $1/3 <  z  < 1$	, el sistema es anti-causal y marginalmente estable.	✓
Si el ROC está dado por $ z  > 1$	, el sistema es causal y marginalmente estable.	✗
Si el ROC está dado por $ z  < 1/3$	, el sistema es anti-causal e inestable.	✓