

ANÁLISIS ESPECTRAL POR CORRIMIENTO DE LA FRECUENCIA DE MUESTREO

Gustavo Caamaño León

Se describe un analizador de espectro de muy bajo desempeño. Pero que puede trabajar en tiempo real con poco costo computacional. Por medio del uso de un filtro pasabanda, cuya localización de la banda pasante depende de la frecuencia de muestreo.

Introducción

entro del campo de aplicaciones de los microcontroladores podemos encontrar soluciones sencillas de procesamiento digital de señales que pueden implementarse con microcontroladores y así tener un ahorro considerable de dinero y circuitos comparado con el uso de un DSP especializado¹.

La mayoría de las aplicaciones requieren que se trabaje en tiempo real, es decir, que el cálculo de un valor de salida $Y[n]$ se realice mientras aún permanezca en la entrada la cuantificación correspondiente a la muestra de entrada $X[n]$. En otras palabras, todo el procesamiento correspondiente a un valor n debe hacerse en un tiempo inferior al período de muestreo T_s . Esta concepción de tiempo real, desde la óptica del procesamiento digital de señales, se esquematiza en la Figura 1.

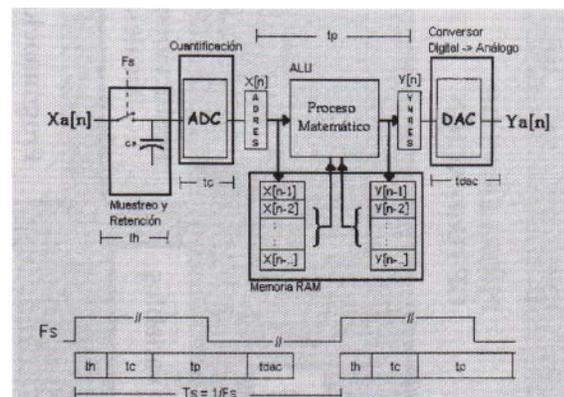


Figura 1. Diagrama en bloques de un sistema para DSP

Es posible cumplir con la condición anterior, usando microcontroladores de propósito general, cuando la frecuencia de muestreo es lo suficientemente lenta para compensar el costo computacional que implica realizar las operaciones matemáticas requeridas con una unidad lógico aritmética de bajo perfil.

* Docente de tiempo completo, Facultad Tecnológica, Universidad Distrital Francisco José de Caldas. Coordinador del proyecto curricular de Tecnología en Electrónica. 1 EMBEDDED CONTROL HANDBOOK, AN540, Implementing MR digital filters, DS00540B, Microchip, 1993

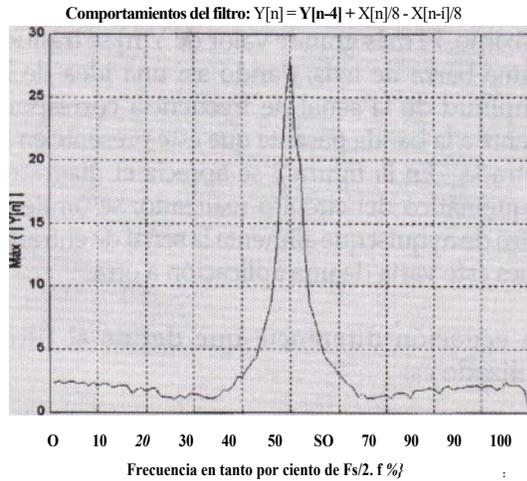


Figura 3. Comportamiento del filtro

En la figura 3 se muestra el comportamiento del filtro.

Como puede verse, la banda pasante está localizada en la cuarta parte de la frecuencia de muestreo ($F_s/4$). Como es normal en los filtros digitales, la frecuencia de banda de paso depende del valor que se asigne a la frecuencia de muestreo.

Por tanto, si se tiene una señal de entrada y se quiere obtener un estimado de la amplitud de una determinada frecuencia, f_1 , lo único que hay que hacer es disponer que:

con la condición de que la señal a analizar permanezca en la entrada el tiempo suficiente para que sean calculadas 50 muestras de la misma. Si se quiere analizar otra componente espectral, f_2 , entonces se cambia la frecuencia de muestreo para que sea:

En este artículo se hace el ejemplo explorando cuatro bandas frecuenciales distintas (Figura 4). Por supuesto, se debe tener cuidado de no

infringir el criterio de Nyquist, el cual limita la frecuencia más alta que pueda presentarse en la entrada a valores menores o iguales que 0.5 veces la frecuencia de muestreo, para no experimentar el fenómeno de solapamiento espectral o aliasing⁴. En la Tabla 1 se describen las cuatro frecuencias escogidas.

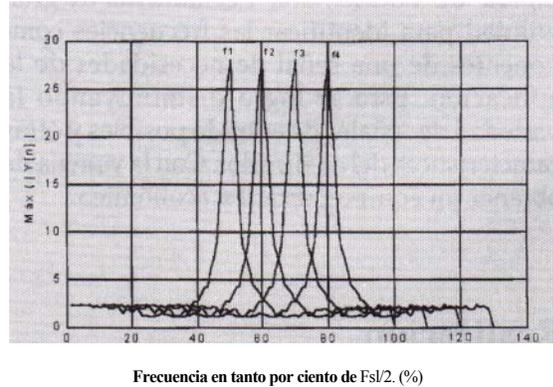


Figura 4. Comportamiento al correr la frecuencia de muestreo

En el Cuadro de Texto 1 se disponen las variables y constantes de uso en el programa, además de los archivos de soporte del mismo. Las constantes BANDA1, BANDA2, BANDAS y BANDA4; contienen el valor apropiado, en hexadecimal, para cada una de las frecuencias de muestreo seleccionadas.

Dichos valores corresponden a la fórmula:

$$CCP2R = T_s/T_l \text{ donde, } T_s = 1/F_{sij}$$

BANDA	FS [HZ]	FS/4 [HZ]
1	1600	400
2	1760	440
3	1936	484
4	2129.6	532.4

Tabla 1. Descripción de las frecuencias elegidas para el experimento

4 OPPENHEIM, Alan - SCHAFFER Ronald, DISCRETE TIME SIGNAL PROCESSING, Prentice Hall, 1989

```

HABILITACION DE INTERRUPCION
BSF INTCON,PEIE
BSF INTCON,GIE

MAIN:
BTFS MUESTRA_EN_BUFFER
GOTO MAIN

BCF MUESTRA_EN_BUFFER

; Y[N] = Y[N-4] + (1/8)*X[N] - (1/8)*X[N-6]

DIV_BIN XN,3
SUMAR_16 YN_4,1,XN,XN_6,YN_1,CUENTA
CORRER_WORD YN_1,XN_6,CUENTA
NEGATIVO_DE XN_6,(1/8)*X[N-6]

VALOR_ABS YN_1,1,ABSOLUTO,CUENTA
MAXIMO VALOR_PICO,ABSOLUTO

DECSZ CUENTA,MUESTRAS,1
;CUENTA CINCUENTA MUESTRAS
GOTO MAIN

BARRAS1 VALOR_PICO,PORTD,IMAGEN

BCF INTCON,GIE
;DESACTIVA INTERRUPCIONES

BCF T1CON,TMR1ON ;DETIENE EL TIMER 1
GOTO OTRA VEZ
;PREPARARSE PARA LA SIGUIENTE BANDA

Cuadro de Texto 4. Programa principal
    
```

```

SERVI:
BTFS PIR1,ADIF
GOTO FIN_CONVERSION

FIN_CONVERSION:
BCF PIR1,ADIF
;REPONE BANDERA DE PETICION DE INTERRUPCION POR EOC
MOVWF ADRES,0
MOVWF XN
;TOMA COMO MUESTRA ACTUAL EL RESULTADO DE LA CONVERSION

BCF XN,7
BTFS ADRES,7
BSF XN,7 ;CONVIERTE A FORMATO Q7

BSF MUESTRA_EN_BUFFER
RETFIE ;FIN DE LA Rutina DE SERVICIO

END

Cuadro de Texto 5. Rutina de Servicio
    
```

Cuadro de Texto 4. Programa principal
cuadro de Texto 5. Rutina de Servicio

voltios a +5 voltios y que, por tanto una señal alterna debe ser montada sobre 2.5 voltios D.C. para no infringir el rango del ADC.

El efecto producido por el cambio de formato se ejemplifica en la Tabla 2 utilizando el formato Q3. Nótese que simplemente es necesario invertir el bit de mayor peso de la muestra.

Desde el cuadro de texto 6 hasta el cuadro de texto 10 se enlistan las

NUM	COD ADC	VOLT IN. [VREF/15]	CÓDIGO Q3	VALOR EN COM2'S
0	0000	0	1000	-8
1	0001	1	1001	-7
2	0010	2	1010	-6
3	0011	3	1011	-5
4	0100	4	1100	-4
5	0101	5	1101	-3
6	0110	6	1110	-2
7	0111	7	1111	-1
8	1000	8	0000	0
9	1001	9	0001	1
10	1010	10	0010	2
11	1011	11	0011	3
12	1100	12	0100	4
13	1101	13	0101	5
14	1110	14	0110	6
15	1111	15	0111	7

Tabla 2. Formato Q3

```

LIST
;FUNCIONES SENCILLAS PARA HC
;EN APLICACIONES DE PROCESAMIENTO DIGITAL DE SEÑALES
NLIST

;DIV_BIN divide el número de 16 bits en COEF+1 por COEF entre un valor que
;es potencia de 2. Por medio de desplazamientos hacia la derecha. FOI2
;indica la cantidad de desplazamientos

DIV_BIN MACRO COEF, FOI2
LOCAL FOI2, a=0

WHILE a<FOI2
BCF STATUS,C
BIFSC COEF,7
BSF STATUS,C
RRF COEF,1
a+=1
ENDW

CLRFO COEF+1
BIFSC COEF,7
COMF COEF+1,1

ENDM

Cuadro de Texto 6. Macros. División entre potencias 2
    
```

Cuadro de Texto 6. Macros. División entre potencias 2

SUMAR_16 suma 3 números de 16 bits, |S1+1|S1|, |S2+1|S2|, |S3+1|S3|
 y guarda la suma en RESULTADO. BUFFER se usa como registro auxiliar

```
SUMAR_16 MACRO S1, S2, S3, RESULTADO, BUFFER
    MOVF S1,0
    ADDWF S2,0
    MOVWF RESULTADO
    CLRF BUFFER
    RLF BUFFER,0

    ADDWF S1+1,0
    ADDWF S2-1,0
    MOVWF RESULTADO+1

    MOVF S3,0
    ADDWF RESULTADO,1
    CLRF BUFFER
    RLF BUFFER,0

    ADDWF RESULTADO+1,1
    MOVF S3+1,0
    ADDWF RESULTADO+1,1

ENDM
```

Cuadro de Texto 7. Suma de 16 bits

MAXIMO determina cual es mayor entre |PICO+1|PICO| y |ABSO+1|ABSO|
 si |ABSO+1|ABSO| >= |PICO+1|PICO|, entonces, |PICO+1|PICO| <--- |ABSO+1|ABSO|
 si |ABSO+1|ABSO| < |PICO+1|PICO|, entonces, |PICO+1|PICO| permanece invariado

```
MAXIMO MACRO PICO, ABSO
    LOCAL SEGUNDA, CAMBIE, FIN

    MOVF ABSO+1,0
    SUBWF PICO+1,0

    BTFSZ STATUS,Z
    GOTO SEGUNDA

    BTFSZ STATUS,C
    GOTO CAMBIE
    GOTO FIN

SEGUNDA:
    MOVF ABSO,0
    SUBWF PICO,0

    BTFSZ STATUS,C
    GOTO FIN

CAMBIE:
    MOVF ABSO,0
    MOVWF PICO
    MOVF ABSO+1,0
    MOVWF PICO+1

FIN:
    ENDM
```

Cuadro de Texto 9. Valor máximo

CORRER_BYTE desplaza un bloque de memoria especificado por:
 ULTIMO+1 = dirección más alta del bloque
 PRIMERO = dirección más baja del bloque
 CONTADOR especifica el registro a ser usado como contador

```
CORRER_BYTE MACRO ULTIMO, PRIMERO, CONTADOR
    LOCAL LOOP, a

a = (ULTIMO-PRIMERO)+1

    MOVLW PRIMERO+1
    MOVWF FSR

    MOVLW a
    MOVWF CONTADOR

LOOP:
    MOVF INDF,0
    DECF FSR,1
    MOVWF INDF
    INCF FSR,1
    INCF FSR,1

    DECFSZ CONTADOR,1
    GOTO LOOP

ENDM
```

CORRER_WORD desplaza un bloque de memoria, en formato de 16 bits,
 especificado por:
 ULTIMO+1 = dirección más alta del bloque
 PRIMERO = dirección más baja del bloque
 CONTADOR especifica el registro a ser usado como contador

```
CORRER_WORD MACRO ULTIMO, PRIMERO, CONTADOR
    CORRER_BYTE ULTIMO, PRIMERO, CONTADOR
    CORRER_BYTE ULTIMO, PRIMERO, CONTADOR

ENDM
```

NEGATIVO_DE obtiene el negativo de un número de 16 bits, |COEF+1|COEF|
 Esto se hace simplemente obteniendo el complemento a 2 del número.
 Se supone que el valor significativo se encuentra en COEF.
 COEF+1 es la extensión del signo.

```
NEGATIVO_DE MACRO COEF

    COMF COEF,1
    MOVLW 1
    ADDWF COEF,1

    CLRF COEF+1
    BTFSZ COEF,7
    COMF COEF+1,1

ENDM
```

Cuadro de Texto 8. Corrimiento y negativo

VALOR_ABS obtiene el valor absoluto de |COEF+1|COEF| y lo
 guarda en |REGIS+1|REGIS|.
 BUFFER se usa como registro auxiliar.

```
VALOR_ABS MACRO COEF, REGIS, BUFFER
    LOCAL LOOP, LOOP1

    BTFSZ COEF+1,7
    GOTO LOOP

    COMF COEF,0
    MOVWF REGIS
    COMF COEF+1,0
    MOVWF REGIS+1

    MOVLW 1
    ADDWF REGIS,1

    CLRF BUFFER
    RLF BUFFER,0
    ADDWF REGIS+1,1
    GOTO LOOP1

LOOP:
    MOVF COEF,0
    MOVWF REGIS
    MOVF COEF+1,0
    MOVWF REGIS+1

LOOP1:
    ENDM
```

Cuadro de Texto 10. Valor absoluto

Conclusiones

El proyecto realizado resulta adecuado al objetivo propuesto, el cual simplemente fue discriminar cuatro bandas distintas sin asociarlas a alguna aplicación específica.

En el mismo desarrollo del proyecto se notó que la barra de leds sólo necesitó la información concierne a los bits 8,7,6 y 5 de VALOR_PICO.

La información de salida es sensible al número de muestras que se tomen. Inicialmente se hizo solamente con 10 muestras; sin embargo, al aumentar la cantidad a 50, la diferencia entre la suma, cuando la señal de entrada coincide con la frecuencia de la banda pasante y las que no, se hace más notoria y por tanto se facilita la lectura de la barra de leds.

Nótese que el número de bandas puede ser superior a 4.

Se sugiere al lector algunas aplicaciones de la idea básica expuesta, con la observación de que ninguna de ellas ha sido probada aún y que es necesario recalcular las frecuencias de muestreo para satisfacer las nuevas condiciones:

- Display de ecualizador gráfico.
- Afinador de instrumentos musicales.
- Detector de fonemas por medición de la presencia de sus formantes.

Se han implementado diversos filtros tanto IIR como FIR en microcontroladores ya sea para aplicaciones de control discreto u otras, con muy buenos resultados. Sin embargo, cuando se trata de análisis espectral el tiempo de ejecución de los algoritmos supera normalmente las expectativas de lograr procesamiento en tiempo real

```

BARRASI MACRO VALOR, PUERTO, SAL
LOCAL FIN, FIN2, CANAL0, CANAL1, CANAL2, CANAL3, CANAL4

MOV LW 0F0
MOVWF SAL
MOVF VALOR+1,1
BTFSZ STATUS,Z
BCF SAL,7

BTFSZ SAL,7
GOTO FIN
BTFSZ VALOR,7
BCF SAL,6
BTFSZ SAL,6
GOTO FIN
BTFSZ VALOR,6
BTFSZ SAL,5
GOTO FIN
BTFSZ VALOR,5
BCF SAL,4
FIN:
BTFSZ STATUS,Z
GOTO FIN
BTFSZ STATUS,Z
BTFSZ CANAL3,7
BCF CANAL3,7
XORWF DIAL,0
BTFSZ STATUS,Z
BTFSZ CANAL2,7
CANAL1: MOV LW 0F0
ANDWF PUERTO,1
IORWF SAL,0
GOTO PUERTO,1
CANAL0: MOV LW 0F0
ANDWF PUERTO,1
MOVE SAL,0
IORWF PUERTO,1
GOTO FIN2
CANAL2: MOV LW 0F0
ANDWF PUERTO,1,1
MOVE SAL,0
IORWF PUERTO,1,1
GOTO FIN2
CANAL3: MOV LW 0F0
ANDWF PUERTO,1,1
SWAPF SAL,0
IORWF PUERTO,1,1
FIN2:
ENDM
LIST
    
```