PUNTO FIJO VERSUS PUNTO FLOTANTE:
PRINCIPIOS BASICOS DE FUNCIONAMIENTO
VENTAJAS Y DESVENTAJAS EN EL CASO PROTOOLS

Este artículo puede descargarse en formato pdf del sitio www.andresmayo.com/data

Introducción:

Las aplicaciones de DSP (Digital Signal Processing, o Proceso Digital de la Señal) en el campo del audio profesional son cada día más variadas y complejas, al punto de que ya no podemos concebir el proceso de producción de un disco sin la intervención de alguna forma de DSP.

Por suerte, la tecnología permite hoy que estos dispositivos sean relativamente económicos y fáciles de conseguir. Sin embargo, no es suficiente con disponer de gran potencia de procesamiento y rango dinámico, ya que la respuesta de un sistema depende en gran medida de la correcta elección de los algoritmos y de las arquitecturas de sistema, que se dividen en dos grandes grupos: Arquitectura de Punto Fijo y Arquitectura de Punto Flotante.

No debe confundirse la capacidad de procesamiento de un chip de DSP (medible, entre otras cosas, por la cantidad de bits que puede manejar en forma simultánea) con la resolución inherente al sistema de audio que utilizamos. Por ejemplo, mientras que el standard de resolución en que trabajamos nuestras mezclas es de 24 bits, los circuitos integrados trabajan internamente con longitudes de palabra de 48 o 64 bits, o incluso bastante más en algunos casos. Lo que diferencia internamente a estos chips es precisamente su arquitectura, es decir la forma que utilizan para representar digitalmente la señal que estamos procesando.

Teoría de las Arquitecturas:

La arquitectura de **Punto Fijo** fue introducida a comienzos de la década del ´80, y está basada en una representación que contiene una cantidad fija de dígitos después del punto decimal. Al no requerir de Unidad de Punto Flotante (FPU), la mayoría de los chips DSP de bajo costo utilizan esta arquitectura, aunque en determinados casos esta alternativa ofrece también mejor performance o mayor exactitud.

Los bits a la izquierda del punto decimal se denominan *bits de magnitud* y representan valores enteros, en cambio los bits a la derecha del punto decimal representan valores fraccionales (potencias inversas de 2). Es decirque el primer bit fraccional es ½, el segundo es ¼, el tercero es 1/8, etc.

Podemos calcular la representación en punto fijo de un número dado de la siguiente manera:

$$2^{m-1} - \frac{1}{2^f}$$

Para representar los números positivos y

$$-2^{m-1}$$

para representar los negativos, donde *m* son los bits de magnitud y *f* son los bits fraccionales.

Por lo tanto, de acuerdo a esta fórmula, si disponemos de 16 bits en total y utilizamos 11 bits para representar los enteros (*m)* y los restantes 5 bits para los fraccionales (*f)*, encontramos que el máximo número positivo representable es 1023,96875. En cambio, si asignamos *m* =12 y *f* =4, el mayor número positivo que podremos representar es 2047,9375 y si *m*=13 y *f*=3 este número resulta ser 4095,875.

Vemos entonces que la arquitectura de punto fijo nos permite representar magnitudes mayores sólo a costa de reducir la precisión después del punto decimal. La pérdida de precisión en los sistemas de punto fijo se produce típicamente en operaciones matemáticas en las que el resultado tiende a ser de mayor orden que los operandos.

0	0	0	1	0	0	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	1	0	1	1	0	0	1
			O	**P**	**E**	**R**	**A**	**N**	**D**	**O**		**A**									**O**	**P**	**E**	**R**	**A**	**N**	**D**	**O**		**B**		

0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	1	1	0	1	1	0	1
O	**V**	**E**	**R**	**F**	**L**	**O**	**W**	**R**	**E**	**S**	**U**	**L**	**T**	**A**	**D**	**O**		**V**	**A**	**L**	**I**	**D**	**O**	**D**	**E**	**S**	**C**	**A**	**R**	**T**	**E**

Esto ocurre por ejemplo en la multiplicación, en la que el producto requiere más bits que los factores: si multiplicamos dos números de punto fijo entre sí (ambos con *m* bits de magnitud y *f* bits de fracción), el resultado puede requerir hasta 2*m* bits para representar los enteros y 2*f* bits para los fraccionales. Típicamente los procesadores de punto fijo toman los bits del medio como válidos y desprecian el resto, por lo tanto se pierden los *f* bits menos significativos (considerados una pérdida "razonable" y los *m* bits más significativos (que deberían valer cero, en caso de que no lo fueran se considera que el resultado es inválido y no puede ser representado en este sistema (condición de *overflow*).

La arquitectura de **Punto Flotante** es más moderna y resulta suficientemente exacta y rápida para la mayoría de las aplicaciones. Es muy frecuentemente utilizada para lograr una buena aproximación del número que se desea representar, pero a menudo requiere de un "redondeo", debido a su limitada precisión. Su representación involucra un número entero (la *mantissa*) multiplicado por una *base* (en nuestro caso la base siempre es 2) elevado a un *exponente*, de tal forma que cualquier número de punto flotante *a* puede ser representado como:

$$a = m \times b^{e}$$

Es posible especificar cuántos dígitos de precisión se requieren, asignando un valor al parámetro *p*.

La gran ventaja de esta arquitectura reside en que permite la representación de un rango de magnitudes mucho más amplio que el de la arquitectura de punto fijo. De acuerdo a la cantidad de bits utilizados para almacenar un determinado número de punto flotante, decimos que éste es de precisión simple (32 bits) o precisión doble (64 bits). En el caso de precisión simple, típicamente se le asignan a la *mantissa* los 23 bits menos significativos (bit 0 a bit 22), luego el *exponente* ocupa los siguientes 8 bits (bit 23 a bit 30) y el bit 31 está destinado a indicar el signo (0 = positivo, 1= negativo). Existen implementaciones de software que emplean hasta 128 bits de punto flotante.

Aplicaciones y algunas conclusiones:

Un caso polémico sobre la aplicación de arquitectura de punto fijo o punto flotante en el campo del audio profesional es el del bus destinado a plug-ins en el sistema ProTools. Mientras que la version económica de ProTools (LE) posee un bus que opera internamente en 32 bits de punto flotante, en las versiones TDM y HD (con arquitectura de mixer de 48 bits de punto fijo) el bus de plug-ins es de 24 bits de punto fijo. Esta característica de diseño llevó inicialmente a muchos reconocidos

Ingenieros de Mezcla y de Mastering a discutir largamente acerca de la conveniencia de utilizar uno u otro sistema. Veamos las verdaderas conclusiones de esta polémica:

- Si nuestro sistema de edición digital permite mezclas en 24 bits, entonces debe trabajar internamente con más de 24 bits de capacidad de procesamiento simultáneo, de lo contrario no se preserva la integridad de nuestro audio. Básicamente, cualquier proceso que se aplique al audio original debería arrojar un resultado en 32 o 48 bits como mínimo, pero con un "piso" de ruido no mayor a 24 bits. De esta forma, cuando se hace el dithering final para volver a los 24 bits originales, la pérdida es realmente despreciable. Esta primera conclusión deja afuera la opción de trabajar en audio profesional con una arquitectura interna de solamente 24 bits de punto fijo.
- La arquitectura de punto flotante no permite el uso de dithering entre las distintas etapas de procesamiento, debido a que el exponente *e* cambia permanentemente, alterando la significación de la *mantissa*. En este sentido, la opción de 48 bits de punto fijo resulta superior dado que se le puede aplicar dithering. Si bien es cierto que en una arquitectura flotante los errores producidos siempre estarán en proporción con la magnitud de la señal procesada (y por lo tanto pueden ser difíciles o aún imposibles de detectar), la opción de dithering es una garantía de que estos errores van a ser eliminados.
- Por otro lado, la verdadera forma de aprovechar las características de un sistema con arquitectura interna de 48 bits pero donde el bus de plug-ins es de 24 bits es utilizando únicamente plug-ins de doble precisión (es decir, que toman el audio sample de 24 bits y lo elevan a 48 para operar internamente y así obtener un resultado de mayor precisión). Estos plug-ins deberán también tener capacidad de realizar dithering a 24 bits para que este proceso de DSP de alta precisión no se pierda en el truncado o redondeo de la información que no puede quedar contenida en los 24 bits de salida. En un sistema como ProTools TDM, el uso de plug-ins de doble precisión con opción de dithering marca una diferencia con respecto al uso de plug-ins de simple precisión.
- El proceso de dithering correctamente aplicado solamente sacrifica rango dinámico (por el pasaje de 48 bits a 24) pero no la linealidad de la señal (lo cual es fundamental). El nivel de error introducido por el dithering en esta etapa está en el orden de -144 dBFS, lo cual es realmente insignificante. En cambio el error producido por el truncado de la señal sin dithering aplicado está en el orden de -100 dBFS (considerablemente mayor).
- Un mismo algoritmo de DSP puede requerir diferentes profundidades de bits en las distintas etapas de su realización. Por ejemplo, un EQ se implementa como un filtro recursivo, donde el feedback juega un papel fundamental. Si el filtro tiene una frecuencia de corte suficientemente baja, la cantidad de feedback generada puede ser muy alta, lo cual amplifica enormemente el error de cuantización original, aumentándolo en dos y hasta tres órdenes de magnitud. Nuevamente, un procesador capaz de trabajar internamente con mayor cantidad de bits asegura una relación Señal a Ruido (SNR) suficiente para amortiguar incluso estas condiciones extremas de uso del DSP.
- Un EQ que trabaja en 32 bits de punto flotante a lo largo de todas sus etapas tendrá una performance de ruido considerablemente peor que un EQ que opera internamente con 48 bits de punto fijo y posee un bus de interconexión de 24 bits
- A su vez, un EQ que opera en 48 bits de punto fijo a lo largo de todas sus etapas será sólo un poco mejor que el que opera con bus de 24 bits, ya que el verdadero responsable del error de cuantización es el alto feedback producido en baja frecuencia.
- Por último, si estimamos el ruido de cuantización producido por la interconexión de plug-ins llegamos a la conclusión de que cada vez que se duplica el número de cuantizaciones, el umbral de ruido aumenta hasta 3 dB. Si suponemos que cada track de una mezcla tiene un máximo de 8 plug-ins insertados, tenemos que el ruido se incrementa unos 9 dB, según la fórmula:

$$3\ dB * \log_2 8$$

Sumando estos 9 dB al umbral de ruido de nuestro sistema de 24 bits (-144 dB) obtenemos un ruido de cuantización total de apenas -135 dB (considerablemente menor que el ruido producido por el propio convertidor), es decir que el encadenamiento de plug-ins no provoca serias degradaciones de calidad de audio, aún trabajando sobre un bus de 24 bits.

- Los filtros digitales, especialmente cuando son de alto Q, tienen respuestas muy variables y con alto grado de error, aún trabajando con 64 bits de punto flotante. En estos casos extremos, la performance de 24 bits punto fijo o 32 bits punto flotante es claramente insuficiente. La solución de 48 bits punto fijo aparece como la más precisa, a la vez que es económicamente realizable. En este caso, la asignación de 8 bits de “extra headroom” (bits 40 a 47) y 8 bits de guarda (bits 0 a bit 7), dejando los 32 bits centrales para el muestreo de la señal dan al sistema suficiente precisión para responder ante condiciones exigentes de filtrado, con la capacidad de sumar hasta 256 canales de audio sin overflow, preservando así un resultado de 24 bits consistente aún después de varias etapas de procesamiento.

Ing. Andrés Mayo
Vicepresidente AES
Región América Latina
aam@aes.org