

Veneto Sound エンジニアのつぶやき

DSD に詳しくなろう！ (1) ~ (6)

<http://blog.veneto-sound.com/?eid=23>

<http://blog.veneto-sound.com/?eid=24>

<http://blog.veneto-sound.com/?eid=25>

<http://blog.veneto-sound.com/?eid=26>

<http://blog.veneto-sound.com/?eid=27>

<http://blog.veneto-sound.com/?eid=28>

DSD に詳しくなろう！ (1)

さて、今回のエンジニアのつぶやきは、前回の予告どおり、今ふたたび、話題になってきた、DSD について、掘り下げてみることにしましょう。題して、“DSD に詳しくなろう！”です。最近、いろいろなオーディオ雑誌に、DSD が取り上げられてきていますので、みなさん、どこかで見たり聞いたりしたことがあると思いますが、はたして、DSD ってなんだろうとか、雑誌を読む限りの多少の知識はあるけど、細かなことはわからないなあという方もいらっしゃることでしょう。また、DSD をよくご存知の方も、なぜ今、DSD なのかって思われたことでしょう。みなさんのこのような疑問が、このブログを読むことで、全て解消できるところまでいけるかはわかりませんが、できるだけわかりやすく解説していきたいと思います。

最近、発売された、PC オーディオを前面に出した雑誌（そのままですね。）に、DSD ファイル再生の記事が載りました。DSD の概要から、PCM との違い、 $\Delta\Sigma$ 変調の概念、そして、DSD ファイルの概要まで、一連の内容を、たった 2 ページで紹介されています。この全ての内容を、2 ページの中に押し込むことは大変なことが伺えますが、それにしても、ニュアンスが違っていたり、明らかに間違っていたりする情報も含まれていました。要は、この記事を書いた評論家の方が、どれほど DSD に関して理解しているかということに尽きるわけですが、理解していないところをさらっと流したつもりでも、文章からボロが出てしまっています。正直、DSD に詳しい方なら、これはちょっと・・・と思われる記事になってしまいました。さらには、DSD に関してあまり知らないという方には、記事が簡略化されすぎていて、なんのことかさっぱりって感じではないでしょうか。結果的に、この記事は、詳しい人には読まれないし、詳しくない人には、この文章を書いた評論家は、いろいろ詳しくてすごいなあって思わせるだけの記事だってことですね。その結果、知らない人には評論家の株が上がるのだから、評論家してみれば書いてよかったということになるのですが、買って読んだ人には、なんのメリットもなく、ちょっと

残念な結果となってしまいました。そこで、このブログでは、DSDに関して、じっくりわかりやすく掘り下げてみて、果たして、この記事が、どこが間違っていたのか、どんなニュアンスのずれがあるのかなど、ひとつひとつ確認していくこととしましょう。

DSD

Direct Stream Digital

そもそも、DSDとは、どんなものなのでしょうか？ DSDとは、ダイレクトストリームデジタル（Direct Stream Digital）の略です。直訳すると、デジタルの直接的な流れ？ 何のことかさっぱりわかりませんね。余計わからなくなるので、直訳はやめにしましょう。ダイレクトストリームデジタル（以降、DSD）は、スーパーオーディオCD（以降、SACD）に使われていますオーディオ信号符号化方式になります。



SUPER AUDIO CD

SACDというのは、CDの次世代規格として、1999年に、ソニーとフィリップスが規格化した高音質用光ディスクで、CDでは、44.1kHz / 16bit のリニアPCMで記録されているのに対して、SACDは、2.8224MHz / 1bit のDSD方式で記録されています。規格が異なりますので、SACDは、通常CDプレーヤーでは再生できず、SACD専用プレーヤーが必要となります。2000年くらいから、高音質再生の流れで、マニア層から普及され、一瞬ブームになったのですが、今は下火になってしまいました。下火になった理由には、SACD独特の音質が上げられ、好き嫌いが分かれる為ということもありますが、それよりは、SACDには、権利保護のためコピープロテクトが施されており、CDのように、個人でリッピングして、CD-Rに焼いて楽しむというようなことができないことが一番に上げられるでしょう。現在、再度ブームに火がつきそうな理由としては、雑誌にしきりに載っているとおり、いままでS

A C Dだけでしか楽しめなかったD S D音源が、ファイル化が進み、P Cでファイル再生が可能となってきたことによります。C Dのときと同じく、“リッピングしてP Cで楽しむ”から、“ハイレゾ音源をP Cで楽しむ”に進んでいったように、D S Dも、S A C D同等の音源から、さらに高解像度のD S D音源へとP C上で楽しむことができるようになってきました。実は、S A C Dのリッピングという話題もあります。PlayStation 3 を使ったリッピングなどが紹介されていますが、著作権などの話は定かではありません。（ということにしておきます。）

さて、D S Dには、そのS A C Dがどの時点でD S D化したかによって、3つの名前があります。“D S Dレコーディング”“D S Dミキシング”“D S Dマスタリング”の3つです。S A C Dのジャケットには、必ずこの3つのどれかが書かれています。S A C Dを製作する段階で区切られており、音声を録音する段階（レコーディング）から、D S Dフォーマットを用いているものを、“D S Dレコーディング”と表示され、録音した素材（アナログ音源やP C M音源）から、ミキシングを行なう段階でD S Dフォーマットに落とされたものを、“D S Dミキシング”と表示しています。また、C Dに落とす段階や、C Dから作り直す段階で、D S Dフォーマットに変更されるものが、“D S Dマスタリング”と呼ばれ、現在のS A C Dは、これが一番多いですね。

なお、S A C Dについて、詳しくのっている H P がありますので、リンクを貼っておきます。ご参照ください。

http://www.sa-cdlab.com/about/about_index.html

また、これは S A C D の表示運用基準のリンクです。S A C D 制作会社でなければ、あまり必要としないものですが、こんな規定があるんだなと眺めるだけでも面白いので、載せておきます。

http://www.riaj.or.jp/issue/ris/pdf/ris_sacd.pdf

さて、D S Dの概要と背景がわかったところで、もう少し深く掘り下げて行くとしましょう。D S D方式とは、いったいどんなものなのか？ 先ほどの文章から、C DのリニアP C Mに対抗するものかなあと、ぼんやりイメージできたことと思います。以前のブログで、P C Mについて触れたことがありました。パルス符号変調のことでしたね。詳しくは、下記にアドレスを載せますので、復習してみてください。

<http://blog.venetor-sound.com/?eid=6>

じつは、D S Dとは、P C Mのように、1 6 b i tとか2 4 b i tとか、多段階の量子化ビットを持っていません。言うなら、1 b i tの信号、つまり、H i g hかL o w、O NかO F F の信号ということになります。以前のブログでは、P W Mに似た手法と書きましたが、詳しくは、P D Mというパルス符号変

調を使っています。PDMとは、パルス密度変調 (Pulse Density Modulation) を略したもので、H i g hの期間/L o wの期間の面積を使って、その密度からアナログ量を表現する手法です。その説明なら、PWM (パルス幅変調) も同じではないかと思われることでしょうか。そのとおりです。アナログ表現の仕方は、非常に似通っています。それでは、PDMとPWMでは、どう異なるのでしょうか？ PWMの概要は、以前このブログでも記事にしました。下記のアドレスです。

<http://blog.venetor-sound.com/?eid=11>

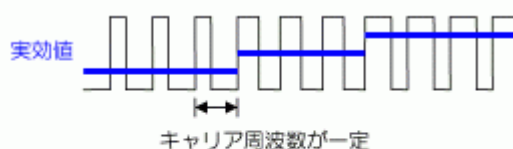


図1 PWM波形

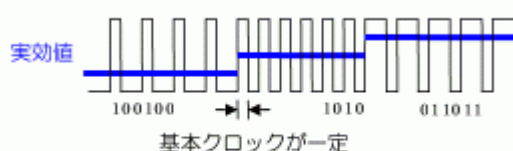


図2 PDM波形

もう一度、まとめてみましょう。図1をご覧ください。これが、PWMを表わした図です。パルスのH i g hとL o wを1パルスと考えると、この1パルス分の周波数のことをキャリア周波数と言いますが、このパルス幅が常に一定になるように動作します。この図では、パルスの立下り位置が左右に移動することで、実効値というアナログ量を変動させています。それに比べて、PDMはどうでしょうか。図2をご覧ください。常に、H i g hもしくはL o wの期間が一定になっていることがわかると思います。H i g h/L o wを繰り返すと、実効値は、半分に、H i g h/L o w/L o wでは、3分の1に、という感じです。このように、PWMの方は、量子化分解能を増やそうとすると、どうしても、パルス幅が大きくなって (キャリア周波数が小さくなる) しまい、元のアナログ信号に対して追従性が悪くなることが推測されます。それに比べ、PDMは、最大追従周波数は、2クロック分であり (実際はそんなことはないが)、より追従性が上がるわけです。しかし、PWMに比べて、制御をかけない状態での分解能は、PWMに劣ります。さらには、ある特定の値の表現に対して、波形むらができてしまい、入力信号には存在しなかった周波数が乗ってしまうこともあります (トーンが発生するという)。このようなデメリットを防ぐ為、微分したり積分したりなどの処理を追加して精度を上げる手段がとられま

す。それが、世に言う「 $\Delta\Sigma$ (デルタシグマ) 変調」です。DSDは、この $\Delta\Sigma$ 変調を用いて、アナログ信号を、HighとLowに分けた、いわゆる、1bitで量子化した方式であるわけです。ですので、DSDのデジタル出力は、PWMと同じように、音声帯域より上の周波数をカットするLPF (Low Pass Filter) を通すことで、簡単にアナログ信号に戻すことが可能になります。この $\Delta\Sigma$ 変調に関しては、次回、詳しく掘り下げてみたいと思います。ちなみに、図2の基本クロックというのが、SACDでは、2.8224MHzとなります。この数値は、CDの44.1kHzの64倍で、オーバーサンプリングしていることとなります。今では、CDの128倍の、5.6448MHzのオーバーサンプリング (DSD128) も存在しています。このオーバーサンプリングという概念も、次回の $\Delta\Sigma$ 変調の回で解説致します。

さて、いかがでしたでしょうか？ なにやら、だんだん難しくなってきたぞと思われたかもしれませんね。しかし、DSDを語るには、まだ序章の段階です。2ページで言えば、まだ最初の10行くらいでしょうか。次回は、第一の山になります、「 $\Delta\Sigma$ 変調」の話です。現在のA/DコンバータやD/Aコンバータには、必ずと言っていいほど使われている変調方式です。すでに理解している方も、おさらいだと思って、読んでみてください。なるべくわかりやすく、イメージしやすい形で書いていきたいと思っています。ご期待ください。

DSDに詳しくなろう！ (2)

さて今回は、前回に引続き、“DSDに詳しくなろう！”の第2弾です。前回は、DSDの概要と背景について触れましたが、みなさん、いかがでしたでしょうか？ 前半の概要は、そんなものかと読み流して頂けたと思いますが、後半、PDMの波形の話になってきますと、ちょっと難しくなってきましたよね。たしかに、波形だけ見せられて、これがなにを表わしているのか、これが、どんな利点につながるのかわからないので、どう判断したらいいかわからないというようなご意見も受けました。たしかにそのとおりですね。ですので、現時点では、PCMや、PWM、PDMのようなデジタルデータをアナログ的に表現する手法があるんだなという理解でいいと思います。これから、回を進めて参りますと、なるほど、こんな利点があるのだなとか、ここに使われているのか、などと話がつながっていきますので、長い目で見ていってください。

それでは、今回は、前回の予告通り、第1の山であります“ $\Delta\Sigma$ (デルタシグマ) 変調”に関して、掘り下げてみましょう。みなさん、オーディオが趣味の方や、そうでなくても、よく雑誌をご覧になる方は、一度は聞いたことがあると思います。最近流行りの、D/AコンバータやA/Dコンバータには、よくこの表現が載ってたりします。なにやらすごいことをやっているのかなと思っ

ているが、使うに挿し当たって影響がないので、詳しくは知らないという方がほとんどではないかと思えます。ですが、ここでは、もう少し“ $\Delta\Sigma$ 変調”に詳しくなってみましょう。そうすることで、DSDにも、デジタルオーディオ全般にも、そういうことだったのかと思われることが拮がり、世に広がる事実や伝説を検証することにつながると信じています。

そもそも、 $\Delta\Sigma$ 変調とは、どこから来たのでしょうか？ まずは内容を理解する前に、背景について触れてみましょう。 $\Delta\Sigma$ 変調とは、早稲田大学理工学部教授の安田靖彦氏が、今から50年前に、まだ、大学院の学生だった頃に創案し、命名されたそうです。それまで使用していたデルタ変調の問題点を解決するために考案されたと述べられています。その辺の開発秘話が、郵政研究所月報に載せられましたので、そのリンクを貼っておきます。興味がある方は、覗いてみてください。

2001.7 郵政研究所月報 巻頭言「技術の生みの親・育ての親」

早稲田大学理工学部教授 安田 靖彦

<http://www.yu-cho-f.jp/research/old/pri/reserch/monthly/2001/155-h13.08/155-foreword.html>

さてそれでは、実際に“ $\Delta\Sigma$ 変調”とはどういったものなのか、進めてまいりましょう。現時点で、この $\Delta\Sigma$ 変調に関して、いろいろな書物やホームページで解説がなされていますが、その多くが、デジタルが得意な学生さん向けだったり、集積回路がご専門な方がアナログ部を理解するためのものだったりしています。たしかにこの $\Delta\Sigma$ 変調技術は、これまでと違って、アナログ部とデジタル部が混在する回路となっている為、集積化（LSI化）するのに、ノウハウが要るからだと思いますが、このブログを読んで下さっているみなさんや、ヴェネターサウンドをごひいきに下さっているお客様には、アナログは得意なだけでデジタルはねえ・・・という方が多く見受けられますので、ここは、限りなくアナログ表現を使って、解説していきたいと思えます。今までの記事で一番アナログ因りだったのは、CQ出版社のトランジスタ技術でしたが、これも、デジタル的などころをいっさいカットした為に、なんのことかわからないところも出てしまったように思われます。今回どこまで説明することができるか出たところ勝負ですが、頑張ってみますので、是非、みなさんもついてきてください。

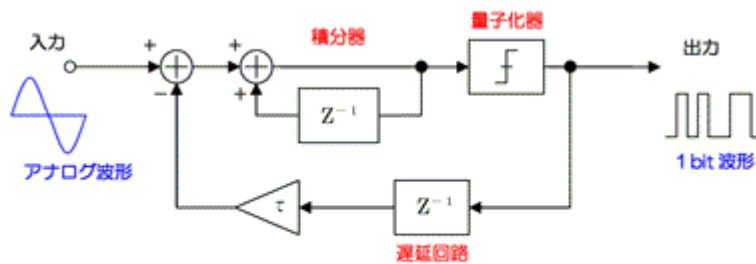


図1 1次 $\Delta\Sigma$ 変調の原理図

さて、のっけから、 $\Delta\Sigma$ 変調の原理図を載せてみます。図1をご覧ください。なんのことかさっぱりという方もいらっしゃることでしょ。わかりやすく説明すると言った先から、難しい回路を取り出してきて、ほんと大丈夫か？と思われたことでしょう。ご安心ください。ひとつひとつわからないところを解説していきます。このブログを読み終わる頃には、なんだ、そんなことだったのかと思われることでしょう。

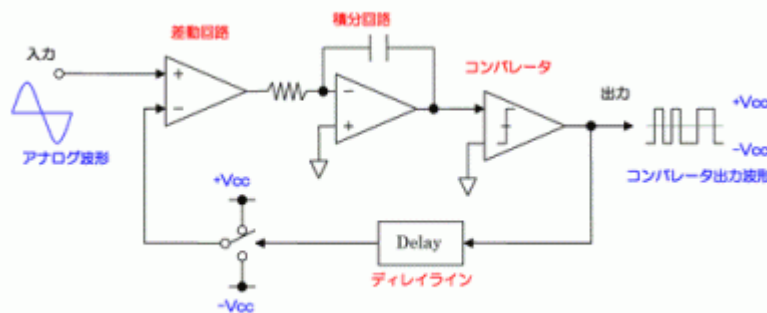


図2 $\Delta\Sigma$ 変調の説明図

アナログが得意な方のためにも、図2とご一緒に見て行くこととしましょう。同じ原理が書いてありますので、比較してみていくと理解が早まるかもしれません。それでは、ひとつひとつ説明していきましょう。まずは、アナログ信号が入力された後で、図1では、なにやら出力側からのフィードバックが書かれています。図2では、オペアンプの差動入力となっています。これは、出力信号との差動アンプを示しています。ここでは、引き算と呼ばさせていただきますが、この引き算が“ $\Delta\Sigma$ ”の“ Δ (デルタ)”に相当します。この“デルタ”とは、数学の授業でもやりましたが、差分とか微分を表わします。前回と今回との差を表わし、直線で言えば「傾き」を表わし、距離で言えば「速度」を表わします。電流・電圧で言えば、「抵抗値」でしょうか。高校教育では、微分・積分を習いましたが、みなさん覚えていますか？ この微分・積分の表現の仕方は、次回、もう少し深く掘り下げてみます。なお、この差動器は、反転入力の足し算ともとれ

ますので、単純に“加算器”という表現をとることもあります。

図1において、さらに進みますと、信号は“積分器”に入ります。図2では、オペアンプによる積分回路を書いてみました。実際にはこれだけでは積分回路として動作しませんが、概念図だと思ってください。この積分器とは、信号をどんどん積算させていく回路のことです。いわゆる足し算ですね。どんどん加算していく回路、高校教育では、“ Σ (シグマ) 演算”と習いました。これが、“ $\Delta\Sigma$ ”の“ Σ (シグマ)”に相当します。この“ Σ (シグマ)”はギリシャ文字で、アルファベットの“S”を表わしていて、英語の summation 「合計」の頭文字をとったものだそうです。

さて、ここで気になる記号が出てきましたね。 Z^{-1} です。ここで、 \wedge とは、乗数を表わし、これは、 Z の -1 乗 となります。この“ Z^{-1} ”とは、 z 変換という、デジタル変換を行なったときに出てくる伝達関数のことですが、ここでは、概念だけに留めておきます。ここで、“ Z^{-1} ”と書かれていたら、デジタル的に、一個前のデータ（前回のデータ）と覚えておいてください。さらに、 z 変換に関して、詳しく知りたい方は、いろいろな参照書が出ていますので、各自、掘り下げてみてください。さて、そんな目で、もう一度、図1の積分器をみてください。“ Z^{-1} ”が前回のデータだと言いましたね。すると、この図では前回のデータが、前側に戻って、足し算させていることがわかります。そうですね。これは過去の数値を積算している、まさに、積分回路となっているわけです。同様に、図1の下の方にも同じく、遅延回路と書いた“ Z^{-1} ”がありますね。これはまさに、前回のデータを取り出すことで、アナログ的に言うと、図2のように、ディレイ回路（ディレイライン）となります。その昔、アナログデータを遅延させるためによく使われたものに、CCDディレイラインというものがあります。CCDとは、よくデジタルカメラの「目」の役割を果たしているものに使われていますが、もともとは、アナログ信号を、バケツリレーの方法で、次から次へと渡していくもので、CCDから出力されてくるのは、過去に入力した信号となるため、信号が一時保存されていることを意味しています。よって、1個のディレイラインがあれば、一回分の過去のデータ（前回のデータ）を取り出せるわけです。図1では、前回のデータを、“ τ (タウ)”と書かれたバッファを通し、先ほど説明した差分回路に流れ込んでいます。これで、現在の入力データと過去の出力データの差分となるわけです。

さて、細かな話は後回しにして、積分器の次は、量子化器ですね。図2では、コンパレータが書かれています。一般に量子化器というと、16ビットとか24ビットに量子化することを想像しますが、ここは、HighかLowの1ビット量子化です。量子化なんてカッコいいことを言っていますが、ここでは、積分器を通過してきたデータが、正電圧なら $+V_{cc}$ 、負電圧なら $-V_{cc}$ を出

力するコンパレータそのものです。そして、出力としては、この信号を使って、 $+V_{cc}$ なら“1”に、 $-V_{cc}$ なら“0”に、置き換えたものが、 $\Delta\Sigma$ 変調回路の出力となります。

以上が、 $\Delta\Sigma$ 変調の回路要素の説明になります。実際、これだけの回路しか入っていません。これだけでほんとに大丈夫なのか？ なにやら狐に摘まれたような感覚に陥りますよね。この回路がきちんと動作するのかを、いろいろな角度から解明した書籍やホームページが存在しますが、数値論は、他の方々に任せるとして、ここでは、概念で理解を深めてみたいと思います。説明が感覚的なものになりますので、数値よりは理解が深まりますが、ニュアンスが掴めないと全く効果を発しませんので、みなさん、一字一句逃さないでくださいね。それでは、見て行きましょう。

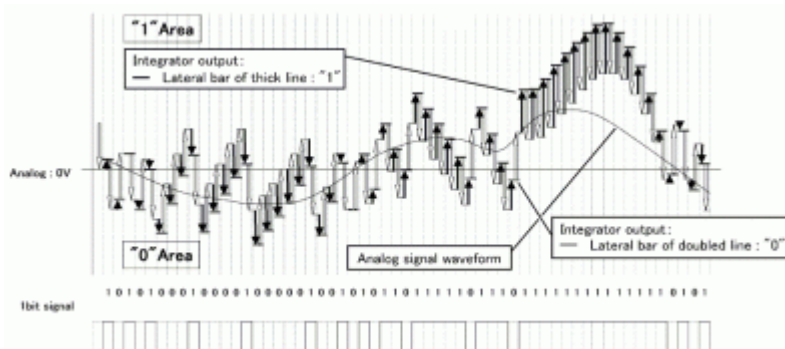
まずは、量子化器ですが、先ほども述べましたが、ここでは、1ビット出力です。HighかLowですね。入ってくる情報はアナログですから、それをHighかLowに振り分けるのですから、ここにはかなり大きな誤差を生じます。いわゆる量子化誤差に相当しますが、この時点での量子化誤差は、計算上6dBです。PCM16ビットの量子化誤差は、96dBですから、もうお話にならないほどです。しかし、 $\Delta\Sigma$ 変調のすごいところは、それを補正しようとする動きが働くことです。デジタルサーボを得意とする方ならすぐおわかりだと思いますが、この出力結果を入力側にフィードバックしています。入力段の差動回路にて、前回の出力値と入力データを差分しています。これにより、指示している値と現在の位置の誤差が算出されています。言わば、指示したんだけど、行き過ぎちゃったとか、まだ追いついていないなどの情報となるのです。この行き過ぎたとか追いつかないっていうのが、量子化誤差になるわけです。次に、その誤差を積分器にて、積算しています。これは、感覚で言うと、行き過ぎが多くなったり、追いつかないことが多くなると、不満が溜まるみたいなもので、過去の怨み辛みを溜めていつているものと想像してみてください。その怨みの溜まった分を吐き出すように、またコンパレータ（量子化器）にかけます。このループを通常より高速に廻すことで、入力信号の追従性を上げようという試みが、 $\Delta\Sigma$ 変調なのです。

この“高速で廻す”というところが、前回紹介した、オーバサンプリングといった手法になります。現在主流になっているのは、2.8224MHzという高速周波数で廻しています。これは、CDの44.1kHzの64倍というスピードに相当します。1ビットという量子化誤差を、CDの64倍のスピードで補正・追従させることで、より元の音声波形に忠実に、デジタル化を行なっているわけです。

さて、 $\Delta\Sigma$ 変調の解説で、けっこうわかりやすく書かれている記事がありますので、ご紹介致します。それは、シャープ株式会社が発行した、2000年8月 第77号 シャープ技報 の1ビットオーディオ の記事です。リンクを貼っておきますので、ご参照ねがいます。

<http://www.sharp.co.jp/corporate/rd/journal-77/pdf/77-14.pdf>

シャープ（株）は当初より、先ほど紹介した、早稲田大学理工学部の安田靖彦教授と共同開発を行い、独自の1ビット符号化技術の開発を行ってきました。この記事は、その開発過程の記事で、とてもわかりやすくまとめてあります。1ビットアンプや $\Delta\Sigma$ 変調に詳しくなりたい方は、是非、一読してください。その中で、 $\Delta\Sigma$ 変調の基本動作説明がありましたので、抜き取って載せてみました。下図をご覧ください。なお、図が少々小さくて、見づらくなってしまったかもしれません。もし、見づらい方は、Shiftボタンを押しながら、図をクリックしますと、別ウインドウで開きますので、図を眺めながら、説明を読んでみてください。



この図で、なめらかなアナログ波形が、入力信号になります。そして、黒塗りの矢印は、入力のアナログ信号の振幅量と電圧の向きを表わしています。その時点でアナログ信号がプラスなら上向きの矢印に、マイナスなら下向きの矢印になっています。矢印の長さが振幅を表現しています。白抜きの矢印は、量子化器からの出力を遅延回路を通して、差動回路に帰還される場所の信号となります。この信号は、HighかLowの2値になっていて、差動回路入力が減算になっているので、符号を反転させて、加算表現になっています。よって、1ならば下向きの信号、0ならば上向きの信号となっています。この矢印の長さは、アナログ量としての加算を表わしており、図1で言えば、 τ 倍のバッファになります。ですので、端的に言うと、矢印の長さは、全て共通で、 τ となるわけですね。そして、黒塗りの矢印の上に横棒がありますが、これが、積分器からの出力になります。この横棒が正の領域に入れば、1重ライン表現になっていて、負の領域に入れば、2重ライン表現になっています。これはすなわち、量子化結果のHighとLowを表わしています。この図から、例えば、入力

信号がプラスの領域にいるときは、積分器からの出力がそれを抑制し、負帰還がかかるように働いています。そして積分器の出力が追いつくと、今度は逆向きに働きます。このように、積分器の一定 (τ) の出力の負帰還が、追いついたり行き過ぎたりを、高速で繰り返すことで、量子化誤差を補正しながら1ビット出力を続けていることがわかると思います。結果、 $\Delta\Sigma$ 変調出力である、1ビット出力は、例えば、Highを+Vccに、Lowを-Vccに対応させておけば、この電圧信号をLPF (Low Pass Filter) に通すことで、元のアナログ信号に復元することが可能であるわけです。

この辺の内容をさらに細かくわかりやすく書かれているホームページがありますので、紹介しておきます。

<http://blog.goo.ne.jp/commux/e/e08c353a7d0d5be49a00c3c61e7b8372>

なお、 $\Delta\Sigma$ 変調のシミュレーションを通して、とてもわかりやすく記載されているホームページがありますので、これも紹介しておきます。Java Applet を使っている数値を変えながらシミュレーションできますのでとても参考になります。

<http://www.okuma.nuee.nagoya-u.ac.jp/~mura/hasi/dsm/behavior.html>

さて、ここまで来て、少し話が進みましたので、前回、雑誌の紹介をしたときに、明らかに間違っていたりニュアンスが違っている文章があったと述べましたが、そのことを思い出してください。ここで、雑誌のある文章を紹介します。

「～、入力されるアナログ信号と出力する1bitデータの差をなくすよう演算を常時行っており、PCMでの量子化誤差のように入力されるアナログ信号の一部を捨ててしまうということが、基本的にはない。」とあります。いかがですか？ たしかに、“差をなくすように演算している”は、合ってますね。しかし、“アナログ信号の一部を捨ててしまうことはない”はどうでしょうか？ ここで、量子化誤差を論じているならば、瞬時の誤差はかなり増大していることを理解しましたね。その誤差を減らすために、高速で演算ループを廻すことをしているのです。また、「信号を捨てる」といったところではどうでしょうか。実は、PCMの信号を、16ビットや24ビット分解能のアナログ信号出力(階段波)で出力した方が、エネルギー的には、捨てるところが少なくなります。 $\Delta\Sigma$ 変調のPDM出力は、あくまで、+Vccと-Vccの矩形波出力ですので、LPFで、余分な信号(エネルギー成分)を吸収・排除する方が明らかに多いし、大変な作業となります。この辺の考え方は、デジタル屋さんには、少々難しいかもしれませんが、このニュアンスの違いは、是非、理解して頂きたいものです。

いかがでしたでしょうか？ $\Delta\Sigma$ 変調の概要を紹介してきましたが、やはり難しかったでしょうか。たしかに、完全に理解しようとしたら、いろいろな文献を

読んだり、数値を解説したりする必要があります。ですので、今日いっきに理解しようとせず、反復しながら理解を深めていけばよいのではないのでしょうか。文献を読めば読むほど、納得と疑問が出てきます。その都度、疑問をつぶして、気づいたら完全に理解していたって感じで進めていきましょう。

今回は、 $\Delta\Sigma$ 変調の概要を掘り下げました。そこには、1ビットという量子化誤差の多さを、帰還をかけることで減らしていく努力をしたものだということがわかりました。その結果、出力されたPDM波形は、LPFを通し、アナログ波形に戻すことが可能となったことも理解できました。それでは、 $\Delta\Sigma$ 変調は、どんな利点があるのでしょうか？ どんなことでも、なにかいいことがないと続かないものです。実は、 $\Delta\Sigma$ 変調には“ノイズシェーピング”というすばらしい効果があります。次回は、ノイズシェーピングとはどんな効果なのかを、z変換をもとに、掘り下げていきたいと思えます。さらに、 $\Delta\Sigma$ 変調を用いたA/Dコンバータの説明から、D/Aコンバータの概略まで進めてみたいと思えます。ご期待ください。

DSDに詳しくなろう！（3）

さてさて、今回も前回に引続き“DSDに詳しくなろう！”の第3弾をお送りいたします。前回の「 $\Delta\Sigma$ 変調」は、流石に難しかったみたいですね。いろいろな方から、難しかったという感想を頂き、また質問や意見も頂きました。引続き、質問や意見など承りますので、お問合せください。今回は、前回に続きまして、もう少し、 $\Delta\Sigma$ 変調を掘り下げてみましょう。前回では、 $\Delta\Sigma$ 変調とはなにか？ 概論を紐解いてみたわけですが、なるほど、そんなことを考えた人がいるのか、でもそれが何に役に立つの？ と思いましたよね。1ビットで表現できたところで、実際にその出力をそのまま、LPFにかけて使っているメーカーを見かけたことがありません。なにかメリットがあって変調しているのでしょうかから、そのメリットを教えて欲しい。了解しました。さっそく、その話に入っていきます。前回の最後に、ちらっと出ましたが、 $\Delta\Sigma$ 変調のすばらしい効果として“ノイズシェーピング”が上げられます。それでは、このノイズシェーピングとは、どんなものなのでしょう。さっそく掘り下げてみましょう。

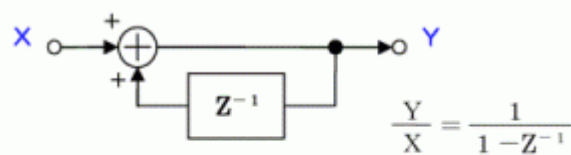


図1 積分器

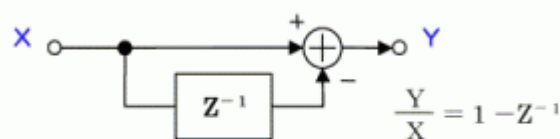


図2 微分器

まずは、概論に入る前に、前回出てきました、 z 変換のおさらいです。 z 変換の詳しい話は今回も割愛させていただきますが、前回、積分器の説明のところで、 z^{-1} という記号が出てきました。これは、デジタル的に、一個前のデータ（前回のデータ）であると書きました。そして、積分器は、前回のデータが、前側に戻って、足し算させている。つまり、過去の数値を積算していることと説明させて頂きました。この辺をもう少しおさらいしてみましょう。図1をご覧ください。これは、前回の $\Delta\Sigma$ 変調の積分器をそのまま抜き出した図となっています。ここで、入力信号を X とし、出力信号を Y とします。だんだん、お勉強になってきましたが、しっかりついてきてくださいね。この図を式に置き換えてみます。

入力部の足し算に着目します。これは、入力 X と、出力からのフィードバックが足されていますね。出力からのフィードバックは、出力 Y の前回のデータということで、式としては、 $Y \cdot z^{-1}$ と表わします。簡単ですね。かければいいのです。“かける”は、「 \times 」でも「 \cdot 」でもいいんですけどね。よって、その足し算が、そのまま出力 Y になっていますから、出力 Y は、入力 X に、 $Y \cdot z^{-1}$ を足したものになり、式としては、

$$Y = X + Y \cdot z^{-1} \text{ です。}$$

これをまとめますと、

$$Y - Y \cdot z^{-1} = X$$

$$Y (1 - z^{-1}) = X$$

よって、伝達関数は、図1に書いてありますとおり、

$$Y/X = 1 / (1 - z^{-1})$$

となります。これは、入力に対する出力のゲイン（利得）みたいなものですね。

この形をよく覚えておいてください。逆に言うと計算した結果、この形になっていたら、その回路は、積分器だということがわかるわけです。

では次に、図2をご覧ください。これは、微分器を表わしています。図をよく見てみますと、今回のデータと、前回のデータの差分をとっていますね。前回と今回の差、つまり、直線なら「傾き」、移動距離なら「時速」、電圧・電流なら「抵抗値」というものです。高校教育では、この辺を、微分で表現していますね。思い出してください。さて、これを先ほどと同様に、式にしてみましょう。今度は、出力部の引き算に着目をして、入力 X から、 X の前回のデータ $X \cdot Z^{-1}$ を引けばいいんですね。さっそく式にしますと、

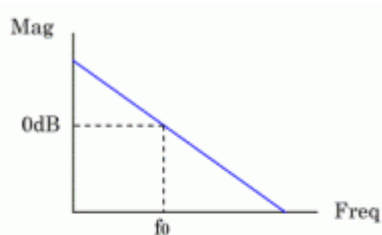


図3 積分器の周波数応答

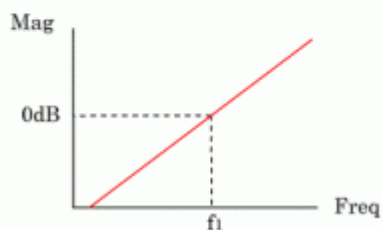


図4 微分器の周波数応答

$Y = X - X \cdot Z^{-1}$ です。

これをまとめますと、

$$Y = X (1 - Z^{-1})$$

よって、伝達関数は、図2のとおり、

$$Y/X = 1 - Z^{-1}$$

となります。先ほどの積分器と比べると、 $(1 - Z^{-1})$ が分子に来ています。ここが重要で、計算した結果、この形になったら、その回路は、微分器であることがわかるわけです。

それでは、積分器や微分器は、こういった特性を持っているのでしょうか。ここは、式ではなく直感を頼りにしますと、積分は、前回と今回の足し算なので、同じデータが入ってくると、どんどんうなぎのぼりに数が増えていきますよね。これがDC成分なら無限大に増えていくことでしょう。しかし、高周波成分では、プラスとマイナスが反復する為、ほとんど積算されません。これをグラフに表わしたのが、図3になります。今度は、微分を考えてみましょう。微分は、

引き算なので、前回と今回の差が著しく大きい場合には、とても大きい値になります。逆に、差がない場合、特に等しい場合には、値はゼロですね。それをグラフに表わしたのが、図4になります。どちらも、0 dBを指し示す周波数は、サンプリング周波数に関係します。ちなみに、傾きを表わすのは、微分や積分の次数（何回、微分・積分をやったか）に関係しますが、これについては、もう少し後で説明します。

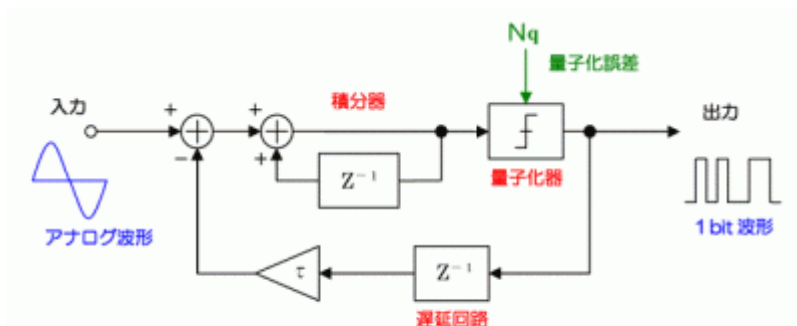


図5 1次 $\Delta\Sigma$ 変調の原理図

さて、数学のお勉強を進めてまいりましたが、実際にこれを役立ててみましょう。もう一度、 $\Delta\Sigma$ 変調の原理図を載せましたので、図5をご覧ください。今回は、量子化器のところに、量子化誤差を入れてみました。 Nq が量子化誤差です。それでは、先ほどと同じように、入力を X として、出力を Y として、 X と Y と Z^{-1} と Nq を使って、図5を式にしてみましょ。まずは、積分器に入力される信号を式にしてみますと、入力 X から、出力 Y の前回のデータ $Y \cdot Z^{-1}$ を引けばいいんですね。今回は、計算を簡単にするために、 τ (タウ) は1として、考えないようにします。

$$X - Y \cdot Z^{-1} \text{ --- (a)}$$

次に、積分器は、先ほどの結果から、

$$1 / (1 - Z^{-1}) \text{ --- (b)}$$

よって、積分器の出力は、(a) \times (b) でよいので、

$$(X - Y \cdot Z^{-1}) / (1 - Z^{-1}) \text{ --- (c)}$$

ここまでよいですね。そして、量子化器に入りまして、この量子化誤差 Nq が重畳されます。(c) + Nq となります。式で書くと、

$$(X - Y \cdot Z^{-1}) / (1 - Z^{-1}) + Nq \text{ --- (d)}$$

ですね。 Nq を足しただけです。そして、この量子化器の出力が、この回路の出力でもあるわけですから、それが、そのまま、 Y となります。よって、きちんと書き直すと、

$$Y = (X - Y \cdot Z^{-1}) / (1 - Z^{-1}) + Nq \text{ --- (e)}$$

となり、これを Y についてまとめますと、

$$Y(1 - Z^{-1}) = (X - Y \cdot Z^{-1}) + N_q(1 - Z^{-1})$$

$$Y(1 - Z^{-1}) + Y \cdot Z^{-1} = X + N_q(1 - Z^{-1})$$

$$Y = X + N_q(1 - Z^{-1}) \quad \text{--- (f)}$$

出ました！これが、 $\Delta\Sigma$ 変調の伝達関数（量子化誤差入り）です。これを見てわかるとおり、出力 Y は、入力 X に量子化器の量子化誤差分 N_q を足したものであることがわかります。そして、 N_q の後ろをよく見てください。なにか思い出しませんか？そうです。先ほど予備勉強をしたところの、微分器がくっついていることがわかります。これは、単純に量子化器の誤差が足されているのではなく、量子化誤差を微分した結果を足していることになります。ということは、どういうことかということ、図4の微分器の周波数特性を思い出してください。通常ノイズというのは、周波数に関係なく、どこも一様に存在しているのが普通です。それをホワイトノイズというのですが、量子化誤差も同じことが言えます。しかし、今回の場合は、その量子化誤差成分を微分器にかけることで、一様ではなくなり、図4のように、高周波の方へぐぐっと寄せ集められたように、シフトした分布となるのです。実はこの、量子化誤差成分の低周波成分が、高周波成分にシフトすることを“ノイズシェーピング”と言われていいます。実際には、可聴帯域に存在している周波数領域のノイズを、可聴帯域以上にシフトしてやることで、可聴帯域の量子化ノイズを低減させるのに使われています。まとめますと、 $\Delta\Sigma$ 変調には、PCM変調にはない、量子化誤差を微分することにより、ノイズ成分を音声帯域外に叩き出し、音声帯域内を低減させる、ノイズシェーピング効果があるということがわかりました。

さらに音声帯域内の量子化ノイズを低減させる手法としては、今回の説明では、積分器はひとつだけでしたが、その積分器の数を増やすことで、微分の傾斜がきつくなり、よりノイズが減ることが確認されています。くわしくは、前回ご紹介したシャープ技報を参照願います。

そして、さらに、この辺の内容がよりわかりやすく書かれているHPがありますので、ご紹介しておきます。

<http://blog.goo.ne.jp/commux/e/85923e13aa4a1a4006d740bce30b84db>

さてここで、恒例の、雑誌記述の添削を試みましょう。雑誌には、「～サンプリング周波数が非常に高いので、雑音はより広い周波数帯域に分散し、高域になるほどノイズは増加する。ノイズシェーピングなどの対策技術が非常に重要。」とあります。 $\Delta\Sigma$ 変調のノイズシェーピング効果について触れられているところは、ここだけの模様です。わずか2ページに収めるには、これだけになってしまうのも致し方ないことだとは思いますが、あまりにもガサツな書き方で面食らってしまいました。いろいろ突っ込みたくなる文章ですね。“サンプ

リング周波数が高い”イコール“雑音の周波数帯域が広い”っていうのもよくわかりませんが、百歩譲って、これを一様のノイズ成分だとしても、“高域になるほどノイズが増加する。”この文章では、そのことがなにか悪いことのように聞こえてきますね。そして、その対策に、“ノイズシェーピングなどの対策が重要”。いままでこのブログを読んで下さった方なら、話が本末転倒していると気づいてもらえましたでしょうか？ 正解は、「広域に渡るノイズ成分を、ノイズシェーピングという効果で、高周波にシフトさせることで、音声帯域のノイズ量を低減させている。」ですね。どう読まれても、おかしくならないように、はぐらかして書かれたつもりでしょうが、結局は、記事を書いた評論家の方の、理解度が足りないことが浮き彫りになっています。まあ、こういう人たちが、オーディオを都市伝説に変えていくのだなあと思いました。我々技術者が、きちんとわかりやすく、読者の方々やユーザの方々に真実を伝えていくことが、真のオーディオの発展につながると信じて、これからも頑張っってブログを続けて参りたいと思っております。

さて今回は、実際に計算式などを表わして進めてまいりました。当初は、 $\Delta\Sigma$ 変調を使ったA/DコンバータやD/Aコンバータの説明まで行きたいと思っておりましたが、ここで休憩としましょう。この辺のお話は、次回に廻したいと思います。今回までで、 $\Delta\Sigma$ 変調のすばらしさがわかってもらえたと思います。次回は、その技術を使って、A/DコンバータやD/Aコンバータがどのように出来上がっているのかを紐解いてみたいと思います。それにより、マルチビットPCMデータの録音・再生と、DSDデータの録音・再生の違いがどうなっているのかまで解説できたらと思っております。ご期待ください。

DSDに詳しくなろう！（4）

さて、今回も引続き、“DSDに詳しくなろう！”の第4弾をお送りいたします。このデジタル解説ももう4回目となりましたが、もう少しお付き合い願います。前回までで、 $\Delta\Sigma$ 変調とは何か？ $\Delta\Sigma$ 変調のすばらしいところなど、話を進めて参りました。こんなすばらしいものがあるなら、なにかに使われているのではないか、その通りです。現在では、ほとんどと言っていいほど、A/Dコンバータや、D/Aコンバータに使われています。それではさっそく、A/Dコンバータについて、掘り下げるとしましょう。

まずは、A/Dコンバータですが、これはみなさんご存知のとおり、アナログ信号（Analog）をデジタル信号（Digital）に変換する（Converter）ものですね。我々が扱っているのは、主にオーディオ信号ですが、他にも、映像信号や制御信号など、いろいろな信号を、アナログからデジタルに変換する際に使われます。ひと昔まえのA/Dコンバータには、サ

サンプルホールド回路という技術が使われていました。これは、アナログ信号を、コンデンサなどにより、あるタイミングによって、一定時間保持することで、その電圧を読み取り、16ビットや32ビットなどに量子化する回路が使われていました。しかし、この方法では、サンプル期間を短くする（高速サンプリング）ことが困難であり、また、部品ばらつきによる量子化誤差を吸収するために、IC内部をひとつひとつ調整する必要があり、生産性が上がらないというデメリットがあったそうです。

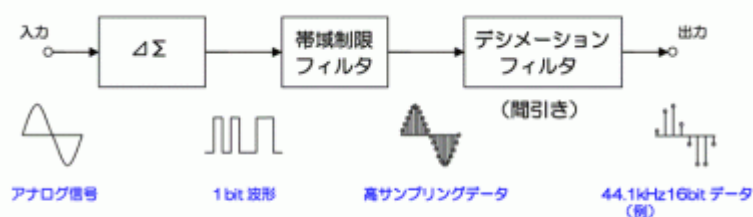


図1 A/D変換 概念図

そこで登場したのが、この $\Delta\Sigma$ 変調方式のA/Dコンバータです。そもそも、 $\Delta\Sigma$ 変調の回でお話ししたとおり、 $\Delta\Sigma$ 変調の出力が、0と1の1bitデジタルですので、これだけでも十分、A/Dコンバータとなり得ますが、この $\Delta\Sigma$ 変調の出力を、PCMに変換したものを、A/Dコンバータと呼びます。それではさっそく、概念図をみていただきましょう。図1をご覧ください。アナログ信号が入力されて、最初のブロックが、ご存知“ $\Delta\Sigma$ 変調”です。ここで、1ビットのDSD信号に変換されます。通常は、2.8224MHzという高速クロックで表現されています。ここまでは、前回までのおさらいですね。次に“帯域制限フィルタ”というブロックに入ります。ここで、高サンプリング周波数のマルチビットPCMデータに変換されます。詳しい説明は、後に廻しまして、さらに、次のブロックの“デシメーションフィルタ”に入ります。“デシメーションフィルタ”なんてカッコいい名前ですが、やっていることは、ただの「間引き」です。高サンプリング周波数データを、何回かおきに、データを取り出しています。2.8224MHzのデータから44.1kHzのデータにするには、64回に1回、データを取り出せばいいという具合です。デシメーションフィルタの処理は簡単ですので、前段の帯域制限フィルタと1つのデジタルフィルタとして実装されていることが多いです。これが、A/Dコンバータの概念図になります。 $\Delta\Sigma$ 変調の出力からは、単なるデジタルフィルタですので、個別調整するようなところがなく、1ビット処理のように、高速で処理できることがわかってもらえたと思います。今回の説明では、出力を44.1kHz/16bitと表現しましたが、あくまで、デシメーションで落としているだけで

すので、さらに、高解像度サンプリングまで対応できます。実際に、 $2.8224\text{MHz}/1\text{bit}$ のDSDデータを、 $176.4\text{kHz}/24\text{bit}$ のPCMデータに変換しても、遜色なく聴くことができます。この場合、間引きは、16となります。

さて、A/Dコンバータの概念は理解できました。それでは、帯域制限フィルタとは、どんなフィルタなのでしょう。これが理解できたら、A/Dコンバータ卒業です。あと少し、頑張ってください。この帯域制限フィルタとは、デジタルフィルタそのものです。いわゆる、FIR型フィルタとかIIR型フィルタというものです。ここの帯域制限フィルタは、一般的に、FIR型のデジタルフィルタを使用することが多いです。このFIR型とは、日本語では、有限インパルス応答とって、過去の有限個のデータから計算し、フィードバックの処理を行わないフィルタです。フィードバックがないため、演算が速く、ストリーミング演算に向いているとされています。逆に、IIR型とは、無限インパルス応答とって、出力を戻すフィードバックが存在しています。よって、計算結果も過去のデータが無限に影響し、演算も大変になってきます。このIIR型は、アナログフィルタと似た特性に作ることができるので、音質面での評価がいいことも挙げられます。ここでは、詳しいデジタルフィルタの解説は割愛させていただきますが、興味がある方は、かなり参考書が出回っていますので、各自、掘り下げてみてください。

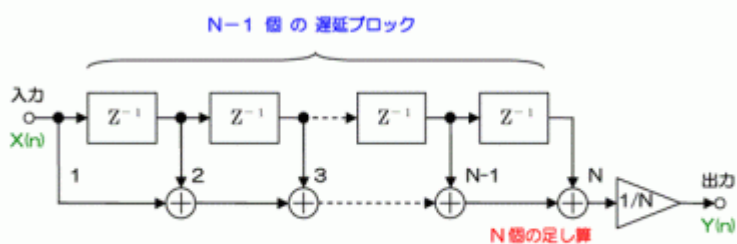


図2 移動平均フィルタ概念図

ここの帯域制限フィルタは、FIR型の中でも、移動平均型フィルタをよく使用します。図2をご覧ください。前回学習した、 Z^{-1} が出てきました。これは、 z 変換の伝達関数で、前回のデータという意味でしたね。この図では、 $N-1$ 回までの過去のデータが蓄積されていることとなります。それらのデータを単純に足しまして、最後に、 N で割っています。これが、移動平均型フィルタのブロック図になります。この移動平均型フィルタとは、FIRのローパスフィルタの重み付け係数を1にしたものに相当します。このフィルタは、あらかじめ N 個の遅延ブロック分の演算をしておけば、あとは、その演算結果と新

規に入ってきたデータと最後のデータから演算することが可能で、いちいち毎回、全てN個のデータから演算する必要がないため、Nが増えても演算が重くなることなく、処理が簡略化されるというメリットを持っています。また、周波数特性が、正弦関数をその変数で割った（ $\sin X / X$ ）、sinc関数であることから、SINCフィルタとも呼ばれています。この帯域制限フィルタには、1ビット出力からの信号を、例えば、ビットレートが16ビットなら、1ビット出力「1」を、0x7FFFに、1ビット出力「0」を、0x8000に置き換えて、入力させることにより、移動平均フィルタ演算によって、16ビット分解能に変換していきます。A/Dコンバータでは、このフィルタを使って、1ビットのマルチビット化とローパスフィルタの帯域制限を行なっています。

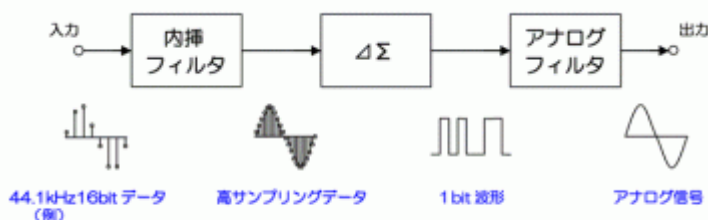


図3 D/A変換 概念図

さて次は、D/Aコンバータについて掘り下げてみたいと思います。とは言っても、実は、概略はもう全て説明されているのです。しかし、若干、順番や名称が変わりますので、順を追って、みていくことにしましょう。図3をご覧ください。まずは入力です。この場合、CD音源を例に挙げてみました。44.1kHz/16bitのデータが入力されるとします。その信号が最初に“内挿フィルタ”を通ります。これは何をするとところかという、オーバーサンプリングを行ないます。44.1kHzのサンプリングを、例えば、2.8224MHzまでオーバーサンプリングします。64倍ですね。それから、内挿フィルタで、あいだを補間します。この“内挿フィルタ”とは、英語で、インターポレーションフィルタ（interpolation Filter）といい、デジタルローパスフィルタの一種ですが、サンプルのある場所では、元のサンプル値をそのまま使い、サンプル間を補間するようにサンプルを作成していくフィルタです。FIRの形となっています。A/Dコンバータのときのデシメーションフィルタとちょうど逆の処理をしてくれます。これで高サンプリングデータになったところで、 $\Delta\Sigma$ 変調です。A/Dコンバータの時は、ここまでアナログ信号だったのですが、D/Aコンバータでは、完全にデジタル構成となっています。そして、 $\Delta\Sigma$ 変調で、ノイズシェーピングを行い、分解能を1ビットにまで低下させます。その

後、ポスト・ローパスフィルタで、ナイキスト周波数以上をカットして、アナログ信号に変換させ、出力します。この図を見ておわかりのとおり、離散信号を、オーバーサンプリングすることで、よりアナログ信号に近い状態にしておいてから、 $\Delta\Sigma$ 変調をかけるという動作を行なっています。ほとんど考え方は、A/Dコンバータそのものですね。

さて、大急ぎでA/Dコンバータ・D/Aコンバータを掘り下げてきました。ここまでくると、実は、マルチビットPCMの信号の流れが理解できているのに気づくでしょう。そして、DSDデータの録音・再生との違いもわかってしまいます。とあるPCオーディオ雑誌には、いきなり“データ処理の複雑度”とあり、わけのわからない表や、意味のわからない語句が書かれてあって、ちんぷんかんぷんだったと思いますが、今や、全てがわかるようになっているはずで

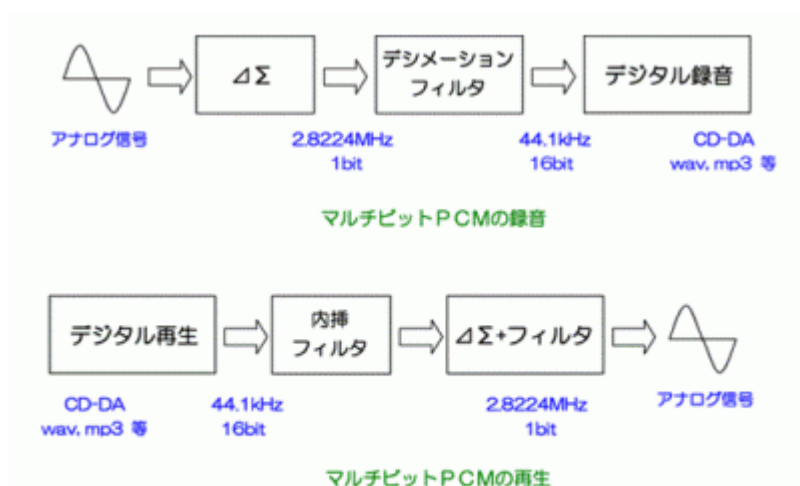


図4 マルチビットPCMの信号の流れ

それでは、図4をご覧ください。上側の図が、PCM音源の録音時の信号の流れを表わしています。どこかで見たことのある図ですよ。先ほど掘り下げた、A/Dコンバータと同じブロックです。実は、世に言う、PCM録音とは、今やどこのものも、 $\Delta\Sigma$ 変調によるA/Dコンバータそのものです。A/Dコンバータから出力されたデータを、CDに焼いたり、パソコンで扱えるデータフォーマットに並べ替えたりしているだけなのです。そして、下側の図を見てください。今度は、再生の流れになります。CDやパソコンデータを再生したことを想定していますが、これも、よく見ると、先ほどやった、D/Aコンバータと同じですね。まあ、あたりまえと思われるかもしれませんが、もう一度まとめさせて頂きました。

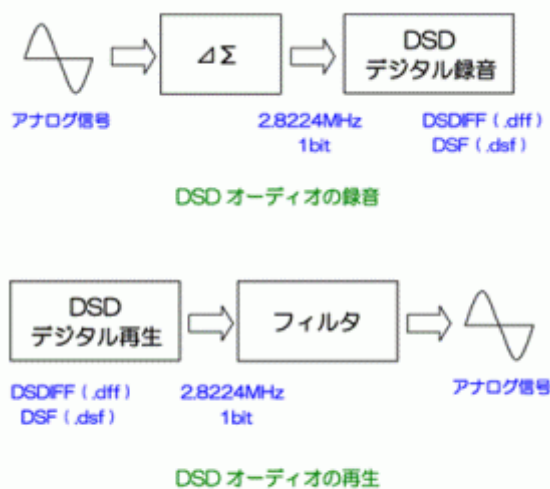


図5 DSDの信号の流れ

それでは、次の図5をご覧ください。今度は、DSDオーディオの信号の流れですが、主に、スーパーオーディオCDの録音・再生に使われていましたが、最近では、パソコン上で再生可能なファイルが見直されています。上側が録音の流れで、下側が再生の流れになります。図4と比べてとてもすっきりしていることがわかります。デシメーションフィルタだの内挿フィルタだのいろいろなデジタルフィルタに囲まれたPCM音源に比べて、DSDの方は、処理がかなり簡略化されています。音質には、できるだけフィルタを通さないのが原則（フィルタを通せば通すだけ、音が悪くなっていく）ですので、これが、DSDの最も魅力的なところになります。

さて、A/Dコンバータ・D/Aコンバータの理解から始まり、マルチビットPCMの録音・再生とDSDの録音・再生の比較をしてきましたが、いかがでしたでしょうか？ 最初は、いかにも難しいことをやっているという印象だったものが、なんだ、そんな簡単なことだったのかと思われたのではないのでしょうか。

A/DコンバータやD/Aコンバータに使われているデジタルフィルタは、処理が簡単とか、計算が速いとかで選ばれています。サンプリング周波数など、いくら音声帯域よりかなり高いところでの演算だとはいえ、フィルタを形成していることには変わりません。もしかしたら、この辺のデジタルフィルタを、音質重視に作り変えることができたならば、より高音質な音源になるかもしれませんね。

また、いままで掘り下げてきたところは、“ICの中だけ”というイメージで進めてまいりましたので、「所詮、きちんと理解しても、ICを作るわけではない

ので、宝の持ち腐れだなあ。」なんて思われているのではないのでしょうか。実は、それがそうでもないのです。それは、次回以降、掘り下げてみようと思っています“ファイルフォーマット”という存在です。先ほどの信号の流れのところでも触れましたが、今まで、DSDというと、SACDのみでしか聴くことができなく、しかも、アナログ信号に変換された後しか扱うことができなかつたため、本当のDSDの良さを実感することが困難でした。これは、私の勝手な思い込みかもしれませんが、SACDのあの「広がりがない、ナローな音質」は、決してDSDが悪さをしているのではなく、後段の嫌なデジタルフィルタや、外部のギンギンにかましたアクティブフィルタによるものではないかと想像しています。

そういった中、最近のPCオーディオブームの煽りで、DSDをPCで楽しむ機会が増えてきました。ということは、DSDの生データが簡単に手に入るようになったということです。その生データを使って、今回掘り下げました、帯域制限フィルタやデシメーションフィルタをパソコン上で構成し、演算してやることで、いわゆる、DSD/PCM変換が、ご自身でできるようになったということです。パソコン上で計算させるのに、ソフトウェアが得意な方なら、BasicやC言語を使えばいいでしょうし、そんなものを使わなくても、Excelが使えたら、できてしまいます。なんか、楽しみが広がってきたとは思いませんか？

今回は、世の中に広まっているDSDデータのファイルフォーマットの中で、SONY提唱のDSFフォーマットを、その次の回では、フィリップス提唱のdffフォーマットを、掘り下げてみたいと思います。乞うご期待ください。

DSDに詳しくなろう！（5）DSDファイル再生って何？

さて今回で、“DSDに詳しくなろう！”の第5弾です。そして、ようやく、サブタイトル“DSDファイル再生って何？”を掲げることができました。このタイトルは、某PCオーディオ雑誌のタイトルと同じですね。やっとのこと、この話ができるようになったということです。過去4回に渡って、DSDの基礎講座を行なってまいりましたが、今までの内容と、これから、2回に分けて、ファイルのフォーマットについて掘り下げてみますが、これらを、2ページという限られたスペースに、押し込むことができた評論家の方に、エールを送りたいと思います。

さて、嫌味はこれくらいにしまして、さっそく、DSDファイル再生について、掘り下げてみましょう。前回までで、DSD方式について、掘り下げました。その結果、CDを代表とする、PCM録音・再生に比べて、DSD録音・再生の処理の方が、格段に、簡潔でわかりやすくなっていることがわかりました。

そして、PCMには、内挿フィルタや帯域制限フィルタなど、多くのデジタルフィルタを通ることによる音質劣化の懸念までわかってきました。

そんなすばらしいDSD録音ですが、今までは、民生レベルでは、スーパーオーディオCD（SACD）に採用されているものの、実際、音楽を聴くには、DSD信号が出力されている機器はほとんどなく、アナログ変換されたものしか聴くことができませんでした。そんな中、昨今のPCオーディオブームの一環でしょうか、DSDディスクというものが見直されてきて、数は少ないですが、DSDディスクを再生できる機器が登場してまいりました。これまでの経験から、何とかディスクと聞くと、また難しいフォーマットで、録音・再生に難しい方式を採用しているんだろうなあって思ってしまう。しかし、このDSDディスクというのは、そんな難しいものではなく、DVDなどのメディアに決められたファイルシステムでフォーマットして、決められたディレクトリ構成で決められたファイルを保存するという取り決めがあるだけなのです。決して、特殊なcodecがあるわけではないのです。この文章を読んで、すぐおわかりになった方もいらっしゃると思いますが、まさに、パソコンでのファイル管理と同じですね。デジカメが詳しい方なら、カメラファイルシステム規格（DCF）に似ています。そして、音声ファイルは、拡張子が、dsfで表されている、DSFファイルになります。デジカメなら、jpgとかbmpとかになりますかね。このdsfファイルフォーマットの中身に関しては、後ほど掘り下げてみたいと思います。

DSD

Disc Format

もう少し、DSDディスクについて、掘り下げてみましょう。そもそも、DSDディスクは、SONYが提唱しています、民生向けのDSDディスクフォーマットのことで、これまでのSACDでは、権利保護のためにコピープロテクトが施されている為、個人で作成する用途には不向きだったところを、個人レベルで録音・再生できる方式として考え出されました。CDのように、聴くだけでなく、作る楽しみというものが、DSDもようやく参加してきたということでしょうか。対応しているメディアは、いちおうDVDメディア（DVD-RやDVD-RWなど）と書かれていますが、特に限定するものではないそ

うなので、これからは、USBメモリなどにも、応用が効くのではないかと思います。

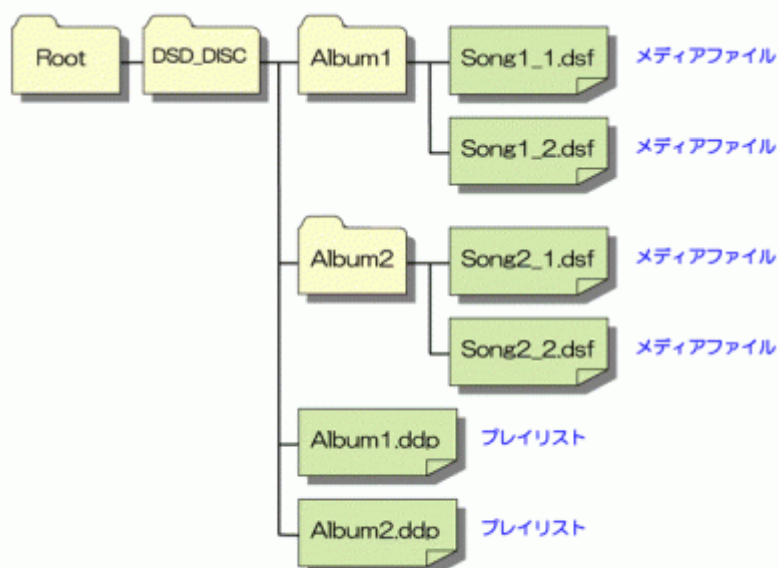


図1 DSDファイルディレクトリ構成の例

それではさっそく、ディスクの中身を見ていきましょう。図1をご覧ください。これは、DSDディスクの中身のディレクトリ構造を模した例です。ルートディレクトリ直下に、“DSD_DISC”というフォルダを作って、その中に、DSDファイルを格納すると定められています。その中には、アルバムごとに、フォルダ化されていたり、これは任意とされていますが、プレイリストが配置されていたりしています。特に難しいことはありませんね。デジタルオーディオプレイヤーなども、似たようなディレクトリ構造を持っていますので、ここには、特に新しい考え方などは入っていません。

それでは、実際のDSD音源が入っています、DSFファイルのフォーマットについて、掘り下げていくことにしましょう。まずは、拡張子の説明からですが、ファイル拡張子は、“dsf”です。これは、“d s d s t r e a m f i l e”の略だそうです。全てを訳すと、「D i r e c t S t r e a m D i g i t a l S t r e a m F i l e」ですね。どれだけ“Stream”が好きなんだって話です。さて、気を取り直して、ファイル構造の話に入っていきます。以前、リニアPCMの話のところで、WAVファイルの構造に触れたことがありました。

<http://blog.venetor-sound.com/?eid=7>

そこでは、WAVファイルには、「チャンク (c h u n k) 」と呼ばれるブロック構造によって、音声データが記録されていると述べました。実は、このD

DSFファイルも同じく、チャンクというブロック構造を採用されています。DSFファイルの場合は、4つのチャンクから作られています。それでは、詳細をみていきましょう。表1をご覧ください。

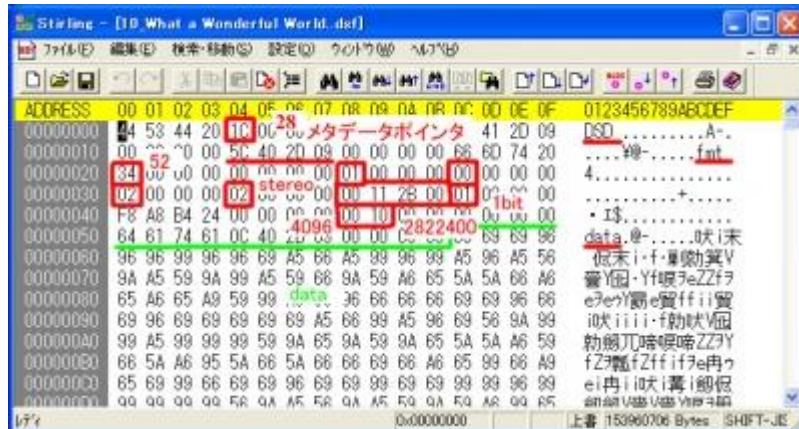
byte	内容	備考	チャンク
4	"D" "S" "D" " "	最後に1文字スペースを含む	DSD チャンク
8	28	このチャンクのサイズ	
8	size	ファイルのトータルサイズ	
8	pointer	metadataチャンクへのポインタ	
4	"f" "m" "t" " "	最後に1文字スペースを含む	fmt チャンク
8	52	このチャンクのサイズ	
4	1	フォーマットバージョン	
4	0	0: DSD raw	
4	チャンネルタイプ	1: mono 2: stereo 3: 3 channels 4: quad 5: 4 channels 6: 5 channels 7: 5.1 channels ※1	
4	チャンネル数	1: mono 2: stereo ... 6: 6 channels	
4	サンプリング周波数 [Hz]	2822400, 5644800	
4	サンプルビット数 [bit]	1, 8	
8	サンプル数	※2	
4	チャンネルごとのブロックサイズ [Byte]	4096 ※3	
4	0	reserve	
4	"d" "a" "t" "a"		data チャンク
8	size	(n + 12) bytes	
size	音声データ	※4	
size	metadata チャンク	ID3v2	metadata チャンク

表1 DSFファイルの構造

順番に、DSDチャンク、fmtチャンク、dataチャンク、metadataチャンクの順番に並んでいます。各チャンクごとに、決められたバイト数ごとに表1のとおり配列されています。順番に説明していきますと、まずは、DSDチャンクです。まず、4byteで、“D”“S”“D”“スペース”のキャラクターが入り、次に、このチャンクのサイズ（28）、ファイルのトータルサイズ、そして、メタデータが入る場合は、metadataチャンクへのポインタが入ります。metadataチャンクがない場合は、ゼロを埋めます。次に、この音声ファイルの情報として、fmtチャンクの説明です。まず、“f”“m”“t”“スペース”のキャラクターが入り、このチャンクのサイズ（52）、そして、ファイルフォーマットのバージョン（1）、フォーマットIDのDSDraw（0）が入ります。次からが、DSDファイルの個別情報となって、チャンネルタイプ。これは、インターリーブの形式を表わしています。インターリーブの詳細は、後に、DSFファイルフォーマットの仕様書のリンクを貼っておきますので、それを参照してください。ちなみに、ステレオモードのインターリーブは、1：左チャンネル 2：右チャンネルの順に並びます。次が、

チャンネル数です。これは単純にチャンネルの数がきます。そして、次が、サンプリング周波数です。通常は、2.8224MHzですが、最近では、5.6448MHz (DSD128) も出てきています。次のサンプルビット数は、1ビット表現か、8ビット表現かを表わしています。もともと、DSDですので、1ビット表現ですが、ここの数値によって、音声データの配列の仕方が変わってきます。音声データのところで説明致します。次が、サンプル数です。これは、1チャンネルごとのサンプル数になります。次が、チャンネルごとのブロック数です。これは固定値(4096)となります。ステレオモードの時は、インターリーブのとおり、左チャンネル、右チャンネルの順に、4096(0x1000)Byteごとに並んでいることを表わしています。なお、音声データの末尾のブロックが合わず、使われなかった領域はゼロ(0x00)で埋めるよう指示されています。最後は予約領域があります。現状は、ゼロで埋めます。次のチャンクは、dataチャンクです。いよいよ、音声データですね。まず、“d”“a”“t”“a”のキャラクタと、チャンクのサイズが入りまして、音声データとなります。先ほどのサンプルビット数のところでも出ましたが、このサンプルビット数の表現によって、音声データの配列が変わります。詳しく見ていきましょう。まず、サンプルビット数が1ビットの場合、音声データは、LSB (Least Significant Bit) ファースト形式で保存されています。これは、8ビット表現をした時、下位ビットから出力されることを意味してしまっていて、例えば、データチャンクに、0x80と書かれていたら、下位ビットから出力されるので、00000001・・・と出力されていきます。次に、サンプルビット数が8ビットの場合はというと、MSB (Most Significant Bit) ファースト形式で保存されています。これは、8ビット表現した時に、上位ビットから出力されるので、データチャンクに、0x80と書かれていたら、10000000・・・と出力されます。

以上がDSFファイルの構造でしたが、文章だけではちょっとわかりづらいですね。それでは、実際のファイルの中身を見ていきましょう。次の表を見てください。



これは、DSFファイルを、バイナリエディタで中身を覗いたものです。このファイルは、PCオーディオfan2の付録音楽データDVDに収録されている「この素晴らしき世界」2. 8MHz DSDファイルを引用させて頂きました。左側がアドレス、真ん中がデータの羅列で、右側にはキャラクタが表示されています。それでは、先ほどの構造表と照らし合わせていきましょう。まず、右側のキャラクタから、“DSD”がありますね。そして、チャンクサイズの28、その後、トータルサイズが入りまして、meta dataチャンクへのポインタです。ここには、表示しませんでしたでしたが、アドレス0x092D405Cには、メタデータが入っていました。次が、“format”キャラクタを確認して頂いて、このチャンクサイズが52、フォーマットバージョンが1、フォーマットIDが0と書かれています。そして次が、チャンネルタイプ、このデータは、ステレオだということがわかります。チャンネル数も2です。そして、サンプリング周波数になりまして、2822400(0x002B1100)が入っているのが確認できます。次がサンプルビット数で、1が入っていますので、先ほどの説明より、音声データは、LSBで取り出されることがわかります。次が、サンプル数です。サンプル数をサンプリング周波数で割ってやると、演奏時間が出ます。

$$0x24B4A8F8 \div 0x2B1100 = 0xDA \text{ (218秒)}$$

よって、だいたい、3分38秒くらいの演奏時間ということがわかります。そして、ブロック数は、4096ですね。目が慣れてくると、右側のキャラクタをスクロールしていくと、4096(0x1000)ごとに音声が変わるのがわかってきます。(左チャンネルと右チャンネルが変わりますので、キャラクタのつながりが途切れるところがあります。)

さらに、dataチャンクにまいりまして、“data”を確認、チャンクサイズを確認したところで、いよいよ音声データとなっています。

データの最初の方は、9とか6が目立ちます。1ビットで表現しますと、1001とか0110とかになっています。例えば、表では見えづらくなってしまいましたが、データ出だしは、

9 6 6 9 6 9 9 6

ですね。これをLSBで抜き出してみますと、

0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1

です。これは、いたい、2つ「1」がきて、2つ「0」がくる感じに見えます。これを、前回お話しした、帯域制限フィルタをかけることによって、おおよそ、ゼロレベルの音圧に変換されるわけです。2つずつのHighとLowなので、2.8224MHzの半分の1.4112MHzになるわけですが、帯域制限のLPFでカットされるため、ほぼゼロ電圧となるわけですね。ここは、いわゆる、曲の先頭の無音領域ですので、当然といえば当然です。実際に、この音声データを抜き出して、お手製のデジタルフィルタで、帯域制限フィルタとデシメーションフィルタを作れば、このデータから、リニアPCMの、WAVデータに変換ができるということです。機会がありましたら、実際に、手製のデジタルフィルタでの変換・逆変換などで、音質の違いを楽しんでみたいと思います。

なお、最新のDSDディスクフォーマットの仕様書や、DSFファイルフォーマットの仕様書は、下記のHPから入手することができます。理解を深めるためにもダウンロードしてみてください。

<http://www.dsd-format.sony.net/ja/index.php>

また、ネットにいろいろ転がっているものも合わせて載せておきます。

<http://www.ps3sacd.com/downloads/DSDDiscFormatSpecs.pdf>

http://sacd-ripper.googlecode.com/svn/trunk/docs/DSF_file_format_specification_E.pdf

いかがでしたでしょうか？今回は、DSDディスクの紹介と、実際に、DSFファイルの中身まで覗いてみました。その結果、意外と簡単な配列で、情報データと、音声データであるDSDデータが並んでいることがわかりました。今まで、いろいろと掘り下げてきましたが、それらを使うと、DSDデータからリニアPCMデータ(DSFファイル⇒WAVファイル)への変換ができるようになったことになります。さて次回は、DSFファイルとともに、一般に知られています、DSDIFFファイルの中身にも、掘り下げてみたいと思います。そして、長々と続けてまいりました、“DSDに詳しくなろう！”の最終回となります。乞う、ご期待ください。

DSDに詳しくなろう！ (6)

長々と続けてまいりましたが、今回の“DSDに詳しくなろう！”の第6弾で、DSDの話も、千秋楽となります。某PCオーディオ雑誌の記事がさっぱりわからないというご用命からスタートして、早や6回が経とうとしています。みなさん、いかがだったでしょうか？完全に理解できなくても、だいたいの概論がわかって頂ければ、幸いです。

さて、前回は、DSDファイルのフォーマットに触れまして、SONYが提唱しています、DSFファイルの中身について、掘り下げてみました。その中身は、「チャンク (chunk) 」と呼ばれるブロック構造になっていて、実は、決められたフォーマットに基づいて、配列されているだけだということがわかりました。今回は、同じくDSDファイルフォーマットで、フィリップスが提唱しています、DSDIFFファイルについて、掘り下げてみたいと思います。それではさっそく、DSDIFFについて掘り下げていきましょう。実は、このDSDIFFファイルも、DSFファイルと同じく、チャンク構造になっています。ですので、前回のDSFファイルをある程度理解していれば、今回は、とても簡単です。とは言っても、このDSDIFFは、DSFと比べて、チャンクの量も多く、さらに、圧縮データまで、取り扱うことができますので、完全に理解しようとしたら、けっこう大変な作業となってしまいます。そこで、今回は、全てのチャンクは、概論に留めさせて頂いて、必須チャンクのみを細かく掘り下げまして、さらに、実際のファイルの中身を参考に、理解を深めるやり方をとりたいと思います。さらに詳細な内容は、あとで資料を紹介致しますので、各自、掘り下げてみてください。

まずは、概論からですが、このDSDIFFとは、Direct Stream Digital Interchange File Formatの略です。ファイルの拡張子は、dff になります。いろいろ他のデータファイルとぶつかりそうですが、オーディオ的には、この拡張子でいけるでしょう。先ほども触れましたが、このDSDIFFには、ロスレス圧縮技術を使った、DSTというエンコード方式ものせることが可能です。今回は、このDSTについては、割愛させて頂きますが、詳しく知りたい方は、あとで資料をのせておきますの

で、調べてみてください。



図1 DSDIFF チャンクツリー構造

ここで、DSDIFFとDSFの違いについて触れますと、DSFは、数字データは全て、リトルエンディアン (Little Endian) で書かれていましたが、このDSDIFFは、ビッグエンディアン (Big Endian) で書かれています。これは、データが8ビット (1バイト) を超えて格納されているときに (例えば、2バイト・4バイトなど)、下位バイトから順に格納されているか、上位バイトから格納されているかの違いでして、DSFは前記を、DSDIFFは、後記に対応しています。この違いは、なにやら歴史や出処に関わっているらしいのですが、私はそんなに詳しくないので、事象だけに留めさせていただきます。あとで実際に、ファイルの中身を見ていくときに、違いがわかってもらえるのではないかと思います。他にも、チャンクサイズ量には、IDやckDataSize分を含むか含まないかなどの違いがありますが、詳細は、あとで紹介する資料を参考にしてみてください。

次に、DSDIFFのチャンク構造について見ていきましょう。図1をご覧ください。DSFと違って、ツリー構造になっています。ひとつのチャンクの中に、複数のローカルチャンクが入っているのがわかると思います。この図の「必須」と書かれているところは、記入が義務付けられています。それに対して、「任意」というところは、入れても入れなくてもよいところとなっています。なお、最初のチャンクは、バージョンチャンクと指定されていて、DSDチャ

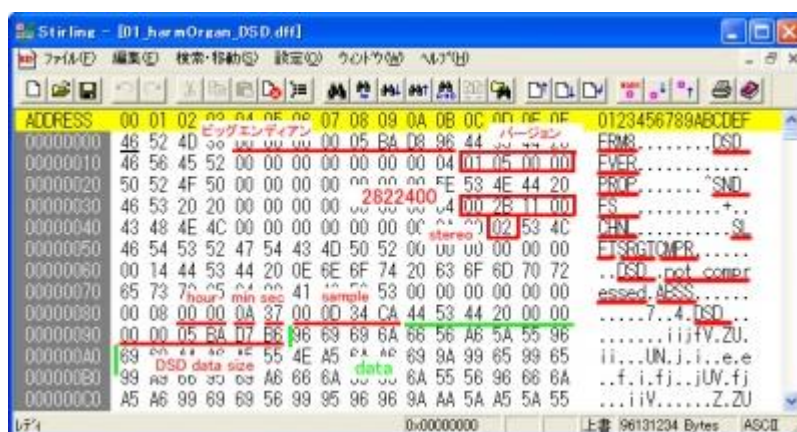
ンクもしくは、DSTチャンクの前に、プロパティチャンクを入れることが指定されています。あとの任意のチャンクの順番は、特に指定されていません。DSDチャンクとは、非圧縮のDSD音声ストリームデータが入っています。また、DSTチャンクとは、DSTエンコーディングされた圧縮データが入っているチャンクになります。よって、これらは、どちらか片方のみとなります。

byte	内容	意味	チャンク
4	"F" "R" "M" "8"	4-byte ID	FRM8 チャンク
8	total flsize - 12		
4	"F" "V" "E" "R"	最後に1文字スペースを含む	FVER チャンク
4	バージョン	3211000000 - ver1.5.01 DSDIFF	
4	"P" "R" "O" "P"	プロパティチャンク	PROP チャンク
8	chunk data size	PROF size - 12	
4	"S" "P" "A" "C" "E"	最後に1文字スペースを含む	FS チャンク
4	サンプリング周波数 [Hz]	2822400, 5644800	
4	"C" "H" "N" "L"	チャンネルチャンク	CHNL チャンク
8	chunk data size	CHNL size - 12	
2	チャンネル数	1: mono 2: stereo ...	
2	ステレオの場合	...	
4	チャンネルID	"LEFT" : stereo left "RIGHT" : stereo right "MLFT" : multi-channel left "MRGT" : multi-channel right "LS" : multi-channel left surround "RS" : multi-channel right surround "C" : multi-channel center "LFE" : multi-channel low frequency enhancement (C, D007, C, 999) : content specific content	
4	"C" "M" "P" "R"	圧縮タイプチャンク	CMPR チャンク
8	chunk data size	CMPR size - 12	
4	comp type	"DSD" : 非圧縮タイプ "DST" : DST圧縮タイプ	
1	comp name	comp name の長さ	
8	comp ratio	"not compressed" : 非圧縮タイプ "DST encoded" : DST圧縮タイプ	
4	"A" "B" "S"	絶対時間指定チャンク	ABS チャンク
8	chunk data size	ABS size - 12	
2	hours	[0 - 23] range	
1	minutes	[0 - 59] range	
1	seconds	[0 - 59] range	
4	samples	[0 - SampleRate-1] range	
4	"D" "S" "D"	圧縮タイプ無し(文字列)・非圧縮タイプの場合	DSD チャンク
8	size	DSD size - 12	
8	音声データ	サンプリングレート・bits/channel 最上位bitsが時間的に古いビットデータ	
4	"C" "O" "M" "T"	コメントチャンク	COMT チャンク
8	size	COMT size - 12	
2	コメント数	コメント文字列の数	
2	timeStampHour	タイムスタンプ 時	
1	timeStampMin	分	
1	timeStampSec	秒	
1	timeStampMilliSec	ミリ秒	
2	commentType	コメントの種類	
2	commentRef	コメントの参照番号	
4	comment size	コメントの長さ	
8	comment	コメント	

表1 DSDIFFファイルの構造

それでは、実際によく使われるチャンクを取り出し、表にまとめたものを見ていきましょう。表1のDSDIFFファイルの構造をご覧ください。少々表が大きくなってしまいましたので、別ウインドウで開いて(s h i f t+クリック)、一緒にみてもらうといいかもしれません。この表は、先ほどのツリー構造の中から、必須なチャンクとよく使われるチャンクを並べたものです。なお、後ほど、実際に、DFFファイルの中身と照らし合わせて進めてまいります。まず、FRM8チャンクです。“FRM8”のキャラクタが入り、チャンクサイズが入ります。全てにおいて、サイズは、チャンクIDとチャンクサイズ表示分の、4 + 8 = 12を、差し引いた値が入ります。次は、DSDチャンクフォームを示す“DSD”キャラクタが入ります。次からが、DSDチャンクフォームになります。まず最初に、“FVER”チャンクです。バージョンが入っています。そして次が、プロパティチャンクです。このプロパティチャンクには、ローカルチャンクがあって、FSチャンク・CHNLチャンク・CMPRチャンク・ABS

Sチャンクが入っています。順を追って説明していきますと、まず、FSチャンクには、サンプリング周波数が入っています。通常は、2.8224MHzですが、高解像度の5.6448MHzもあります。次が、CHNLチャンクで、チャンネル数とチャンネル順が入っています。ステレオの場合は、チャンネル数が2で、チャンネルIDは、ステレオ左・ステレオ右の順になっています。そして次がCMPRチャンクで、格納されているストリームデータが圧縮か非圧縮かを示しています。非圧縮の場合は、“DSD”と、圧縮の場合は、DSTエンコーディングによるものわかるように、“DST”と書かれており、それに伴う文字列が入っています。次のABSSチャンクは、任意ではありませんが、記載されていることが多いので、載せておきました。時・分・秒・残りのサンプル数の順番で格納します。ここまでの、PROPチャンクとなります。他にも、LSCOチャンクがありますが、ここでは割愛します。そして、いよいよ、音声データが格納されているDSDチャンクになります。先ほどの圧縮タイプによって、このキャラクタも変わります。なお、1サンプル分は、8ビット表現となっています。ですので、ステレオですと、左チャンネル8ビット/右チャンネル8ビットの順で、交互に書かれていきます。なお、DSDデータは、1ビットですので、上位ビットが時間的に古いデータとなります。ここでも、MSB (Most Significant Bit) ファーストとなっています。そして、音声データが格納された後に、コメントチャンクがあります。これも任意ですが、多くのDSDIFFファイルに格納されていますので、載せてみました。COMTのキャラクタの後に、チャンクサイズがあり、コメントの数と、タイムスタンプが書かれています。次のcomTypeとcomRefは割愛します。そして最後に、コメントが書かれています。



さて、例によって、表と文章ではわかりづらいので、実際のDSDIFFファイルの中身と照らし合わせてみましょう。次の表を見てください。これはDS

D I F Fファイルを、前回同様、バイナリエディタで中身を覗いたものです。このファイルは、PCオーディオ f a n 5 の付録音楽データDVDに収録されている「01 J. S. バッハ：フルートソナタ 変ホ長調BWV1031～ 第2楽章『シチリアーノ』」D S D I F Fファイルを引用させて頂きました。さっそく見ていきましょう。まず最初に、右側のキャラクタ欄を参照して頂いて、“FRM8”からスタートです。次に、フォームサイズが書かれていますが、ここでお気づきだと思いますが、数字の書き方が、前回のD S Fファイルと違っていますね。D S Fファイルの場合は、左側に数字が張り付いていて、下位バイトから書かれていたのに対して、このD S D I F Fファイルは、右側に数字が張り付いていて、上位バイトから書かれています。これが、エンディアン表現で、D S D I F Fは、上位から書かれているので、ビッグエンディアンということになります。どんどん進めましょう。次に、“D S D ”キャラクタが入り、F V E Rチャンクの“F V E R”が見えます。バージョンが、1. 5. 0. 0 ということがわかります。現在最新ですね。そして、プロパティチャンクの“PR OP”があり、“SND ”IDが入っています。そして、プロパティローカルチャンクに入り、“FS ”でサンプリング周波数がきています。前回のD S Fファイルにもありました、0 x 2 B 1 1 0 0 です。2. 8 2 2 4 M H z ですね。それから“CHNL”キャラクタに続いて、ステレオを表わす、2 があり、“S L F T”“S R G T”で、ステレオ左／ステレオ右の順であることがわかります。そして、“CMPR”キャラクタで、非圧縮ファイルが格納されていることの明示です。“n o t c o m p r e s s e d ”という文言が伺えます。そして任意の“ABS S ”チャンクです。ここでは、10分55秒と書いてありますが、なにを表わしているのかわかりません。トータル演奏時間なら、次のD S D データサイズから算出すると、2分16秒となり、0 0 0 2 1 0 0 0 0 A 5 6 D 8 となるはずですが、そもそも、ここは、絶対開始時間なので、特に制約がなければ、0 0 : 0 0 : 0 0 : 0 0 からスタートすればいいと思われれます。そして、いよいよ、D S D チャンクで、音声データが格納されています。先ほど、非圧縮データと明示されていたので、“D S D ”が書いてあります。その後、データサイズです。このサイズは、8ビット（1バイト）単位ですので、実際は、8をかけて（1ビットD S D なので）、左チャンネルと右チャンネルが入っているので、2で割ると、総サンプル数となります。これが、2. 8 2 2 4 M H z でサンプリングされるので、2 8 2 2 4 0 0 で割ると、演奏時間の秒数が出てきます。当然ですが、雑誌に書かれた、演奏時間とも一緒ですし、プレーヤーにかけても一緒になることでしょう（確認はしていません。あしからず）。さて、データサイズの次から、実際のストリームデータです。ここで注意しなくてはならないのが、先ほども説明しましたが、MSBファーストです。実際に

追っていってみましょう。

データが、9 6 6 9 6 9 6 A 6 6 5 6 . . . と並んでいますね。

左 ⇒ 右 ⇒ 左 ⇒ 右 ⇒ 左 ⇒ 右 の順に並んでいるのだから、

左チャンネル： 9 6 6 9 6 6 . . .

右チャンネル： 6 9 6 A 5 6 . . .

ここまではいいですね。

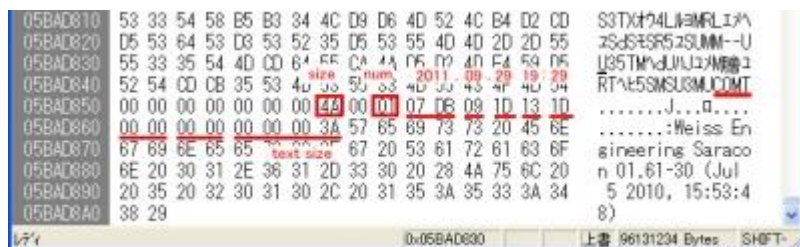
これを、1ビット表現に直しますと、MSBファーストですので、

左チャンネル： 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 0 1 1 0 0 1 1 0 . . .

右チャンネル： 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 1 0 . . .

こんな感じで音声データが流れていきます。

これは、DSFファイルのときの、サンプルビット数が、8のときの動作と同じですね。



さて、次の表は、このファイルの最後付近を表示したものです。音声データの後に、コメントチャンクがあるのがわかります。ここには、タイムスタンプとコメント文が表示されています。このコメントには、コメントタイプによって、チャンネル用コメント・音声ソース用コメント・ファイル履歴用コメントが用意されています。詳細は割愛させていただきます。

なお、DSDIFFの仕様書などのリンクを貼っておきます。詳しく調べたい方は、入手して理解を深めてください。

http://www.sonicstudio.com/pdf/dsd/DSDIFF_1.5_Spec.pdf

http://www.sonicstudio.com/pdf/dsd/DSDIFF_1.5_RecommendedUsage.pdf

これにて、“DSDに詳しくなろう！”のお話も終わりになります。皆様、長々とお付き合い、ありがとうございました。なかなか、時代の最先端のお話を、わかりやすく書くのは、非常に困難が伴いました。どうしても、私自身、あいまいなところもあり、きちんと解説できたかどうか不安ではありますが、とりあえず、やり通したことだけを誇りに思うことにします。もし、落丁乱丁がございましたら、深くお詫び申し上げます。みなさまも、これを基に、難しそうだと思っていたデジタルの世界を、少しでも覗いて見ようかなと思われたとしたら、この題材を取り上げた甲斐があったというものです。もし、今後も取り上げてほしい題材などありましたら、どしどしお問合せください。

さて、次回は、2011年も最後になります。しばらくデジタルのお勉強が続きましたので、一旦、コーヒースタンプを入れましょう。「棚からつぶやき」第3弾をお送りいたします。そして、2012年第1回目は、基本に立ち返り、アナログの基礎の基礎をお送りいたします。乞う、ご期待ください。