Design Wave 設計コンテスト 2005

ディジタルFMレシーバ 設計仕様書

和田知久

今回はディジタル方式のFMレシーバ回路の設計を行います．FM（frequency modulation）は，周波数変調の意味で，音声などのアナログ信号の値の変化（強弱）を正弦波の周波数の変化に変換して電波として伝送する方式です．FM変調は有名な方式なので，多数の通信系の教科書に解説や回路に関する記述があります．しかし，教科書で紹介されている回路はアナログ方式の回路です．今回はFM変調された正弦波信号をA-Dコンバータ（アナログ-ディジタル変換器）にてディジタル化したことを前提に，ディジタル方式のFM復調回路の設計を行います．（筆者）

コンテストなので，小さめのディジタル回路を設計することを念頭に，なるべく簡単な回路構成要素を組み合わせてFM復調を行う回路を設計します．今回設計するディジタル方式のFMレシーバは，入力信号に追随して同期する正弦波発生回路をベースとしています．一般的にはPLL（phase-locked loop）と言われる回路をディジタル回路で構成します．FM変調では正弦波の周波数が変化します．PLLはその周波数の変化する正弦波に同期するので，周波数値に比例する信号も同時に生成されます．まさにその信号を取り出せば，FM復調を行ったことになります．

HDL（VHDLもしくはVerilog HDL）による設計と論理合成を行います．FPGA向けに無償で提供されている設計ツールでも参加できます．また，余裕のある方はFPGAなどへ実装すれば，努力を認められ，高い評価が得られると思います．FPGAへの実装もぜひトライしてみてください．

ディジタルFMレシーバの構成

図1に今回想定するシステムのブロック図を示します．システムは大きく分けて，送信信号生成部（SigGen），そのSigGenの出力値に比例して周波数が変わるVCO（voltage controlled oscillator），A-D変換器（ADC），そしてそのディジタル化されたFM信号を復調するFMレシーバ（FM

図1 システム・ブロック図

Receiver）から構成されます．**図1**では簡略化してSigGenの出力を2値（1，－1）としていますが，0.5などの値をとってもかまいません．その値に対応する正弦波がVCOより出力されます．VCOは電圧制御発振器であり，通常の回路構成では入力値は電圧信号で与えられるので，VCOという名まえになっています．**図1**からわかるように，SigGenの値が－1のときはある低めの周波数の正弦波を生成し，1のときは高めの周波数の正弦波を出力しています．ADCはある一定のクロック周期ごとにアナログ波形の電圧値をディジタルの数値に変換します．**図1**ではディジタル化を示すために，波形を棒グラフで示しています．後で詳しく説明しますが，ADCの出力ビット幅を今回は8ビットに設定しているので，棒グラフの棒の高さは2^8＝256通りの値を取ることになります．FM Receiverはその8ビットの値をクロックごとに受け取って処理します．最終的にはFM復調した波形を出力します．

今回の課題は，ディジタル信号処理をディジタル回路を用いて実現することです．ディジタル回路で整数の演算を行うのは比較的わかりやすいのですが，ADC出力値は正，負の両方の値を取る小数点表現を用います．設計すべきことは比較的単純なので，ディジタル設計の知識のある方は自信を持って課題に取り組んでください．

● PLL（phase-locked loop）

図2はPLLとそのループ・フィルタ出力にローパス・フィルタ（LPF）を接続したものであり，実はこれが今回設計するFM Receiverの全体回路構成になります．単純でしょう！

ここではいくつかの数式を用いてPLLの動作原理を説明します．まず，入力信号はFM信号なので正弦波が入力されます．したがって，入力信号$in(t)$は，

図2　PLLのブロック図

$$in(t) = \sin\left\{\omega_0 t + \theta_i(t)\right\} \tag{1}$$

です．$\theta_i(t)$は，位相項です．定数にすることで，位相を変化させることができます．また，時間によって変化させることで，正弦波の周波数を変化させることができます．

NCOはVCOと同様な発振器であり，正弦波を出力します．ただし，その発振周波数は入力の値で制御されるので，数値（numerical）ということで，NCOと呼ばれています．ディジタル回路のVCOとも言えます．

PLL回路は図のようにフィードバック・ループを形成しており，ループ安定時にはNCO出力は入力とほぼ同じ周波数の正弦波を出力するので，出力信号$ref(t)$は以下のようになります．

$$ref(t) = \cos\left\{\omega_0 t + \theta_o(t)\right\} \tag{2}$$

ここでcosを用いているのは，後に行う計算の過程でつごうが良いからです．この二つの信号を乗算すると，以下のように$c(t)$が求まります．

$$\begin{aligned} c(t) &= in(t) \cdot ref(t) \\ &= \sin\left\{\omega_0 t + \theta_i(t)\right\} \cdot \cos\left\{\omega_0 t + \theta_o(t)\right\} \\ &= \frac{1}{2}\sin\left\{\theta_i(t) - \theta_o(t)\right\} + \frac{1}{2}\sin\left\{2\omega_0 t + \theta_i(t) + \theta_o(t)\right\} \end{aligned} \tag{3}$$

この1項目は$in(t)$と$ref(t)$の位相成分の差であり，2項目は高周波の信号となります．ループ・フィルタでこの高周波成分を除去することにより，位相差成分を検知することができます．これが位相比較器（phase comparator）のしごとです．

ループ・フィルタは乗算結果$c(t)$とDフリップフロップの値に固定値α（＝15/16＝0.9375）を乗算したものを加算する回路になっています．したがって，$c(t)$がループ・フィルタに入力されたときのループ・フィルタの出力は，以下の式で与えられます．

$$loopout = c(t) + \alpha \cdot c(t-1) + \alpha^2 \cdot c(t-2) + \cdots \tag{4}$$

αは1以下の値なので，$c(t)$のこれまでの平均のような演算です．ただし，古いデータほど影響力は小さくなります．平均処理なので，高周波成分をカットする特性を持ち，上記位相比較器の2項目をキャンセルする作用があります．

● NCO（numerical controlled oscillator）

図3にNCOの回路構成を示します．

図3　NCOのブロック図

図4　FIRフィルタのブロック図

詳細な説明を始める前に，システムの動作クロック周波数を16MHz＝16,000,000Hzとし，NCOの出力する正弦波の中心周波数を1MHzとします．したがって，図中に示すように，1MHzの正弦波（周期1μs）では1周期の間に16回のサンプリング点を持つことになります．

NCO回路は，input入力値＝0の場合に16クロック（サンプリング点）で1周期の正弦波を生成したいので，オフセット値として1/16を持つ回路になります．各クロックごとに，Dフリップフロップの値にオフセット値が加算されるので，ちょうど16サイクルで1/16*16＝1になり，加算器の出力accumは1増加します．これは累積計算なので，いわゆるアキュムレータ（accumulator）になっています．アキュムレータ出力に対して2πを乗算し，cosの値をROM（read only memory）から取り出します．cos関数は2πで1周期なので，図中の正弦波をサンプリングした棒グラフのような出力が得られます．

ここでinputの値が正の値であれば，アキュムレータの累積スピードは速くなり，16サイクル以下でaccumは1増加し，1周期の正弦波が生成されるので，高い周波数を生成したことになります．また逆に，inputの値が負の値であれば，アキュムレータの累積スピードは遅くなり，低い周波数を生成したことになります．すなわち，inputの値によって1MHz中心周波数を変化させることができます．

● ローパス・フィルタ（low-pass filter）

一般的にディジタル・フィルタとして有名なのはFIR（finite impulse response）フィルタです．**図4（a）**に16タップのFIRフィルタの構成を図示します．この図から，FIRフィルタの一般式は以下のように与えられます．

$$output(n) = \sum_{i=0}^{15} a_i \cdot x(n-15+i)$$

$$\begin{aligned} output(15) &= \sum_{i=0}^{15} a_i \cdot x(i) \\ &= a_0 \cdot x(0) + a_1 \cdot x(1) + \cdots + a_{15} \cdot x(15) \end{aligned} \qquad (5)$$

ここでa_0，a_1，…，a_{15}の乗算係数を適切に定めれば，特性の良いローパス・フィルタを作ることができます．

しかし，これらの係数を最適とすると図からもわかるように乗算器が16個必要になり，回路規模（面積）がかなり増加します．今回の設計課題では，あまり良い特性ではないのですが，回路の簡略化のために，すべての係数を1/16＝0.0625に固定した移動平均フィルタを代用として使うことにします．

図4（b）に，この移動平均フィルタの回路図を示します．1/16の係数乗算は加算の後で共通化しています．実際には1/16はシフト操作で実現できるので，乗算器は不要となります．

● 固定小数点フォーマット

今回の課題では，上記の1/16＝0.0625のような小数点以下の数をディジタル回路で取り扱う必要があります．例えば4ビットの2進数“0111”ですが，小数点をどこの位置に置くかで表す値が変わってきます．例えば，“01.11”ならば10進数表現では＋1.75，“0111.”ならば，10進表現では＋7です．

0	1	1	0	1	1	1	1
符号ビット	整数部		小数部				

（a）<8，2，t>の意味

0	1	1	0	1	1	1	1
整数部	小数部						

（b）<8，2，u>の意味

図5　固定小数点の属性表記方法

表2　4ビット長の固定小数点の属性のいくつかの例と，対応する2進数，値のレンジ，解像度

	属　性	2進数表現 Sは符号ビット Xはデータビット	レンジ	解像度
整数	〈1，1，u〉	X.	0～1	1
	〈4，4，u〉	XXXX.	0～15	1
	〈4，3，t〉	SXXX.	－8～7	1
小数	〈4，0，u〉	.XXXX	0.0～0.9375	0.0625（1/16）
	〈4，0，t〉	S.XXX	－1.00～＋0.875	0.125（1/8）
その他	〈4，2，u〉	XX.XX	0.0～3.75	0.25（1/4）
	〈4，2，t〉	SXX.X	－4.0～＋3.5	0.5（1/2）
	〈4，5，u〉	XXXX0.	0～30	2
	〈4，5，t〉	SXXX00.	－32～28	4
	〈4，－1，u〉	.0XXXX	0.0～0.46875	0.03125（1/32）
	〈4，－1，t〉	S.SXXX	－0.5～0.4375	0.0625（1/16）

また，4ビットの2進数ですが，それが符号なしの数で正または0の数を表しているのか，2の補数表現で正または負の数を表しているのかを区別する必要があります．“11.10”は，符号なしの数であれば10進表現で＋3.50となり，2の補数表現であれば10進表現で－0.50となります．

すなわち，同じ4ビットの2進数でも，「小数点の位置」と「符号なし表現か2の補数表現か」を明確にしないと，まったく異なった数に対応してしまいます．

ここでは米国CoWare社のディジタル信号処理設計ツール「SPW（Signal Processing Workbench）」で用いられている固定小数点の属性表記方法を紹介します．

<8，2，t>と表記すると，その信号は，全体のビット数が8ビット，整数ビットが2ビット，two's complementということで2の補数表現，すなわち最上位ビットは符号ビットという意味になります．したがって，“01101111”なる8ビットの数の属性が<8，2，t>とすると，**図5（a）**のように小数部は5ビットとなり，10進表現では＋3.46875に対応します．

<8，2，u>と表記すると，その信号は，全体のビット数が8ビット，整数ビットが2ビット，unsignedということで符号なし，すなわち負の数を表さないという意味です．したがって，同じ8ビットの数“01101111”の属性が<8，2，u>とすると，**図5（b）**のように小数部は6ビットとなり，10進表現では＋1.734375に対応します．

表1　属性による値の違い

4ビットの数	属性〈4，2，u〉の場合の10進表現	属性〈4，1，t〉の場合の10進表現
0000	＋0.00	＋0.00
0001	＋0.25	＋0.25
0010	＋0.50	＋0.50
0011	＋0.75	＋0.75
0100	＋1.00	＋1.00
0101	＋1.25	＋1.25
0110	＋1.50	＋1.50
0111	＋1.75	＋1.75
1000	＋2.00	－2.00
1001	＋2.25	－1.75
1010	＋2.50	－1.50
1011	＋2.75	－1.25
1100	＋3.00	－1.00
1101	＋3.25	－0.75
1110	＋3.50	－0.50
1111	＋3.75	－0.25

整数部のビット数が多いほど表現できる最大数，すなわちレンジが大きくなります．そして小数部のビット数が多いほど表現できる最小数が小さくなり，解像度が向上します．

表1に4ビットの数が<4，2，u>の属性を持つ場合と，<4，1，t>の属性を持つ場合の10進表現を示します．

表2に4ビット長の固定小数点の属性のいくつかの例と，対応する2進数，値のレンジ，解像度を示します．**表2**からもわかるように，同じ4ビット長を用いて多様なレンジと解像度の数値を表現できます．当然のことですが，解像度を悪く（大きな値に）すると，より広いレンジを表すことができます．

● ディジタルFMレシーバの回路構成

図6にディジタルFMレシーバの全体回路構成の一例を示します．以下の回路構成で正常に動作することは確認済みです．すでにこれまで各回路の動作原理を紹介しているので，各回路の意味は理解できると思います．

図7にシミュレーション波形を示します．サンプリング周波数を16MHz＝16,000,000Hzとしています．FM変調波出力はサンプリング周波数の1/16の1MHz＝1,000,000Hzを中心に10kHz変調されています．1MHzに対して±10kHzで1％の変調なので，波形を見ただけでは周波数が変化していることはわかりにくいでしょう．ここではサンプ

図6
ディジタルFMレシーバの回路図の一例

図7　シミュレーション波形

上段から，1，−1の送信データの繰り返し，それをFM変調したもの，ディジタルFMレシーバが生成した正弦波（1Kワード×8ビットROMの出力），phase comparatorすなわち乗算器の出力，アキュムレータの出力，そしてFIRフィルタの出力であるdmoutの波形を示している．本来はサンプリングされた値であるが，その値を直線でつないで表示している．

図8　拡大シミュレーション波形

リング周波数を16MHzとしましたが，8MHzでもかまいません．その場合はその比（ここでは1/2）でほかの周波数値がスケールされれば，同じ回路パラメータで動作することになります．

シミュレーションの初期にdmoutの値が大きくオーバシュートしていますが，これは同期途中であり，その後，正常な復調動作を行っていることがわかります．**図8**に一部の拡大図を示します．**図8**ではディジタルの値を棒グラフで示し，ディジタル処理らしい表示にしています．

表3　基本課題用ピン配置

信号名	入出力	ビット幅	説　明
clk	IN	1	クロック入力
reset	IN	1	‘1’でリセット
fmin	IN	8	入力データ　属性は〈8，0，t〉
dmout	IN	12	復調波形出力　属性は〈12，4，t〉

リスト1　50入力XOR回路のVHDLソース・コード

```
library  IEEE;
use IEEE.STD_LOGIC_1164.all, IEEE.NUMERIC_STD.all;

entity PARITY is
    port ( A : in  unsigned(49 downto 0);
           Y : out std_logic );
end PARITY;

architecture RTL of PARITY is
begin

    process(A)
    variable TMP : std_logic;
    begin
        TMP := ‘0’;
        for i in 0 to 49 loop
            TMP := TMP xor A(i);
        end loop;

      Y <= TMP;
    end process;
end RTL;
```

図9　三角波のFM送受信

課題

● レベル1：基本課題

図7に示すように，基本課題では，送信データとして1，−1の繰り返しデータを変調したFM波形の復調が正常動作することが最低条件です．回路数を削減したり，高速化などを行ってください．**表3**にFM Receiverの入出力信号ビット幅を示します．これを基本としますが，変更してもかまいません．

図7の上から2段目に対応するFM変調データは，コンテストのホームページ（http://www.cqpub.co.jp/dwm/contest/）からダウンロードできます．各行に属性<8，0，t>の8ビット・データが1,000点分あります．

また，1024ワード×8ビットROMのデータもダウンロード可能です．address＝0〜1023に対応して，1周期分のcosの値すなわち，data＝cos（2*π*address/1024）の8ビットの値が属性<8，0，t>で格納されています．cosの値なので，簡単に生成することもできますが，必要な方はご利用ください．

● レベル2：上級課題

上級課題では，送信データとして，−1〜1の範囲で変化する連続値（三角波形）をFM変調したものを復調できることが最低条件です．**図9**に三角波の送受信のシミュレーション波形を示します．FM復調出力がほぼ三角波になっていることが確認できます．**図9**の上から2段目に対応するFM変調データは，コンテストのホームページからダウンロードできます．

● 社会人部門

社会人部門は，課題をより実践的にするため，FPGAに実装することを前提として設計していただきます．ターゲットFPGAなどの制約条件は，2004年11月10日にコンテストのホームページで発表する予定です．また，本誌2004年12月号〜2005年2月号に掲載する「コンテスト参加要領」にも明記します．かならずご確認ください．

設計する回路は，レベル1（基本課題）でもレベル2（上級課題）でもかまいません．

速度の測定単位

筆者の大学では，論理合成ツールとして米国Synopsys社の「Design Compiler」を使用していますが，このツールはだれもが使えるわけではありません．そこで，使用する論理合成ツールで50入力のXOR回路を合成していただき，その1段当たりの遅延時間を単位時間とし，速度の単位とします．

50入力XOR回路のVHDLソース・コードを**リスト1**に示します（このファイルも本コンテストのホームページからダウンロード可能）．Design Compilerでは6段のXOR回路が合成されます．クリティカル・パス遅延はreport_timingコマンドにより7.17であることがわかります．そこで，7.17/6＝1.195を単位（UNIT）とします．例えば，ある遅延が20ならば，20/1.195＝17.74UNIT遅延ということになります．

ちなみに面積は，report_areaコマンドのtotal cell areaになります．

提出レポートについて

レポートには以下の内容を含めてください．また，ページ数は少なめに，コンパクトにまとめてください．

＜表紙＞

1）代表者の氏名，チーム名，社会人/大学院修士/大学学部生/高専生の区別
2）共同設計者全員の名まえ（最大3名まで），会社/学校名（学籍番号，学年），住所，電話番号，E-mailなどの連絡先
3）取り組んだ課題（レベル1/レベル2）

＜内容＞

1）設計した回路ブロックの構成説明（ブロック図と説明）
2）設計した回路ブロックの動作説明（動作波形図やパイプライン動作などの説明）
3）くふうした点，オリジナリティを出した点（アピールが重要！）
4）クリティカル・パスのスピード，論理合成後の回路規模
5）VHDLもしくはVerilog HDLのコード
6）正常動作しているVHDL/Verilog HDLシミュレーションの波形
7）そのほか自由意見など

レポートはPDFファイルを推奨します．PDFファイルを作成できない場合はご相談ください．

社会人か学生かによって評価の方法が異なります．それにともない，レポートの送付先が異なります．

社会人部門：contest.dwm@cqpub.co.jp
学生部門　：wada@ie.u-ryukyu.ac.jp

締め切りは2005年1月31日（月）必着です．

審査のポイント

速度，回路規模だけでなく，アーキテクチャのユニークさ，アイデア，おもしろさを十分考慮して審査します（ちゃんとアピールしてね！）．

社会人は，Design Wave設計コンテスト審査委員会が審査を行います．学生は，琉球大学で1次審査を行い，最終審査に残ったチームは，2005年3月18日に沖縄産業支援センターで開催予定の発表会に招待され，その会場で最終審査を行います．大学院修士，学部生，高専生のレベルに応じて審査します．

仕様書に従ってまじめに作るのもけっこうですが，おもしろいアイデアを歓迎します．他人と違ったことをしよう！ 仕様の部分変更など，柔軟に受け付けます．

ENJOY HDL!　沖縄で会おう！

＊　＊　＊

本設計コンテストの学生部門は，主催：LSIコンテスト実行委員会，共催：沖縄産業振興センター，協賛：ソニーLSIデザイン，九州経済局により実施されています．

また，今回の課題は，トヨタIT開発センターの横山明久氏に回路案をご提供いただき，作成しました．

わだ・ともひさ
琉球大学 工学部 情報工学科