第3章

cosテーブルの線形補間とローパス・フィルタのくふう

Design Wave設計コンテスト2005 Student部門第1位

チーム スーパー最迅（小川陽太，若林秀明）

Design Wave設計コンテスト2005のStudent部門第1位の設計を紹介する．課題で示されたPLL方式をベースにしている．線形補間を用いることで，比較的回路規模が大きいcosテーブルを小さくした．また，ローパス・フィルタのくふうやレシーバ回路のパイプライン化により，FMレシーバ回路全体の規模を抑え，高速化を図った．（編集部）

わたしたちの所属している研究室では，ハードウェアを設計して数値シミュレーションを高速化する研究などを行っています．研究の中には，HDLで回路を設計し，FPGAに実装して動作させるといったこともあります．

昨年のコンテスト（Design Wave設計コンテスト2004 Student部門/LSIデザインコンテスト in 沖縄 2004）では研究室の先輩が準優勝を収めました．そこで，今年もコンテストに応募して沖縄の発表会を目ざしました．また，コンテストを通してハードウェア設計の技術を高められると考え，課題のディジタルFMレシーバの設計にチャレンジしました．

PLL方式のFMレシーバ回路を設計する

まず，設計仕様書に示されていた復調回路をVHDLで記述することにしました．HDLによるハードウェア設計は初めてだったので，とにかく復調できる回路を完成させることを最初の目標にしました．そして，回路を設計していく過程で何かおもしろいアイデアが浮かんだらそれを取り入れていくことにしました．

● PLL回路の設計

設計仕様書に従って最初に設計したFMレシーバのブロック図を**図1**に示します．このFMレシーバは，入力信号に追随して同期するPLL（phase-locked loop）回路を用いて復調を行っています．回路全体は，大きく分けて，位相比較器，ループ・フィルタ，NCO（numerical controlled oscillator：数値制御発振器），LPF（ローパス・フィルタ）の四つのブロックから構成されています．

FMレシーバ回路の完成後，論理合成とシミュレーショ

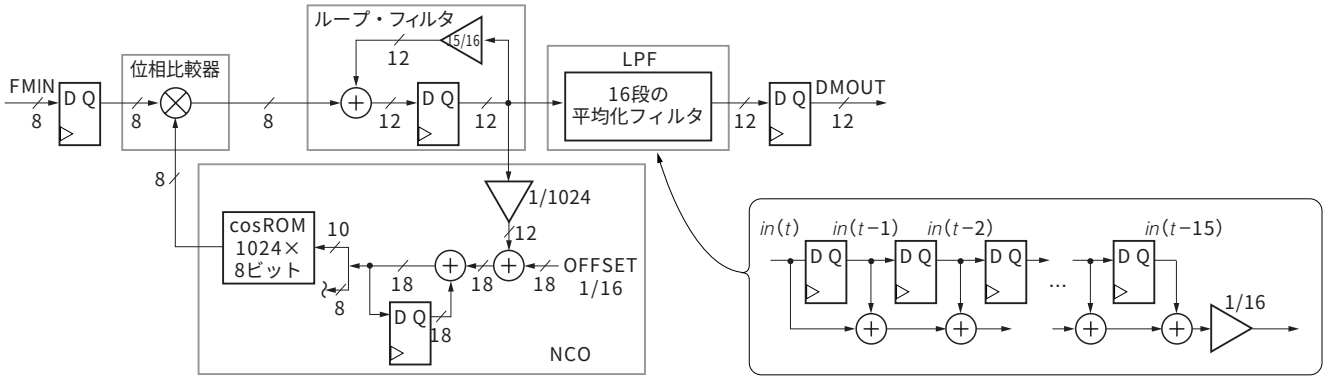

図1　最初に設計したFMレシーバ
設計仕様書に例示されている，PLLを用いたFMレシーバ回路．

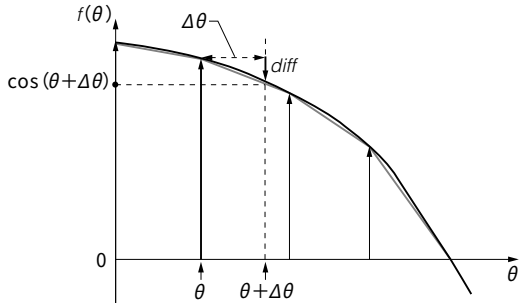

図2　線形補間の方法

cosの曲線をいくつかに分割し，間を直線でつないで近似する．*diff*の値は，直線の傾き×⊿θで計算することができる．

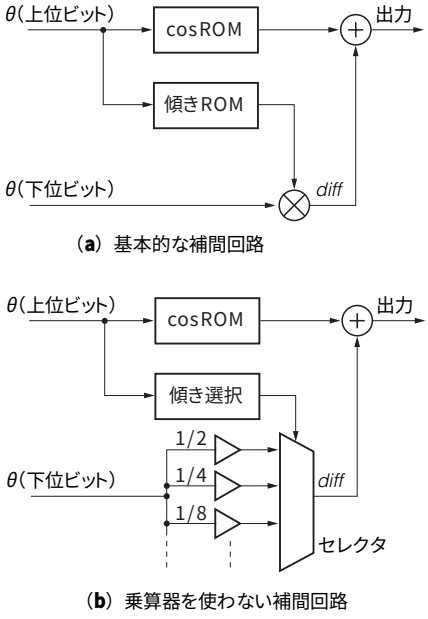

図3　補間回路のくふう

(a)が最初に考えた補間回路．ここで傾きをビット・シフトで計算できる値に限定し，回路を(b)のように変更することで，乗算器を使わないようにすることを考えた．

ンを行い，課題の矩形波と三角波が正常に復調できることを確認しました．

この回路をベースにして，回路規模の縮小と高速化を図ることにしました．今回は，復調方式と基本的な回路構成は設計仕様書のままとし，各ブロックの内部をくふうすることで回路の効率化を行いました．

● cosテーブルの線形補間

NCOの内部には，cosの値を計算するブロックがあります．設計仕様書に示された回路では，cosの計算に1024×8ビットのROMテーブルを使用していました．そしてこのROMはFMレシーバ全体の中で比較的大きな回路規模を占めていました．

本誌2005年1月号のコンテスト関連記事では，テーブルの補間について紹介されていました(1)．そこで，cosROMに格納する値の個数を減らし，テーブルの間を補間でつなぐことによって，cos計算ブロックの回路規模を縮小させることを考えました．

1）回路方式の検討

補間方式として，アルゴリズムが簡単な線形補間を用いました．これは，**図2**のようにテーブル間を直線とみなして近似する方法で，テイラー展開の1次近似，

$$\cos(\theta+\Delta\theta) \quad \cos\theta+(d\cos\theta/d\theta)\,\Delta\theta$$

の考えかたに相当します．**図2**におけるcos(θ＋⊿θ)の値は，cos θの値と，直線の傾きと幅⊿θを掛けた値*diff*とを足し合わせて求めることができます．

このアルゴリズムをもとに最初に設計した回路が**図3(a)**の回路です．cos波形の1周期をいくつかに等分して，とびとびのθに対応するcosの値とそのθ付近のグラフの傾きをROMに格納しておきます．次に，入力θの上位の数ビットを使って，cosの大まかな値をROMから取得します．その後，θの下位の数ビットと直線の傾きを掛け合わせて，テイラー展開の1次項に相当する値*diff*を求めます．これを足し合わせてcosの近似値を求めます．

しかし，**図3(a)**の回路では傾きROMから値を取り出した後で乗算と加算を行うため，動作速度が遅くなりそうです．また，cosROMのほかに傾きROMと乗算器が必要になってしまうので，回路規模がそれほど小さくならないのではないかとも思いました．

乗算器や傾きROMを使わずに回路を構成できないかと考えていたところ，ベースとなるPLL回路の設計において各部で乗算器の代わりにビット・シフトを用いたことを思い出しました．そこで，**図3(b)**のように傾きを簡単に計算できる値に限定して，乗算を行う代わりにセレクタでいくつかの傾きから最適な値を選択する，という方法で補間を行うことにしました．

2）精度と誤差の検討

cosROMの容量や回路の詳細を決定するため，精度と誤差について検討しました．

まず，cos波形の1周期を何等分するかについて考えました．cos波形の1周期分の曲線と，それを64等分して間