QPSK変調を利用した送信機の設計

――信号処理シミュレータとHDLシミュレータの協調シミュレーションで動作を確認

第3章

原田博司，横山明久

ここでは，ソフトウェア無線機の送信機の設計について解説する．変復調にはQPSK（quadrature phase shift keying）を用いる．QPSK送信機を作るために，PN符号生成器やフィルタなどの機能ごとに三つのブロックに分け，HDL化した．また，それぞれのブロックごとの評価やシステム・レベルの評価を行うためのツールの使用方法についても説明する．なお，ここで紹介した設計データは，本誌2005年1月号の付属FPGA基板上で動作させることができる．（編集部）

ここでは，第2章で述べた変復調方式の基礎をもとに，FPGAを用いたソフトウェア無線機を設計します．本稿では，とくにQPSK変調を用いた送信機を実現するためのFPGA設計，および基本的なツールの使用方法などを解説します．なお，ここではシステム・レベルのシミュレーションとHDL（hardware description language）レベルのシミュレーションを共存させるため，以下のツールを使用します．また，HDLの記述には，適当なテキスト・エディタを用いました．

- **システム・レベル・シミュレーション用ツール**
  MATLAB[注1]/Simulink（米国The MathWorks社製）
  System Generator for DSP（米国Xilinx社製）[注2]
- **HDLシミュレーション用ツール**
  ModelSim[注3]（米国Mentor Graphics社製）
- **論理合成，配置配線用ツール**
  ISE（Xilinx社製）[注2]

設計の流れは次のとおりです．

1）変調器の全体の構成を考えて，それをブロック化する
2）それぞれのブロックについて， HDLでソース・コードを記述する
3）そのソース・コードをSimulinkへロードする（System Generator for DSPが有するBlackbox機能を用いる）
4）ソース・コードを評価するためのモジュールをSimulink上で用意する
5）SimulinkとModelSimを協調させてシミュレーション（コシミュレーション）を行う
6）必要であれば，ほかのソース・コードを追加して，3）

注1：Version R13で動作を確認した．
注2：System Generator for DSPとISE Foundationの評価版は，本誌付属のDVD-ROMに収録されている．なお，筆者らは今回，論理合成，配置配線用ツールとしてISE WebPACK 6.3iを使用した．
注3：Version 5.8bで動作を確認した．なお，ModelSim XEは，本誌付属のDVD-ROMに収録されているISE Foundationの評価版に含まれている．

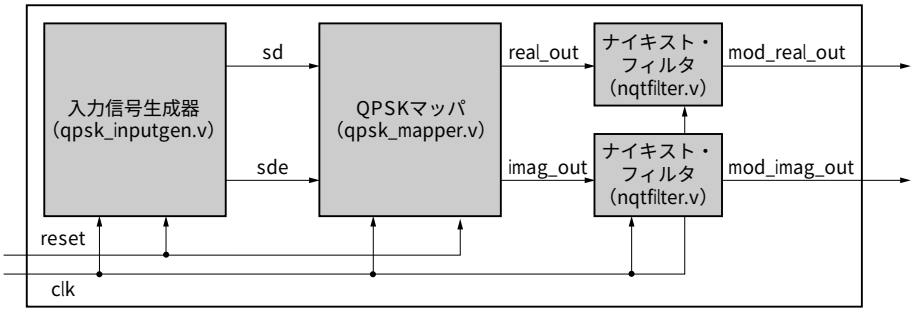

図1
設計する送信機の構成図
本特集で設計するQPSK変調処理システムの内部構成を示している．PN系列を生成する入力信号生成器，IQ信号を生成するQPSKマッパ，帯域制限処理を行うナイキスト・フィルタから構成される．

からの動作を繰り返す

7）すべて終了した後に，FPGAに対する論理合成，配置配線用のプロジェクト・ファイルを作成する

1 全体の構成を考えてブロック化する

では，QPSK変調を利用した送信機を設計していきます．

● 三つのブロックでシステムを構成する

まず，全体のブロック構成ですが，大きく分けると次の3種類のブロックからなります（**図1**）．

1）QPSK_INPUTGENブロック

2）QPSK_MAPPERブロック

3）NQTFILTERブロック

1）によって，QPSKの変調器に対する入力信号を生成します．ここでは疑似ランダム系列であるPN（pseudo noise）符号からなるランダム・データを作ることにします．入力はクロックclkとリセット信号reset，出力はランダム・データsdとsdデータに同期したイネーブル信号sdeです．

2）は，入力された信号sdとイネーブル信号sdeを受け，それらを**図2**に示すQPSK用の信号点に配置します．今回は2ビットの信号aとbを受信し，出力のデータreal_out（*I*チャネル）とimag_out（*Q*チャネル）を出力します．この信号は，後でフィルタをかけることを考慮して4クロックごとにデータを出力し，それ以外は‘0’データを出力します（詳細は本特集 第2章を参照）．

3）は第2章の**図9**（p.49）で説明したナイキスト・フィルタのブロックです．入力されるreal_out，imag_out，およびクロック信号clkを受け取り，**図3**に示すフィルタをかけます．これは，タップ数が17でロールオフ率αが1のナイキスト・フィルタです．このブロックから，フィルタ処理を行った変調信号mod_real_outとmod_imag_outが出力されます．

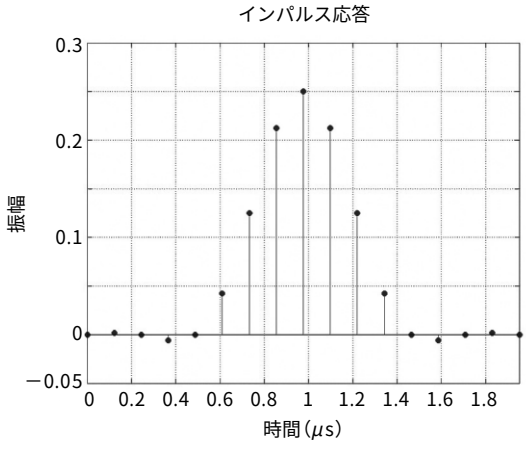

図3　ナイキスト型フィルタ

帯域制限を行うために利用するロールオフ率100%のナイキスト・フィルタのタップ係数値とインパルス応答値を示す．信号点間で影響を及ぼさないように，中心から4点離れた場所では係数が0となっていることがわかる．

2 ソース・コードを作成する

次に，各ブロックのソース・コードを作成します．

● QPSK_INPUTGENブロックでPN符号生成器を実現

まず，QPSK_INPUTGENブロックです．今回はHDLとしてVerilog HDLを用います．

リスト1は，**図4**に示すPN符号からなるランダム・データを生成するブロックのプログラムです．このPN符号生成器は，0～6の番号が割り振られた七つのシフト・レジスタで構成されます．初期値として“1000000”が入力されていますが，クロックの入力に合わせて6番目のシフト・レジスタに入っている値が出力されます．このとき，6番目

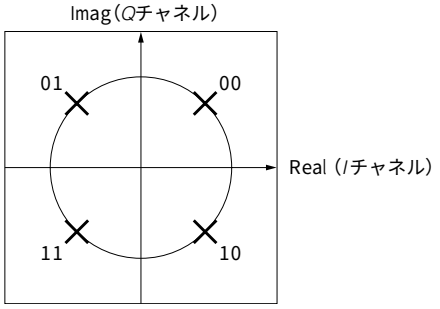

図2　QPSK変調における信号空間ダイヤグラム

QPSK変調において1信号点は2ビット入力信号により決定される．図に示すように，信号点の*x*軸側を*I*チャネル（Real），*y*軸側を*Q*チャネル（Imag）と呼ぶ．

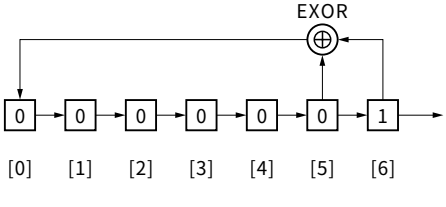

図4　PN符号器

7ビット・シフト・レジスタを利用して，[0]ビットについては[5]と[6]の排他的論理和を入れ，そのほかのビットについてはそれぞれ1ビットずつシフトさせる．出力は[6]ビットから取得する．初期化時に[6]ビットに‘1’を，そのほかのビットには‘0’を格納しておく．