基礎から学ぶ Verilog HDL & FPGA 設計

第1回
全加算器をHDLで設計してみよう

中野浩嗣，伊藤靖朗

デバイスの記事　ビギナーズ　関連データ

Verilog HDLによるFPGA（field programmable gate array）設計を基礎から学ぶための連載記事である．今回は，最も単純な組み合わせ回路の一つである全加算器を設計する．

（編集部）

この連載では，シミュレーションやFPGAボードによる動作を体験しながら，Verilog HDLによるFPGA設計手法を学びます．具体的には，さまざまな簡単な回路をVerilog HDLで設計し，それらを組み合わせることによって小型CPUを実現します．最終的には，C言語のような高級言語で記述したプログラムを，設計した小型CPU上で動作させます．

設計した回路データは，実際にFPGAボードにダウンロードして動作を確認します（**写真1**）．ここでは米国Xilinx社の「Spartan-3Eスタータ・キット」を利用します．このスタータ・キットには，有効ゲート規模が約50万ゲートの「XC3S500E」が搭載されています．

FPGAベンダが提供している無料の設計ツールを用いて，シミュレーションによる動作確認も行うので，FPGAボードを持っていなくても学習には差し支えありません．ただし，FPGAボードで動作確認した方が，FPGA設計の楽しさをより実感できるでしょう．

● FPGAをVerilog HDLで設計する

Verilog HDLは，HDL（hardware description language；ハードウェア記述言語）と呼ばれる，ハードウェアを設計するための言語の一種です．ゲート・レベル（フリップフロップやANDゲートなどの論理回路の接続の記述）から，ビヘイビア・レベル（ハードウェアの動作やアルゴリズムの記述）までの，さまざまな抽象度で回路を設計することができます．記述方法はC言語のそれと似ており，回路図で設計するのが困難だった大規模な回路を，設計しやすくなります．

Verilog HDLと並んで有名なHDLとして，VHDLがあります．VHDLもVerilog HDLと同様に，さまざまな抽象度で回路を設計できます．VHDLはVerilog HDLと比べて，文法が厳格で記述量が多くなりやすい特徴があります．つまり，同じ動作をする回路をVHDLとVerilog HDLで設計すると，多くの場合はVerilog HDLの方が簡潔に記

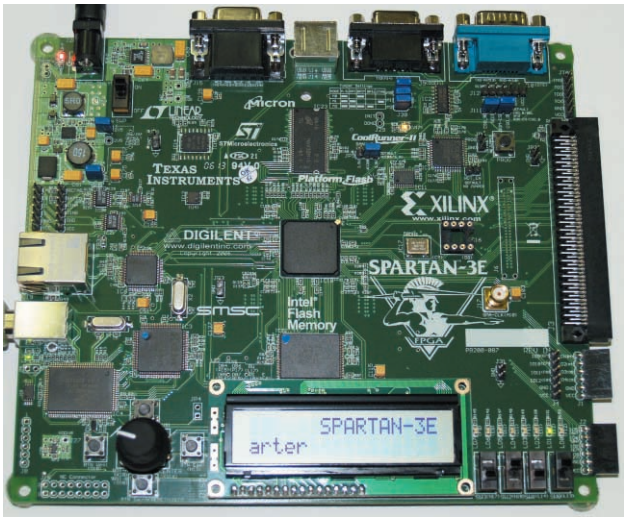


写真1　Spartan-3Eスタータ・キット

Spartan-3Eスタータ・キットのボードには，50万ゲート相当のFPGA「XC3S500E-4FG320C」が搭載されている．各種メモリやインターフェースも充実しており，FPGA設計の入門者に適したキットである．

KeyWord　Verilog HDL，FPGA，HDL，全加算器，モジュール，ポート，シミュレーション，テストベンチ

述できます．しかし，文法が厳格な方が設計ミスが起こりにくいとも言えるので，Verilog HDLとVHDLのどちらが優れていると決めることはできません．本連載では，記述量が少なくて済み，初心者にとっつきやすいことから，Verilog HDLを選択しました．

● FPGA設計の基本的な流れ

Verilog HDLによるFPGA設計の基本的な流れを以下に示します(図1)．

1) 最初に，Verilog HDLを用いて回路を作成します(これを「デザイン入力」と呼ぶ)．
2) 次に，作成した回路が正しく動作するかをシミュレーションによって確認します．意図したように動作しない場合は，デザイン入力に戻って回路を修正し，再びシミュレーションを行って，正しく動作するまで繰り返します．
3) シミュレーションで正しく動作することが確認できたら，Verilog HDLで記述された回路記述を，ネット・リスト(基本的な回路から構成された回路記述)に変換します．これを「論理合成」と呼びます．
4) 次に，使用するFPGAに合わせて，作成した回路の配置配線を行います．
5) 配置配線が完了したら，ビット・ファイル(FPGAにダウンロードする回路データ)を作成します．最後に，作成したビット・ファイルをFPGAにダウンロードします．

実は，以上の作業は，FPGAベンダの提供するFPGA開発ツールを用いて行うことができます(特に，米国Xilinx社の「ISE WebPACK」，米国Altera社の「Quartus II Web Edition」などは無償で提供されている)．これらのツールは，Verilog HDL記述を基にして自動的に論理合成と配置配線を行い，ビット・ファイルを生成します．そのため，ユーザがネット・リストを見る必要はありません．

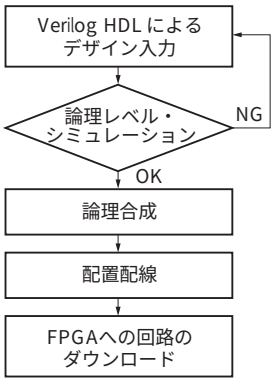


図1
Verilog HDLによる回路設計の流れ
HDLで回路を記述し，シミュレーションで動作を確認してから論理合成を行う．その後，使用するFPGAに合わせて配置配線を行い，FPGAにダウンロードするビット・ファイルを作成する．

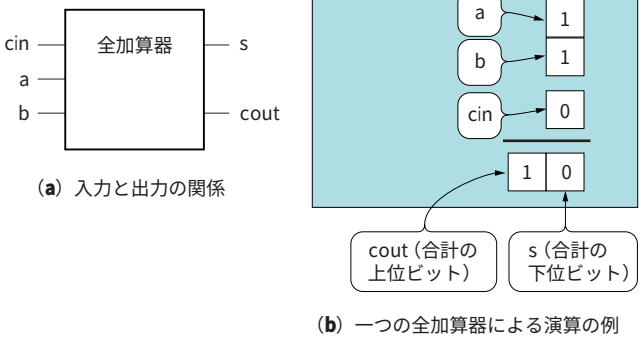
1 全加算器を設計してみる

それでは，実際にVerilog HDLを用いて回路を記述してみましょう．今回は全加算器(full adder)を取り上げます(図2)．

● まずは入出力の確認から

全加算器の入力はa，b，cinの3ビットで，出力はsとcoutの2ビットです．入力3ビットの和は，2ビットの2進数で表すことができます(“00”，“01”，“10”，“11”の4通り)．全加算器は，その上位ビットをcoutに，下位ビットをsに出力します．全加算器の入出力の対応を表した真理値表を表1に示します．

3ビットの入力a，b，cinのうち，奇数個が‘1’である場合に，出力sは‘1’となります．また，coutが‘1’となるのは，これらの3ビットのうち2ビット以上が‘1’の場合です．よって，論理式で書くと，



(a) 入力と出力の関係

(b) 一つの全加算器による演算の例

図2 全加算器
二つの値(a，b)を加算し，けた上がり入力(cin；carry in)を含めて演算する．結果として，出力値(s；sum)とけた上がり出力(cout；carry out)を出力する．全加算器を複数個用いることにより，複数ビットの加算に対応した加算器を作ることができるが，この設計方法については次回に詳しく説明する．

表1
全加算器の真理値表

入力			出力	
a	b	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1