基礎から学ぶ Verilog HDL & FPGA 設計

第3回
マルチプレクサと算術論理演算回路

中野浩嗣，伊藤靖朗

デバイスの記事　ビギナーズ

本連載は，さまざまな回路をVerilog HDLで設計していき，最終的には小型のCPUを実現することをねらいとしている．今回は，CPUの主要な構成部品である，算術論理演算回路を設計する．また，そのための準備運動として，3入力のマルチプレクサを設計する．（編集部）

算術論理演算回路（ALU：arithmetic logical unit）は，算術演算と論理演算を実行する回路であり，CPUを構成する主要な部品の一つです．今回は，この算術論理演算回路を設計してみたいと思います．

まず，算術論理演算回路を設計するための準備運動として，3入力のマルチプレクサを設計します．

● マルチプレクサの設計

3入力マルチプレクサは，1ビットの入力ポートa，b，cと2ビットの入力ポートf，1ビットの出力ポートsを持ちます（**図1**）．入力fは選択入力として機能します．すなわち，fが“00”のとき，出力ポートsからは，aに入力されている値が出力されます．同様に，“01”，“10”のときは，sからはそれぞれb，cの値が出力されます．また，fには“11”が入力されることはなく，もし入力されたとしても，sから出力される値は何でもよいものとします．

マルチプレクサをVerilog HDLで記述したものを**リスト1**に示します．8行目から始まるalways文で3入力マルチプレクサの動作を定めています．ここでは，a，b，c，fのいずれかの値が変化するたびに，後に続くcase～endcase間の文が実行されます．

always文の直後のcase文は，引き数（ここではf）の値によって動作を決定します．このcase文は，C言語のswitch文とよく似た機能を持っています．

10行目の2'b00は，2進表現で“00”の値を持つ2ビットの数値です（数値表現については，p.130のコラム「Verilog

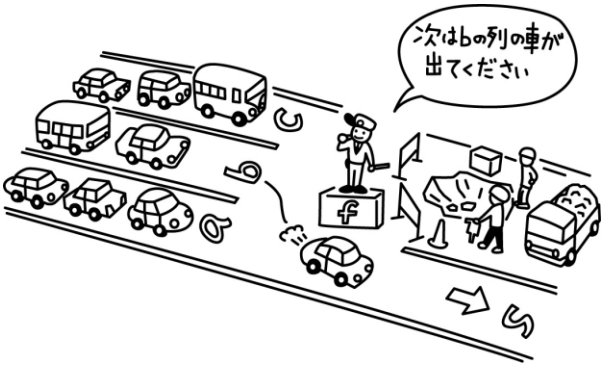

図1　マルチプレクサのイメージ

リスト1　マルチプレクサのVerilog HDL記述（multiplexer.v）

```
 1  module multiplexer(a, b, c, f, s);
 2
 3    input a, b, c;
 4    input [1:0] f;
 5    output s;
 6    reg s;
 7
 8    always @ (a or b or c or f)
 9      case (f)
10        2'b00: s = a;
11        2'b01: s = b;
12        2'b10: s = c;
13        default: s = 1'bx;
14      endcase
15
16 endmodule
```

fの値によってsに代入される値が変わる

KeyWord　Verilog HDL，FPGA，算術論理演算回路，ALU，マルチプレクサ，数値表現，不定値，非同期ラッチ

HDLの数値表現」を参照）．ここでは，fの値が“00”のときに，レジスタ型変数sにaの値を代入することを表しています．11行目と12行目は，それぞれfが“01”と“10”の場合の動作を定義しています．

13行目は，fがこれらの値以外の場合の動作を定めています．ここでは，sに1'bxが代入されていますが，このxは不定値である（sに書き込まれる値はどのようなものでもかまわない）ことを意味しています．このように定義しておくと，設計ツールはfが“11”のときのsの値を考慮しなくてすむので，よりコンパクトで高速な組み合わせ回路を生成することが期待できます．

ここでは，13行目のdefault文は「省略可能ではないか？」と思われるかもしれません．しかし，省略すると，fの値が“11”のときにはsの値は変更されないことになります．つまり，sの値は「現在の値を保持し続ける」ことになり，設計ツールはそのために非同期ラッチを生成してしまいます．今回のように，always文中にcase文を用いて組み合わせ回路を設計する場合は，その最後に必ずdefaultを含むようにしましょう（非同期ラッチの生成については，p.132のコラム「always文による非同期ラッチの生成」を参照）．

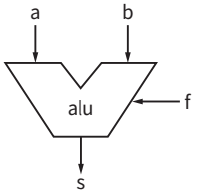

図2
算術論理演算回路aluのブロック図
aとbを入力し，sを出力する．演算機能はfで選択する．

● 算術論理演算回路の設計

マルチプレクサの考え方に基づいて，算術論理演算回路を設計します．ここでは，二つの16ビットの入力ポート（aとb），機能選択を行うための5ビットの入力ポート（f），16ビットの出力ポート（s）を持つ算術論理演算回路を考えます（**図2**）．

算術論理演算回路の機能は自由に決めることができますが，ここでは以下のように定めます（**表1**）．機能選択ポートfのビット数を増やすことにより，より多くの演算をサポートすることも可能です．

なお，この算術論理演算回路がCPUを構成する部品となることを考慮し，入力ポートaとb，出力ポートsのビット列が数値を表す場合，そのビット列は2の補数表現であるとして演算を行うことにします[注1]．

1）算術演算

加算（ADDition），減算（SUBtraction），乗算（MULtiplication），符号反転（NEGation）をサポートします．

2）ビット・シフト演算

左ビット・シフト（SHift Left）と右ビット・シフト（SHift Right）をサポートします．**表1**に示した式は，bのビット列をaの値の分だけシフトすることを表しています．

注1：補数表現については，前回の記事（本誌2007年6月号のpp.130-131）を参照のこと．

コラム1

Verilog HDLの数値表現

Verilog HDLの数値表現は，基数を指定することによって，2進数表現や16進数表現を指定することができます．指定しなければ10進数表現とみなされます．基数の指定は，bまたはBで2進数(binary)を，oまたはOで8進数（octal）を，dまたはDで10進数（decimal）を，hまたはHで16進数（hexadecimal）を表します．また，ビット数も指定できます．指定しなければ32ビットとして扱われます．

基数とビット数を指定する場合の形式は，

［ビット数］'［基数］［数］

です．**表A**に，各数値表現と対応するビット列の具体例を示します．

なお，Verilog HDLでは，1ビットの値は‘0’，‘1’，‘z’（ハイ・インピーダンス），‘x’（不定値）の4通りの値をとることができます．ハイ・インピーダンスについては次回以降で詳しく説明しますが，直感的には，値がない状況を表すのに用います．

不定値は，論理設計でのドントケア（don't care）を表現するのに用いられます．また，回路シミュレーションでは，一つの信号線に複数の値が同時に書き込まれた場合や値が一つも書き込まれないときなどに，その値は不定値となります．

表A　Verilog HDLの数値表現とビット列の具体例

数値表現	ビット列
15	00000000 00000000 00000000 00001111
−3	11111111 11111111 11111111 11111101
7'b110	0000110
b110	00000000 00000000 00000000 00000110
16'hfffe	11111111 11111110
4'hz	zzzz
4'hx	xxxx