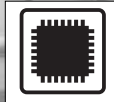


基礎から学ぶ Verilog HDL & FPGA 設計

第9回

バスと命令フェッチ回路

中野浩嗣, 伊藤靖朗



デバイスの記事



ヒギナース

今回は、CPUの重要な構成部品であるバスを設計する。このバスと、これまでの連載で設計したカウンタ、ステート・マシン、メモリ(ブロックRAM ; Block RAM)を構成部品として用いて命令フェッチ回路を設計し、シミュレーションとFPGAボードを用いた動作確認を行う。(筆者)

● バスとは

CPUでは、内部でデータの転送を行うのにバスを用います。バスを用いることにより信号線の数を削減することができます。設計も容易になります。

図1(a)に示すように、三つのモジュールA, B, Cから二つのモジュールDとEへデータ転送を行う場合を考えてみます。ここで、複数の組み合わせ転送が同時に起こることはなく、A, B, Cのうち一つからDまたはEへのデータ転送が行われるものとします。このとき、図1(a)のような1対1の配線によるデータ転送だと $3 \times 2 = 6$ 通りの組み合わせの配線が必要となります。

バスを用いると配線を節約できます。図1(b)のように、

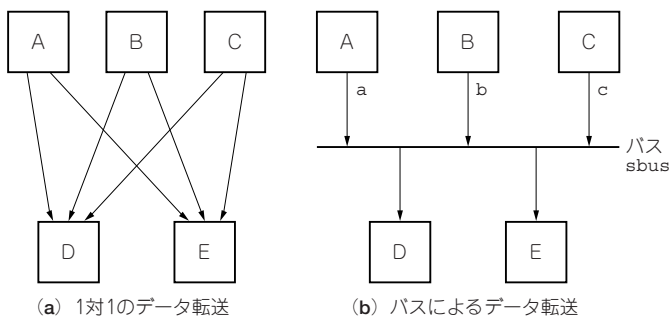


図1 1対1のデータ転送とバスを用いたデータ転送

A, B, Cの出力がバスに接続されており、これら三つのうち一つがバスにデータを書き込みます。DとEはバスからデータを読み出すことにより、データ転送が実現できます。このようにバスを用いると、各モジュールとバスの間の配線だけで済みます。

● バスの設計

バスをVerilog HDLで設計する場合、どのデータをバスへ書き込むかを制御する必要があります。ここでは、マルチプレクサを用いる方法と、3ステート・バスを用いる方法の二つの記述法を紹介します。

具体例として、図1(b)の三つのモジュールA, B, Cの出力を16ビットのa, b, cとします。そして、これらのうち一つの値がバスsbusに書き込まれる状況を例に説明します。1ビットの制御線ase1, bse1, cse1を用いて、ase1が'1'のときは、aの値をsbusに書き込み、bse1が'1'のときはbの値をsbusに書き込み、cse1が'1'のときはcの値をsbusに書き込むものとします。ase1, bse1, cse1のうち二つ以上が同時に'1'になることはなく、またすべて'0'の場合は、sbusの値はどのようなものでも構わないものとします。

● マルチプレクサを用いる方法

リスト1はマルチプレクサを用いたバスのVerilog HDL記述です。連載第3回目(2007年8月号, pp.129-133)のマルチプレクサの記述と本質的に同じです。always文中の代入文でsbusの値を決定するので、sbusはreg文で宣

Keyword

バス, 命令フェッチ回路, マルチプレクサ, 3ステート・バス, ハイ・インピーダンス, プログラム・カウンタ, 命令レジスタ, ステート・マシン, メモリ, 出力バッファ, 連結演算子

言したレジスタ型変数とします。7行目のelse文で、sbusに不定値16'hxxxxを書き込んでいます。連載第3回目で述べた通り、7行目を削除すると、asel, bsel, cselがすべて'0'の場合はsbusに値が代入されないの、意図しないラッチを生成してしまうことに注意してください。

● 3ステート・バスを用いた記述法

リスト2は3ステート・バスによるバスの記述法です。wire文で、16ビットのネット型変数sbusを宣言します。3~5行目のassign文で、sbusに値を書き込んでいます。各assign文の右辺は条件演算子が用いられています。

条件演算子は、C言語のものと同じです。例えば3行目の場合、aselが'1'(真)のとき、sbusにaが書き込まれ、'0'(偽)のときは、16'hzzzzが書き込まれます。このzはハイ・インピーダンスを表します。ハイ・インピーダンスzを書き込むということは、直感的には、「値を書き込まない」ということを意味します。

sbusには一つしか値を書き込めないの、asel, bsel, cselのうち、'1'になるのは、一つでなければなりません。一つだけが'1'の場合、sbusの値は、それに対応するa, b, cのいずれかの値になります。三つとも'0'の場合、sbusの値はハイ・インピーダンス(16'hzzzz)になります。もし、二つ以上が'1'になってしまった場合、シミュレーション上はsbusの値は不定値16'hxxxxになり、複数の16'hzzzzでない値が書き込まれたことを知ることができます。

リスト1 マルチプレクサによるバスのVerilog HDL記述

```

1  reg [15:0] sbus;
2
3  always @(a or b or c or asel or bsel or csel)
4  if (asel) sbus = a;
5  else if (bsel) sbus = b;
6  else if (csel) sbus = c;
7  else sbus = 16'hxxxx;

```

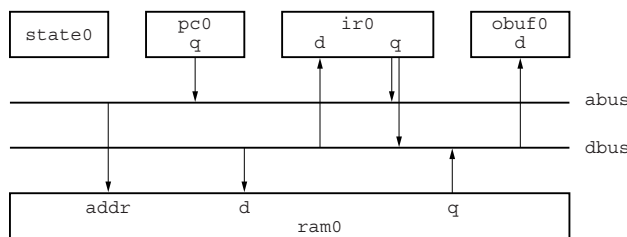


図2 命令フェッチ回路の概略

● 命令フェッチ回路の仕様

バスの例として命令フェッチ回路を作ってみましょう。この命令フェッチ回路では、これまでに設計したカウンタ(counter.v)、ステート・マシン(state.v)、およびメモリ(ブロックRAM, ram.v)を用います。カウンタをインスタンス化して、プログラム・カウンタ(pc0)、命令レジスタ(ir0)、および、出力バッファ(obuf0)を作ります。

命令レジスタと出力バッファは16ビット、プログラム・カウンタは8ビットとします。ステート・マシンとブロックRAMは、そのままstate0とram0としてインスタンス化します。

メモリは、アドレス(addr)の幅が8ビット、データ(dとq)の幅が16ビットとします。これら五つのモジュールのデータ転送を8ビットのアドレス・バス(abus)と16ビットのデータ・バス(dbus)を用いて行います。

アドレスに関する値の転送はアドレス・バスを用い、そのほかのデータについてはデータ・バスを用いることにします。図2は命令フェッチ回路の概略です。アドレス・バスがメモリのアドレス入力(addr)に直結し、データ・バスがデータ入力(d)に直結しています。

命令フェッチ回路では、メモリに格納されているデータ(機械語命令)を0番地から順に読み出し、16ビットの命令レジスタ(ir0)に書き込みます。現在の番地は、プログラム・カウンタ(pc0)に格納されており、読み出すたびに、プログラム・カウンタの値はインクリメントされます。これらの動作は、状態FETCHAとFETCHBで行います。

命令レジスタに書き込むだけでは単純すぎるので、この値をもとに、メモリの読み書きを行うCPUの疑似動作を行うことにします。命令レジスタの値が'0'のとき、動作を停止する、つまり状態IDLEに遷移します。命令レジスタの下位8ビットが'0'で、上位8ビットの値が'0'でない場合、上位の8ビットを番地としてメモリからデータを読み出し、その値を出力バッファ(obuf0)に書き込みます。命令レジスタの下位8ビットが'0'でないとき、上位8ビットを番地、下位8ビットをデータとしてブロックRAMに

リスト2 3ステート・バスによるバスのVerilog HDL記述

```

1  wire [15:0] sbus;
2
3  assign sbus = (asel ? a : 16'hzzzz);
4  assign sbus = (bsel ? b : 16'hzzzz);
5  assign sbus = (csel ? c : 16'hzzzz);

```