連載 Verilog-HDLによるコンピュータ設計

第1回 アーキテクチャを設計する

渡辺 玲

今回より，1年間の予定で，ハードウェア記述言語を使ったCPUの設計を行っていきます．

第1回目の今回は，命令セット・アーキテクチャの仕様を暫定的に定めます．2回目以降では，ここで決めた仕様にしたがい，Verilog-HDL言語を使って，アーキテクチャ・レベルでコンピュータを設計します．最終的には，簡単な試験プログラムで動作を確認するまでを解説していきます．

連載のはじめに

●コンピュータ設計の開発ストーリ

これからVerilog-HDLを駆使してコンピュータを設計していきます．

Verilog-HDLやVHDLの解説記事やマニュアル本は，今までにも何種類か世の中に出ています．いくつか読まれた読者の方も多いと思います．

そこで今回は，そういったVerilog-HDLの機能を一から十まで逐一解説するマニュアル的な構成から，思い切って離れてみようと考えました．

この連載では，コンピュータという，比較的大きな規模のシステムを作る過程（開発ストーリ）をとおして，Verilog-HDLのいろいろな側面に違った角度から光を当てていきます．そして，単にVerilog-HDLの長所ばかりでなく，欠点にも触れることができればと思います．

●連載の目的

この連載は，筆者がCQ出版発行の月間誌，『Interface』で昨年1年間連載した，「C++による仮想コンピュータの実現」の姉妹編に当たります．しかし，連載の目的はまったく別です．

▼コンピュータの動作解説はC++で

「C++による…」では，コンピュータのハードウェアの中身をブラック・ボックスとしてとらえがちなソフトウェアやシステムのエンジニアたちに，ハードウェアの仕組みを解説するために企画された連載です．

ハードウェア記述言語ではなく，彼らにもっとなじみのあるC++言語を使って，コンピュータのハードウェアを記述しました．そして，作ったコンピュータのモデルをシミュレーションで走らせることで，CPUの動作についてより理解を深める方法をとりました．

このように，常にハードウェアの解説がテーマの中心でした．

▼テーマはHDL設計にあり

しかし，この連載では，実際にチップを作っているような読者の方々を対象とします．したがって，コンピュータの動作のような基本的なハードウェア動作の解説は省くことにします．

実際のところ，連載１回あたりのページ数が限られていますので，多くのページをVerilog-HDL関連の話に割くため，ハードウェアの解説は最小限にとどめる予定です．

表1 Interface誌連載「C++による仮想コンピュータの実現」の内容

1996年	
1/2月号	コンピュータの部品ー論理素子
3月号	計算をするー加算器と減算器
4月号	算術論理ユニット（ALU）を組み立てる
5月号	プロセッサ＋メモリ＝コンピュータ
6月号	比較，条件分岐，ロード／ストア命令の追加
7月号	構造レベルで記述した仮想コンピュータ
9月号	パイプライン・プロセッサを作る
10月号	パイプラインの流れが止まる時
11月号	BIUの製作
12月号	キャッシュによる性能向上
1997年	
2月号	仮想記憶がプログラミングを変える
3月号	オブジェクト指向言語によるハードウェアの記述

表2 本連載の計画

第1回	アーキテクチャの設計
第2回	マイクロ・アーキテクチャの設計ーパイプライン・プロセッサ
第3回	バス・インターフェース・ユニット（BIU）とメモリ・システムの設計
第4回	キャッシュと仮想記憶の設計
第5回	ALUの設計
第6回	論理ゲートの設計

●理解に行き詰まったら

もし，この連載を読んでいて，ハードウェアの話で行き詰まったり，よくわからなくなったときには，1996年1/2月合併号から1997年3月号までInterface誌で連載された「C++による…」を参照してください．コンピュータのハードウェアについてのやさしく詳しい解説が載っています（**表1**）．

Interface誌の連載とこの連載とは，話を進める順序が多少違います．「C++による…」はボトムアップ的な解説でしたが，今回はトップダウン的になります（**表**

2)．しかし目的となる対象がコンピュータという点で共通していますから，内容的にはオーバラップすることも多くあります．どの回を見ればよいかは，すぐにわかるはずです．

ちなみに，今回の連載の内容は「C++による仮想コンピュータの実現」では，第4回(1996年5月号)と第5回(1996年6月号)に相当します．

暫定的に命令セットを決める

今回はコンピュータをゼロから作りますので，まず，アーキテクチャの仕様を決めます．

コンピュータでアーキテクチャといえば，それは命令セットのことだと考えて，まずまちがいありません．RISCとかCISCとかいったことも命令セット(インストラクション)に関する分類です．

アーキテクチャは一般的な概念ですが，「命令セット・アーキテクチャ」という言葉は，特にコンピュータのために作られた言葉です．

●命令の種類を決める

今回は，革新的なコンピュータを作るのが目的ではありませんので，肩の力を抜いて，一般的に必要な命令を考えてみましょう．

▼算術・論理演算命令

まず，算術・論理の演算命令として，AND命令，OR命令，NOT命令，XOR命令，SHIFT-LEFT命令，SHIFT-RIGHT命令，ADD命令，SUB命令を考えます．

▼分岐命令

また条件分岐命令として，BE命令(等しい)，BNE命令(等しくない)，BL命令(小さい)，BG命令(大きい)，BLE命令(以下)，BGE命令(以上)の命令を考えます．さらに分岐条件を設定するためにCOMP命令(比較)を加えます．

▼メモリ-データ間の転送

メモリとデータのやりとりをするためにLOAD命令と，STORE命令も必要です．また，即値(Immediate)を使うためにSET命令も加えます．

▼システム関連

システム関係としては，チップがパワー・ダウン・モードに入るPOWER-DOWN命令を組み込んでみました．

これ以外の未定義の命令はすべてNOP命令と考え，何もしないことにします．

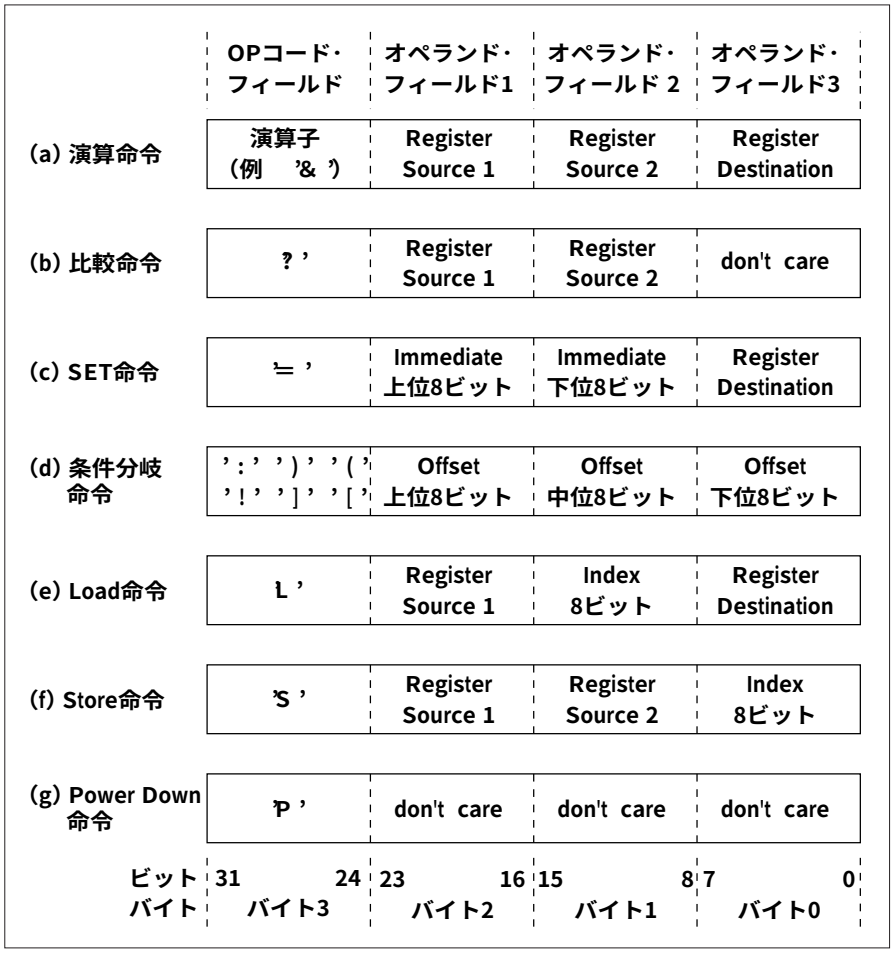

図1　命令フォーマット

●命令フォーマットを決める

命令の種類と並んで重要なのは，命令のフォーマットです．

▼固定32ビット長を選択

まず，可変語長にするか，固定語長にするかを決めます．

今回は実装が楽な固定語長にします．語長は32ビットです．この32ビットをいくつかの小さいフィールドに分割します．そのやり方もいろいろありますが，今回は単純に上位1バイトをOPコードとし，下位3バイトをオペランドとします．**図1**に詳細を示します．

▼機械語は1語で記述する

またOPコードの定義ですが，機械語を書きやすくするために，LOAD命令なら‘L’，ADD命令なら‘+’，AND命令なら‘&’とASCII文字から意味の近い1文字を選んで割り付けました．

例えば，「レジスタ3＝レジスタ1＋レジスタ2」は，一般的なアセンブラでは，

```
ADD r1, r2, r3
```

となりますが，ここで設計するコンピュータの機械語では

```
'+' 1 2 3
```

すなわち，

```
0x2b010203
```

になります('+'はASCIIコードで0x2b