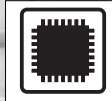


基礎から学ぶ Verilog HDL & FPGA 設計

第13回

アセンブラの設計

中野浩嗣, 伊藤靖朗



デバイスの記事



ビギナーズ

前回(2008年12月号 pp.88-92)は、16進数で書かれた機械語プログラムによりメモリを初期化し、学習用CPU [TINYCPU]上で機械語プログラムを動作させた。今回は TINYCPU をターゲットとするアセンブラ [TINYASM] を、Perl 言語を用いて設計する。このアセンブラはニーモニックとラベルで書かれたアセンブリ言語プログラムを16進数の機械語命令の列に変換する。TINYASM から出力された機械語プログラムを Verilog HDL のメモリ初期化部分に変換することで、FPGA に実装できる。(筆者)

● アセンブラを作ってプログラミングを容易にしよう

図1は、コンパイラとアセンブラを用いて学習用CPU [TINYCPU]向けプログラムを生成する処理の、全体の流れを表しています。

一般に機械語プログラムを見ても、どのような処理を行っているのかを解読するのは困難です。しかしアセンブリ言語プログラムは、各命令がどのような動作をするかを見ていけば、処理の手順を分析できます。また、機械語で直接プログラミングを行うのはほとんど不可能ですが、アセンブリ言語によるプログラミングは比較的容易です。

そこで本稿では、アセンブリ言語プログラムを機械語プログラムに変換する TINYCPU 向けアセンブラ [TINYASM] を設計します。

アセンブラから出力された機械語プログラムは、そのまま Verilog HDL のメモリの初期化部分に変換します。この初期化部分を、前回と同様に、メモリ (ram.v) の initial 文に挿入することで、メモリの初期値とすることができます。そして、ビット・ファイルを生成し、FPGA にダウン

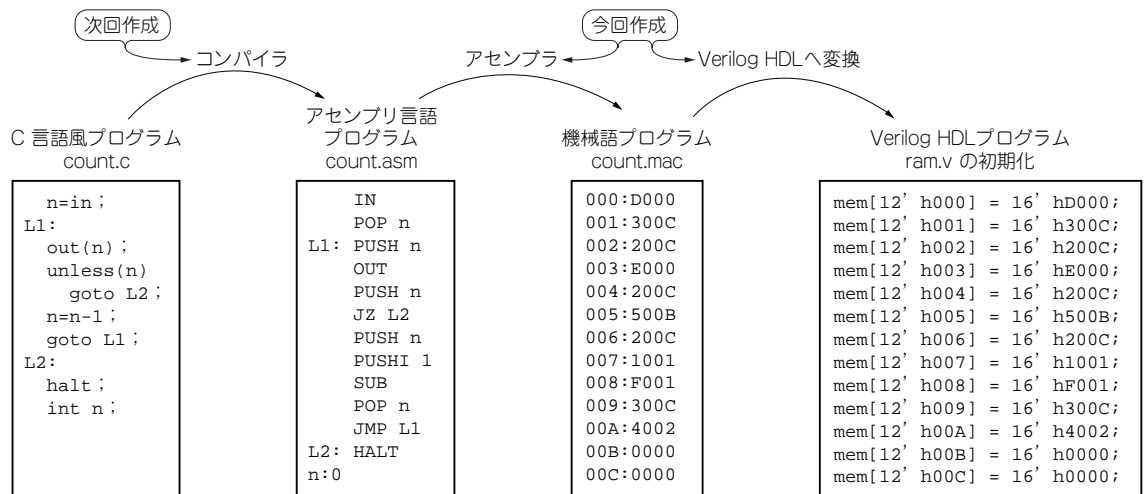


図1
コンパイラとアセンブラ

Keyword

アセンブラ, アセンブリ言語プログラム, 機械語プログラム, コンパイラ, Cygwin, Perl, Flex, Bison, gcc, ラベル, ニーモニック, オペランド, ラベル・リスト

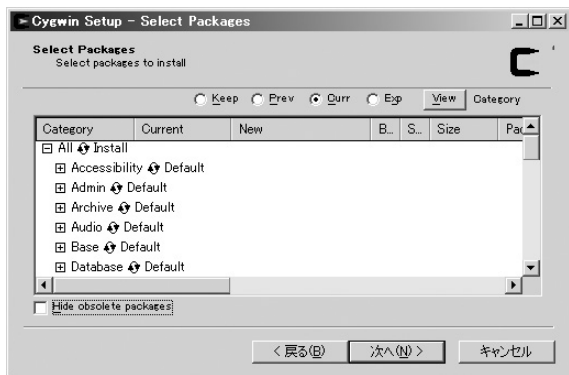


図2 Cygwinのパッケージ選択画面

Allを選択する。

ロードすることにより、TINYCPUは機械語プログラムを実行できます。

コンパイラはC言語風に記述されたプログラムをニーモニック(命令)とラベルからなるアセンブリ言語プログラムに変換します。コンパイラの設計は次回紹介する予定です。

● アセンブラ TINYASM の開発環境として Cygwin を利用

アセンブラ TINYASM の設計は Perl 言語で行います。コンパイラの設計では、字句解析ツール「Flex」、構文解析ツール「Bison」、C 言語コンパイラ「gcc」を用います。TINYCPU 向けのプログラミング環境を構築するためには、合わせて四つのソフトウェアが必要です。

Linux などの UNIX 環境では、これら四つのソフトウェアが標準でインストールされていますが、Windows では別途インストールする必要があります。そこで Windows 上で動作する UNIX エミュレータである「Cygwin」を用いることにします。Cygwin はフリー・ソフトウェアで、<http://www.cygwin.com/> からインストーラ (setup.exe) をダウンロードできます。インストーラを起動すると、インターネット経由で Cygwin をインストールできます。インストール途中でパッケージ選択が要求されますが、四つのソフトウェアをインストールするために All を選択します(図2)。

インストールが完了すると、Windows のデスクトップに Cygwin のアイコンが現れます。これをクリックすると、Cygwin のターミナルが開きます。念のため Perl, Bison, Flex, gcc がインストールされているかを確認してみましょう。このターミナル・ウィンドウで、

```
$ perl --version
```

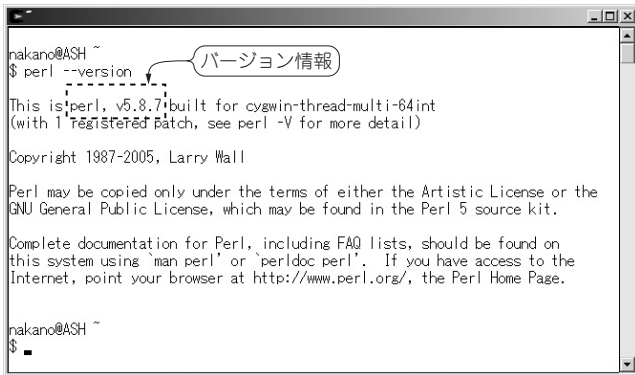


図3 Cygwinのターミナル・ウィンドウ

perlのバージョン情報が表示されている。

と入力します。図3のように perl のバージョン情報が表示されれば、インストールされていることが分かります。

同様にオプション --version を用いて、bison, flex, gcc のバージョン情報をそれぞれ表示し、インストールされているか確認しましょう。

● アセンブラに入力するアセンブリ言語の仕様

アセンブリ言語では、各行にニーモニックとラベルを記述します。各行の形は、以下の四つのいずれかになっています。

ラベル宣言：

「ラベル：」の形をとる。分岐先を表す。

ニーモニック：

ニーモニック(とオペランド)をそのまま記述。

ラベル宣言&ニーモニック：

「ラベル：ニーモニック(とオペランド)」の形で、ラベル宣言とニーモニックを1行で記述。

変数宣言：

「変数：初期値」の形をとる。変数に割り当てる領域とその初期値を表す。

オペランドはラベル、変数、即値のいずれかです。例えば図1のアセンブリ言語プログラム count.asm の「L1: PUSH n」は、「ラベル：ニーモニックとオペランド」に当てはまります。また、「n: 0」は「変数：初期値」に対応します。

● アセンブラの動作を理解しよう

アセンブラの動作を理解するために、ハンド・アセンブ