

言語レベルでのCPUの設計

1H-1

†柳澤 秀明、森 秀樹、上原 稔
東洋大学工学部情報工学科

1 はじめに

半導体集積回路 (VLSI) 技術の発展は目覚しく、VLSI の集積度の向上は現在でも進んでいる。この傾向は今後も続き、ますます増加すると思われる。VLSI の集積度が進むにつれて、複雑な回路でも1つのチップ (SOC : System On a Chip) で実現できるようになってきている。SOC 化により小型化、低消費電力化や製造コストの抑制などができるようになり、ますます利用範囲が広がり様々な種類の CPU を開発することが求められている。システム開発が大規模になり、複雑、困難になってきているにも関わらず、今まで以上に開発時間を短縮することが求められている。

本論文では、大規模、複雑化しつつある CPU の開発を言語レベルで行うことにより、ハードウェアの開発と、ソフトウェア開発環境を同時に生成することで、いままでの HDL による CPU の開発より短期間で、簡単に行なえることを示す。

2 HDL での CPU 設計

現在のハードウェアの開発は動作レベルで記述、検証することができるハードウェア記述言語 (HDL : Hardware Description Language) を用いた開発が行なわれている。HDL での開発では、設計仕様を決め、動作レベル記述を行ない、動作レベルでのテストをし、RTL 記述に書き換え、RTL レベルでのテストを行なった後で、論理合成を行ないハードウェア化を行なう。(図 1)

ハードウェアの機能テストは各段階で考えられる全ての bit パターンを入力テストパターンとして用意し、入力に対して予想される全ての出力パターンも用意する必要があり、シミュレーション

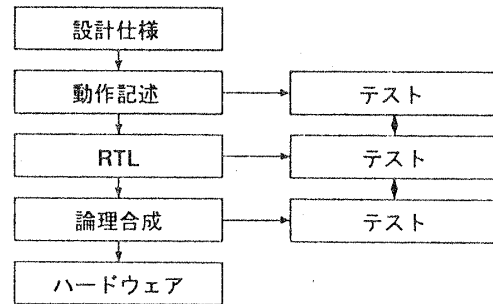


図 1: HDL を用いた設計

による出力が期待どうりの出力パターンと一致するかで動作確認を行う。しかし、大規模、複雑化しつつある回路では、全てのテストパターンを用意したとしても、期待どうりの動作をしているかを確認することは困難であり、機能検証を完全には、保証できなくなっている。

また、CPU の開発には、ハードウェアの開発とソフトウェアの開発が必要であり、回路規模が大規模、複雑になるにつれて、ハードウェアを制御するプログラムも大規模で、模複雑になりプログラムを生成をする為の負担が大きくなり、CPU の開発期間を短縮するために、ソフトウェアの開発を短期間で行なうことが求められる。

3 言語レベル設計

言語レベル設計とは、目的とする CPU のアセンブラの仕様記述を基にして、HDL によるハードウェアと、プログラミング環境であるアセンブラ、逆アセンブラ、ソフトウェアシミュレータの開発を同時に行なう。(図 2)

3.1 C'

本研究では、CPU の開発を目的とした設計自動化システムである C' (C-like Design Automation SHell) を用い、言語レベルでの CPU 開発を行なう。

C' ではビット列とニモニックを対応させアセン

[†]Design of CPU based on language level
Department of Information and Computer Sciences, Toyo University
†Hideaki YANAGISAWA, Hideki MORI, Minoru UEHARA

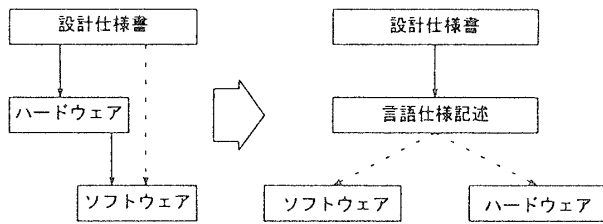


図 2: 開発手順の比較

ブラ仕様をC'言語に似た文法で記述することで、ソフトウェア開発環境（アセンブラ、逆アセンブラ、シミュレータ）と、HDLによるハードウェアの同時開発を可能にしている。

C'での機能検証は、アセンブリ言語でのプログラムを用いて、論理的機能検証を行ない、正しく動作ができることを確認してからハードウェア記述に変換することで目的とする動作を最低限保証したハードウェアの開発を行なう。

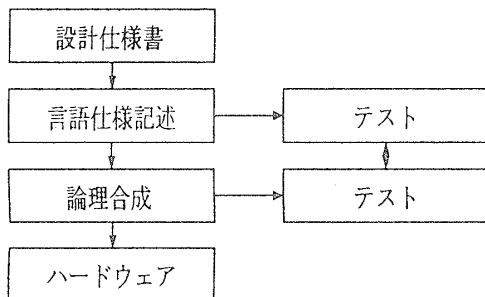


図 3: C'を用いた開発

3.2 C'記述例

```

instruct add {10000ddd}{
           ニモニック   ビット列
           動作記述
}

asm add {<label><opcode><reg>}{
        ニモニック           書式
}

```

d、は機械語となるビット列と対応する部分ビット。

3.3 ライブラリ

現在ライブラリ化している関数として次のようなものがある。

`apply_reset` ハードウェアの初期化。

`prog_load` 実行ファイルをメモリに読み込む。

`fetch` PCが指すメモリの内容をirレジスタにセットしPCを進める。

`decode` 命令の解析。

`execute` 命令の実行。

`write_result` 実行結果の出力。

`main_process` `fetch`, `execute`, `write_result` を呼び出す。

4 おわりに

本研究では、言語レベルでの開発を行なうことにより、ハードウェアと、ソフトウェア開発環境を同時に生成し、必要な機能を最低限保証したCPUを短期間で開発するための設計自動化を研究している。

現在では、シミュレーションレベルまで可能となったが、論理合成が完全に行なえるまでは至っていない。今後は、論理合成を行ない、CPUの実装を目指す。

参考文献

- [1] R.Lipsett, C.Schaefer and C.Ussery 著杉山尚志/増田洋一郎/新妻靖明/金沢彰 訳、「VHDL: 言語記述によるハードウェア設計へのアプローチ」マグローヒル出版株式会社、1990.
- [2] 中村行宏/小野定康「ULSIの効果的な設計法」オーム社、1994.
- [3] E.Sternheim/R.Singh/R.Madhavan/Y.Trivedi 著井上博史/鈴木隆 訳「Verilog-HDLによるトップダウン設計」QC出版、1995.
- [4] 小林優 著「入門 Verilog-HDL 記述」QC出版、1996.
- [5] 桜井至 著「HDLによるデジタル設計の基礎」テクノプレス、1997.
- [6] 深山正幸/北川章夫/秋田純一/鈴木正國 著「HDLによるVLSI設計」共立出版株式会社、1999.