4E-1

C'によるRTL記述の生成

†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

携帯電話や、PDA、カーナビなどいろいろな分野でCPUやDSPが利用され、短いサイクルで小型化、高性能化されているが、集積回路の微細化に対して、開発の生産性は、あまり向上しておらず、生産性の向上が求められている。また、最近のFPGA/PLDの集積度の向上により、製品としての利用が考えられるようになりつつある。

本研究では、CPUの設計自動化ツールC'の研究を行なっており、C'からRTL記述（Verilog-HDL）を出力することで論理合成可能なCPUモジュールの自動生成を行なった。

2 C'(C-like Design Automation Shell)

C'は、CPU専用の設計自動化ツールであり、アセンブラーの仕様記述により、ハードウェア(Verilog-HDLでのCPU記述）と、ソフトウェア開発環境（シミュレータ、アセンブラ、逆アセンブラ）の自動生成を同時に行うことができる(図/refabstruct）[9, 10]。設計の流れを示す。（図2）

今までのC'から出力される記述は、シュミレーションを目的としたものであったため論理合成することはできなかった。今回の改良によりFPGA/PLDへの実装を考えたRTL記述での出力をすることが可能になった。

2.1 RTL記述のための対策

論理合成できるのは、Verilog言語のサブセットであり完全な仕様をサポートしているわけではない。論理合成用にVerilogモデルを記述する場合、論理合成ツールの制約を考慮しなければならない[3, 5, 6]。今回利用する論理合成ツールとしてAlteraのMAX+Plus2を利用する。MAX+Plus2を利用するにあたり、以下のような制限が生じた[7]。

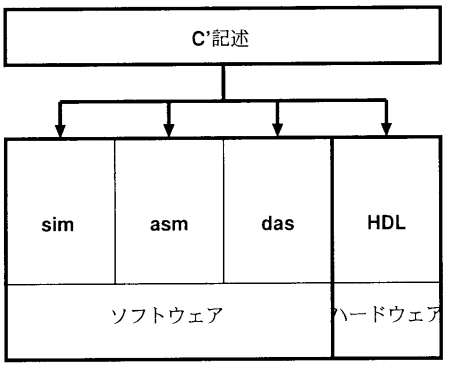

図 1: C'の概要

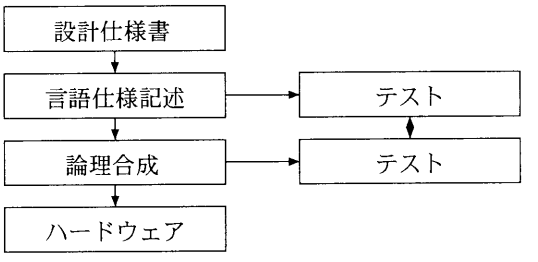

図 2: C'での設計の流れ

メモリライブラリの利用 レジスタ配列の利用ができない

各部をモジュールとして扱う taskが利用できないためモジュールとして扱う(fetch, decode, execute, write_result, main_process)

1モジュールを1ファイルとして出力 ファイル名とモジュール名が一致しないと扱えない

回路記述ととシミュレーション記述の分離 シミュレーションを目的としたテストmoduleを別ファイルとして生成する。

---

†generation of RTL descriptions using by C'
Department of Information and Computer Sciences, Toyo University
†Hideaki YANAGISAWA, Hideki MORI, Minoru UEHARA

2.2 論理合成可能なC'記述

論理合成を考慮したC'での記述では、instruct命令の記述は、演算のみの記述を行うことである。また、ロード・ストアを前提とし、各命令の実行サイクルは、単一であるものとする。

今までの記述では、演算部でレジスタ選択を行い、その出力を直接レジスタに戻す記述であった。論理合成するために各部をモジュールごとに分割するために直接レジスタを扱うことはできなくなる。論理合成可能な記述をList 1に示し、生成される演算部をList 2に示す。

List 1

```
//論理合成可能な記述

instruct ADD {01000fff} {
    REG(0) = REG(0) + REG(f);
}
instruct SUB {01001fff} {
    REG(0) = REG(0) - REG(f);
}
```

List 2

```
//execute module
module execute (port list) ;
 .
function [WIDTH:0] alu;
input [NUM:0]opcode;
input [WIDTH-1:0] bus_a;
input [WIDTH-1:0] bus_b;
begin
    case (opcode)
    'ADD: alu = bus_a + bus_b;
    'SUB: alu = bus_a - bus_b;
    'AND: alu = bus_a & bus_b;
      .
    endcase
end
endfunction

        .
    assgin carry = alu[WIDTH];
        .
```

3 評価

C'で簡単な10命令を定義し、C'から出力された各モジュールをMAX+Plus2で論理合成した結果 以下のようになった(表1)。

表 1: 論理合成結果

モジュール	セル数
fetch	62
decode	13
execute	101
control_unit	165
reg_file	301

4 まとめ

C' から生成される CPU の記述を RTL (Verilog-HDL) 記述にすることで論理合成可能なものとなった。しかし、どんな記述でも論理合成可能な出力をができるわけではなくユーザが意識しながら記述しなければならない。

今後の課題として、ユーザが意識しなくても論理合成可能な記述の生成をおこなう。

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」マグロウヒル出版株式会社、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] 「*Synthesis and Simulation Design Guide*」,Xilinx Development System, 1997.

[6] 「*Synergy Verilog*モデリング・スタイル」,Cadence Design Systems, 1997.

[7] 「*MAX+PLUSII Verilog HDL*」,ALTERA 1998.

[8] Design Wave Magazine 1999/10.

[9] 柳澤秀明/森秀樹/上原稔「言語レベルでのCPUの設計」, 情報処理学会全国大会, 1999.

[10] Dash manual