

VAX-11/780 互換プロセッサ開発によるコンピュータシステム教育

近藤 信行

清水 尚彦

東海大学電子情報学部コミュニケーション工学科 〒 259-1292 神奈川県平塚市北金目 1117
 {1adt2415, nshimizu}@keyaki.cc.u-tokai.ac.jp

1 はじめに

SoC などのハード・ソフト一体型システムが主流となってきた今、組み込みシステム技術者はハード・ソフトを合わせたシステム全体を理解することを求められている。

かねてから我々の研究室ではプロセッサ開発によるエンジニア教育を行ってきた [4][5][6]。この中で、ある程度複雑性を有するプロセッサの開発は、エンジニア教育という点において非常に効果が高いことを実証した。そこで今回は、ハードソフト両面からコンピュータシステムを包括的に理解できるよう、OSを含めたコンピュータシステム全体を構築を目的としてプロセッサ開発を行った。本報告では開発概要と性能評価及び開発による教育効果について報告する。

2 開発概要

2.1 VAX11-780 命令互換

今回開発したプロセッサは VAX-11/780 命令互換とした。理由は以下のようなことが挙げられる。

- ・ VAX-11/780 は豊富な命令とアドレッシングモードを持ち、メモリ管理ユニットやキャッシュ、プロセス管理機構など現在のプロセッサに必要な機能をほとんど備えている 32 ビット CISC プロセッサである。
- ・ gcc が移植されており、シミュレーションの際の命令データを容易に用意できると同時に、高級言語からマシン語に至る過程を理解するのに適している。
- ・ NetBSD がネイティブに動作するため、開発したプロセッサを用いて OS を動作させる際のスタートアップが早い。

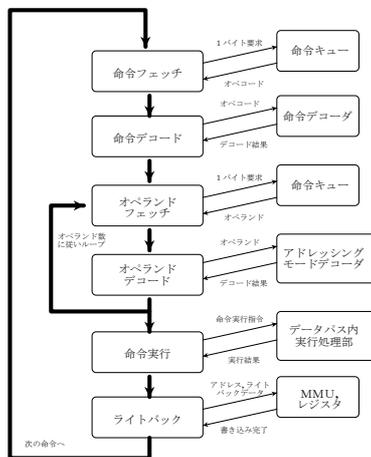


図 1 プロセッサ状態遷移図

開発は HDL を用いて全てハードワイヤードとして論理を設計した。

2.2 ステートマシン

開発した互換プロセッサはステートマシンとし、大きく分けて図 1 に示す 5 つの状態を遷移することで処理を行う。オペランドデコード、命令実行などのステートは内部でさらに詳細なステート遷移をしながら動作をする。

2.3 データパス

今回設計した互換プロセッサのデータパスを 2 に示す。データパス設計については ALU に対するファンイン、ファンアウトを減らすよう設計をした。

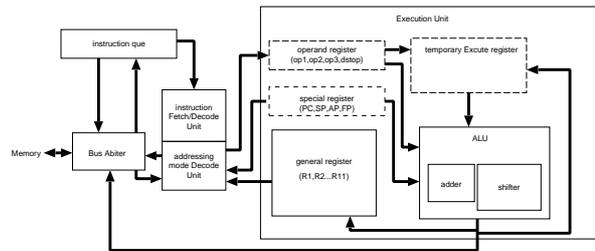


図 2 プロセッサ内部ブロック

2.4 命令キュー

VAX-11 アーキテクチャの可変長命令を実装するために、命令キューを設計した。命令キューは 32 ビット長で 4 エントリとし、1,2,4 バイトの要求を受け付け出力する。

3 開発環境

今回の開発における開発環境は表 1 に示す通りである [7]。SFL による互換プロセッサの記述量は 3,000 行弱であり、sfl2vl[2] を用いて VerilogHDL に変換、シミュレーションを行った。

表 1 論理設計環境

論理設計言語	SFL
シミュレーション	EDA linux(Icarus Verilog, GTKwave)
HDL 変換ツール	EDA linux(sfl2vl)
コンパイラ	gcc3.3.1
論理合成・実装	QuartusII4.1 Web edition

4 性能評価

開発した互換プロセッサの性能評価には、VAX 用 gcc にてコンパイルした DHRYSTONE BENCHMARK を用いた。最適化オプション "-O1", "-O2" をそれぞれ付加してコンパイルした DHRYSTONE BENCHMARK を

プロセッサ	1DHRYSTONE LOOP	VAX-MIPS	LEs	最大動作周波数
開発したプロセッサ (-O1)	4669 clock	6.56	6118	52.55 MHz
開発したプロセッサ (-O2)	4763 clock	6.43	6118	52.55 MHz

表 2 DHRYSTONE BENCHMARK 及び FPGA への実装結果

動作させた結果を表 2 に示す。回路規模、動作周波数についてはターゲットを Altera の Stratix EP1S10F780C5ES にして Quartus2 Ver4.1 WebEdition を用いて論理合成した際の最大動作周波数、LEs(ロジックエレメント数)を示す。

5 開発履歴

今回の互換コアの開発フローについて述べる。開発は学部生 1 人で 2004 年の 4 月から開始した。

2004/04

- ・アーキテクチャの調査
- ・命令キューの開発
- ・基本データパス開発

2004/05

- ・アドレッシングモードデコーダ開発
- ・Bus Arbiter の実装

2004/06

- ・算術演算ユニット (ALU) の開発
- ・整数演算命令の実装

2004/07

- ・論理合成とデータパスの見直し & 修正

2004/08

- ・分岐命令の実装

2004/09

- ・アドレッシングモードデコーダのバグを発見し、修正
- ・サブルーチンコール命令などの実装

2004/10

- ・DHRYSTONE BENCHMARK にて動作テスト
- ・オペランド属性などに起因するバグの修正

2004/11

- ・gcc が利用する命令の中で未実装である命令を実装

以上のような流れで開発を行ってきたが、開発の中で様々な問題が発生した。以下に、どのような問題が起きたかの一例を述べる。

- ・PC 更新のタイミングに関する問題

命令キューを用いて命令データを取得するため、データパス内の PC の値は常に更新せず、ライトバックステートにおいて行っていた。このため PC の値を利用するアドレッシングモードにおいて問題が発生した。PC の値をアドレッシングモードデコード中に更新することでこの問題を解決した。

- ・オペランド属性の問題

VAX-11 アーキテクチャでは命令ごとに決められたオペランドの読み書き属性が存在する。それを無視して、オペランドフェッチの際にライト属性のオペランドなどをリードしてしまっていた。解決策として、命令デコードの際に専用のフラグをセットすることとした。

6 今後の開発予定

今回の開発の最終目的は OS のブートである。よって今後は MMU, 割り込み処理機能, 未実装命令, 各種 I/O などの実装を行ったうえで NetBSD をブートさせるところまで開発を行う予定である。

7 教育的効果

現在まで開発を行ってきた、アドレッシングモードコードの方法, 関数コールなどの具体的な処理などかなりの知識を習得できた。また, DHRYSTONE BENCHMARK を動作させる際のデバッグにおいて, C 言語で書かれたコードと変換したアセンブリコードを対応させながらデバッグを行ったことで, C 言語での記述がどういったアセンブリコードに変換されているか理解すると同時にデバッグ時に C のコードをみながら大体の当たりをつけるデバッグ手法を学んだ。今後 OS のブートというところまで開発を続けることで, アーキテクチャ, コンパイラ, OS などの複合的な知識を一層深めることができると確信している。以上のようなことから, プロセッサ開発による教育効果は非常に大きいといえる。

8 まとめ

VAX-11 アーキテクチャは現在のコンピュータシステムを形成する基本的な機能を持ちながら, 学習用として適した規模である。HDL による論理設計教育用として効果的であり, オープンソースの OS と共に FPGA で実動作させることができる。また開発環境はフリーであり教育を受ける個人が容易に準備ができ特定の開発場所を必要としないため, 教育用途には非常に有益である。

参考文献

- [1] 近藤信行 清水尚彦:「SFL 開発環境を用いた VAX-11/780 命令互換 32 ビットプロセッサの開発」, 第 25 回パルテノン研究会, 2004
- [2] 清水尚彦:「The Design of Sfl2v1: SFL to Verilog Converter Based on a LR-parser」, 2003
- [3] Compaq:「VAX MACRO and Instruction Set Reference Manual」, PDF release, 2001
- [4] 飯田佳洋:「PARTHENON/SFL を用いた PDP-11 互換コンピュータシステムの開発」, 第 22 回パルテノン研究会, 2003
- [5] 飯田佳洋 清水尚彦:「組み込み向け PDP-11/40 互換プロセッサの開発」, SWEST, 2003
- [6] 大山将城 清水尚彦:「i8086 命令互換プロセッサ開発によるシステム設計教育」, SWEST, 2004
- [7] 飯田佳洋 清水尚彦:「システム LSI 統合開発環境 EDA Linux」, SWEST6 予稿集, pp.55-61, 2004