

32 ビットマイクロプロセッサの設計

氏名: 赤松 佑哉 (akamatsu, YUYA)

学生番号: 09B23595

E-mail: p5785hcf@s.okayama-u.ac.jp

出題日: 2024 年月日

提出日: 2024 年月日

締切日: 2024 年月日

概要

本実験では、ハードウェア記述言語および CAD (Computer Aided Design) ツールを利用したマイクロプロセッサ設計を通じて、論理回路、コンピュータアーキテクチャ、およびコンピュータシステムに関する理解を深めることを目的とする。特に、デジタル回路の設計から高位のシステム構築に至る一連の工程を体験することで、ハードウェアとソフトウェアの連携の重要性を理解することを狙いとする。実験内容としては、まずハードウェア記述言語と CAD ツールの使用方法を学び、それらを用いた簡単な論理回路の設計を行う。最終的な目標は、32 ビットマイクロプロセッサの設計である。実験ではまず、ハードウェア記述言語 (HDL) と CAD ツールの使用方法を学び、簡単な論理回路の設計と検証を通して基本的な回路設計の流れを習得した。次に、それらの知識を応用し、最終的には 32 ビットマイクロプロセッサの設計を目指す。プロセッサ設計においては、命令のデコード、レジスタファイルの設計、ALU (算術論理演算器) の構築、制御信号の生成といった基本機能の実装を行った。また、基本的な CPU の演算機能の実現に加えて、除算や乗算といったやや複雑な演算機能の拡張実装や、加算演算における効率化の手法の導入など、発展課題にも取り組んだ。プロセッサ設計の実験では、アセンブラ、シミュレータ、およびハードウェア設計支援システムなど、複数のツールを使用する。それぞれのツールごとに個別の設定を行わず、すべてのツールに共通する設定ファイルが用意されておりそれらを使用し実験をした。

1 設計したプロセッサの概要

本実験では、MIPS アーキテクチャのサブセットをベースとした、32 ビット RISC マイクロプロセッサ「p32」の設計を行った。このプロセッサは、教育目的に特化した構成を採っており、命令数を抑えたうえで、ハードウェア設計、命令処理のパイプライン化、およびデータパスの理解を深めることができるように設計されている。以下に、命令セット、内部構造、処理方式の概要を示す。

(1) 命令セット

p32 は MIPS のサブセットに基づき、以下のような命令群を備えている。命令形式は R 形式、I 形式、J 形式の 3 種類があり、以下の表に代表的な命令の分類を示す。

(2) 内部構造の概略

p32 プロセッサの主要な構成要素は以下の通りである。

表 1 p32 の主な命令セット

種類	命令内容 (例)
ロード/ストア命令	lw, sw, lb, sb : メモリとレジスタ間のデータ転送
演算命令	add, sub, and, or, xor, nor, sll, srl, sra, addi, subi : 算術・論理・シフト演算
条件分岐命令	beq, bne : 条件による分岐
ジャンプ命令	j, jal, jr, jalr : 無条件ジャンプ, リンク付きジャンプ
その他	lui, syscall : 定数操作, システム呼出し

- レジスタファイル : 32 ビット幅の汎用レジスタを 32 本搭載し, 2 ポート出力・1 ポート入力を持つ。
- 実行ユニット : 32 ビット ALU とシフタを含み, 加算, 論理演算, シフト演算などを 1 サイクルで実行可能。乗算器や除算器の拡張も可能。
- 制御ユニット : 命令の op コードや funct フィールドに応じて制御信号を生成し, データパスを制御。
- メモリインターフェース : 命令メモリとデータメモリを分離したハーバードアーキテクチャとする。
- メモリアーキテクチャ : ビッグエンディアンにより配置している。

(3) 処理方式の概略

p32 では複数の実行方式が検討され, 最終的には以下のような 5 ステージパイプライン構成が採用された。

- IF (Instruction Fetch) : 命令メモリから命令を取得
- ID (Instruction Decode) : 命令のデコードとレジスタ読み出し
- EX (Execute) : ALU による演算実行
- MEM (Memory Access) : メモリアクセス (ロード/ストア)
- WB (Write Back) : 演算結果をレジスタに書き戻し

また, データハザード対策としてフォワーディング機構が導入されており, 一部のハザードには NOP を挿入して対処している。パイプラインの導入により, 命令の同時実行を通してスループットの向上が図られている。

2 実施状況の報告

今回問われた課題について自分の実施状況について以下の表にまとめる。

表 2 プログラミング課題，設計課題および発展課題の実施状況

課題			状況
(プログラミング課題)			
1.	【プログラミング課題 1】 N 個の語の加算		(2) 完了
2.	【プログラミング課題 2】 N 語のメモリコピー		(2) 完了
3.	【プログラミング課題 3】乗算		(2) 完了
(設計課題 2)			
4.	【設計課題 2-1】32 ビット加算器	add32	(2) 設計完了
5.	【設計課題 2-2】32 ビット ALU	alu32	(2) 設計完了
6.	【設計課題 2-3】32 ビットシフト	shift32	(2) 設計完了
(発展課題 2)			
7.	【発展課題 2-1】32 ビット整数乗算器	mult32	(9) 非担当
8.	【発展課題 2-1】32 ビット整数除算器	div32	(2) 設計完了
(設計課題 3)			
9.	【設計課題 3-1】レジスタファイル	regs32x32	(2) 設計完了
10.	【設計課題 3-2】実行ユニット	p32ExecUnit	(2) 設計完了
11.	【設計課題 3-3】デコードユニット	p32DecodeUnit	(2) 設計完了
(設計課題 4)			
12.	【設計課題 4-1】プロセッサ	p32m1	(2) 設計完了
13.	【設計課題 4-2】プロセッサ	p32m2	(2) 設計完了
14.	【設計課題 4-3】プロセッサ	p32p1	(2) 設計完了
(発展課題 4)			
15.	【発展課題 4-1】改良		(2) 設計完了
16.	【発展課題 4-2】乗算機能の実装		(2) 設計完了
17.	【発展課題 4-3】自由課題		(1) 設計中

3 課題に関する報告

3.1 プログラミング課題に関する報告

3.2 プロセッサ設計課題に関する報告

3.3 追加課題や発展課題に関する報告

4 検討・考察

5 工夫した点や特に力を注いだ点

6 本実験の成果と実験を実施して得られたこと