

概要

今回の実験で設計したプロセッサは SIMPLE/B をベースにしている。SIMPLE/B は以下のような特徴を持つプロセッサである。

- 1 語は 16 ビットに設定されている。
- メインメモリは 1 つ搭載され、メモリ空間のサイズは 64KW である。
- 汎用レジスタのサイズは 16 ビット × 8 である。
- 命令は 2 オペランド方式であり、命令セットはシンプルだが基本的な命令は揃っている。
- 5 フェーズで 1 命令を完了し、並列化はされていない。

今回の実験では、SIMPLE/B にいくつかの改良を施した。

ハーバードアーキテクチャ

SIMPLE/B において、搭載されているメインメモリの数は 1 つであったが、今回の実験で設計したプロセッサは命令の格納されるメモリとデータの格納されるメモリが分離されている、いわゆるハーバードアーキテクチャに基づいて設計されている。

即値加算命令の追加

SIMPLE/B に搭載されている命令セットにおいて用意されている加算や減算は、レジスタの値同士の演算のみが許されていた。今回設計したプロセッサでは、演算命令におけるオペコードの空きを利用し、「あるレジスタに 4 ビット符号付き即値を加算した値を別のレジスタに格納する」ような命令を追加した。即値加算は、命令が以下のような形式であった場合に実行される。

ビット数	16 15	14 13 12	11 10 9	8 7 6 5	4 3 2 1
命令値	11	Rs	Rd	0111	d

即値加算は、アドレスが Rd であるレジスタに、(アドレスが Rs であるレジスタに格納されている値)+(d を符号拡張した値) を加算する演算である。

即値加算により、4 ビット符号付き整数で表せる範囲の数値を加算する際、レジスタに値をロードする分の命令数を削減できるだけでなく、レジスタを 1 本余らせることが出来る。例えば for ループにおいては、変数の値を 1 あるいは -1 増加させる命令が実行命令列中に多く含まれる。もしレジスタに余裕があれば、余っているレジスタに 1 をロードし、逐一そのレジスタと演算を行えばよい。即値加算の追加による恩恵は少ない。しかしレジスタに余裕がない場合、即値加算命令がなければ、レジスタのどれかひとつを一時的にメモリにストアし、空いたレジスタに 1 をロード、演算後に元の値をレジスタに戻す、という命令が必要であり、しかも即値加算のごとに命令のストア・ロードを行わなければならない。for ループを n 回繰り返す場合、全体での命令数は $O(n)$ 増加することになり、このような場合には即値加算命令の追加が効力を発揮することが分かる。

フェーズ 1・フェーズ 5 の同時実行

SIMPLE/B における 5 フェーズは以下のようなものである。

1. プログラムカウンタ値の指す番地に格納された内容をフェッチする。
2. 命令を解釈し、レジスタから必要な値を読み出す。
3. 指定された演算を実行する。
4. メモリにアクセスし、データのロードないしストアを行う。また、入出力命令においては、スイッチによる入力と 7SEG LED への出力を行う。
5. レジスタへの書き込みを行う。

今回設計したプロセッサでは、フェーズ 1 とフェーズ 5 の同時実行、すなわちメモリからの命令フェッチとレジスタへの書き込みを同時に行う。この 2 フェーズの同時実行によってパイプラインハザードは生じず、全体でおよそ 1.25 倍の処理速度向上が望める。