計算機科学実験3HW中間レポート

下田康世、橘大佑
1029-31-6231,6811
2019年度入学

2021/05/05

1 要求仕様，設計目標，設計方針，特長

1.1 実現する機能

1.1.1 即値オペランドの強化

基本アーキテクチャでは、演算命令のオペランドはいずれも汎用レジスタであるのでひとつの動作をさせるために 2 命令必要となる。そこで、表 1 の演算命令のうち、d フィールドを使わないシフト以外の演算命令に対し、レジスタ Rd と d フィールドで指定した即値を演算させるという拡張が考えられる。d の中の 1 ビットをフラグに用いることで、r[Rd]+r[Rs] と r[Rd]+sign ext(d) の切り替えを考慮する必要がある。これは SIMPLE で設計した ALU の部分を書き換えることで可能であると考えられる。

1.1.2 入出力命令の強化

基本アーキテクチャでは，ボードからの入力として IN 命令、ボードへの出力として OUT 命令がある。これらの入出力命令の入出力対象は特定の 16 ビットの入力／出力に固定されている。しかし入力／出力対象ともに 16 ビットを越える数があり、入力／出力ともそれらの中から選択して使える方がプログラムを作成する側に便利である。そこで、IN 命令ではフィールド Rs と d，OUT 命令ではフィールド Rd と d が未使用なことを利用し，そのフィールドを利用して入出力先を変更する拡張が考えられる。

1.2 性能の目標数値

即値オペランドの強化によって演算命令が高速になると考えられる。その影響は 2 命令が 1 命令になるので単純計算で 2 倍と考えられる。しかし、演算命令以外の命令は速度はそのままで速度の向上はしないので全命令の中に演算命令がどれだけの割合で含まれているかによって速度の向上の割合は変化すると考えられる。

2 高速化/並列処理の方式

2.1 拡張命令

一般的なコンパイラでは、関数呼び出しに対して、以下のような動作をするコードを出力する。
1. 関数を呼び出した命令の次の PC をレジスタに格納し、関数の先頭にジャンプする。

2. 関数を実行する。
3. 1. でレジスタに格納された値を PC に書き込み、関数を呼び出した命令の次の命令からの実行を開始する。
上記のように、わざわざレジスタに戻り先のアドレスを格納するのは関数を呼び出す場所が 1 箇所とは限らないからである。関数呼び出しをサポートするために上記の 1. の動作を行う Branch And Link(BAL) 命令と，3. の動作を行う Branch Register(BR) 命令を実装すれば，任意の場所から呼び出せる関数を実装できるようになると考えられる。

2.2 並列化, パイプライン化

2.2.1 p1/p5 の並列化

フェーズの並列実行による命令サイクルの短縮を考える。最も簡単に実現できるのがフェーズ 1 とフェーズ 5 の並列実行である。命令サイクルは 4 サイクルになる．分岐先アドレスを PC へセットする操作を工夫すれば実装できる。

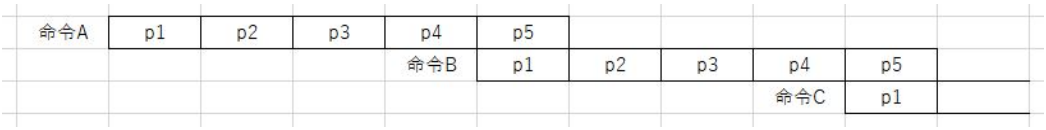

図 1: p1/p5 の並列実行

2.2.2 p1/p3 と p2/p5 の並列化

命令サイクルは 3 サイクルになる。p1/p3 を常に並列実行するためには、分岐命令を実行するために分岐アドレス計算に p3 を使わないなどの工夫が必要となる。p2/p5 の並列実行については、汎用レジスタを更新する命令の直後に、同じレジスタを参照する命令が現れた時の処置を工夫する必要がある。

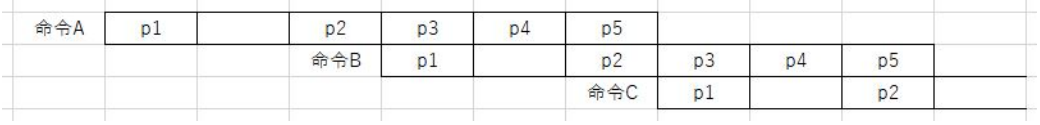

図 2: p1/p3 と p2/p5 の並列実行

2.2.3 p1/p3/p5/と p2/p4 の並列化

命令サイクルは 2 サイクルになる．2. のデータ依存の解決を更に工夫する必要がある。

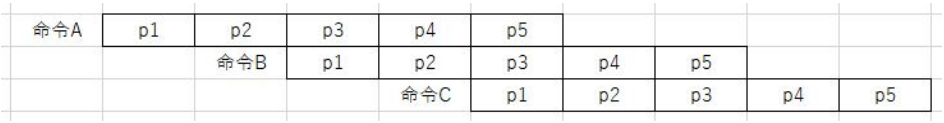

図 3: p1/p3/p5/と p2/p4 の並列実行

2.2.4 p1/p2/p3/p4/p5 の並列化

パイプライン処理と呼ばれるものである。命令サイクルは 1 サイクルとなる。

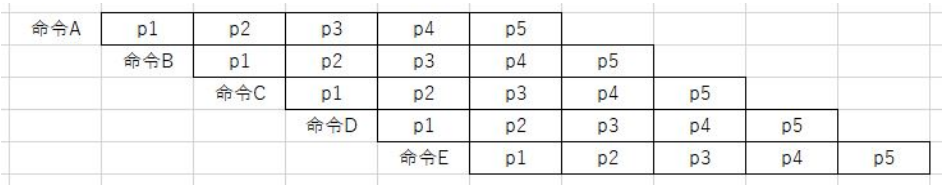

図 4: p1/p2/p3/p4/p5 の並列実行

2.3 動作周波数

並列化における動作周波数の向上について考察する。p1/p5 の並列化ではサイクルが 5 から 4 になるので単純計算で周波数は 1.25 倍になると考えられる。p1/p3 と p2/p5 の並列化ではサイクルが 5 から 3 になるので単純計算で周波数は 1.67 倍になると考えられる。p1/p3/p5/と p2/p4 の並列化ではサイクルが 5 から 2 になるので単純計算で周波数は 2.5 倍になると考えられる。p1/p2/p3/p4/p5 の並列化ではサイクルが 5 から 1 になるので単純計算で周波数は 5 倍になると考えられる。ただし、ハザードやスト－ルによってそこまでの性能にはならないと考えられる。

3 性能/コストの予測

3.1 SIMPLE/B に比べて性能

即値オペランドの強化によって演算命令が高速になると考えられる。また、並列化によって周波数も単純計算でそれぞれ 5/4,5/3,5/2,5/1 倍ほどになると考えられる。

3.2 SIMPLE/B に比べてハードウェア量

基本アーキテクチャでは、ボードからの入力による割り込み処理のサポートをしていない。そこで、ボードからの入力によって割り込みを発生させ、ボードからの入力等をトリガとして特定の処理を開始することができるようにするという拡張を行うことが考えられる。基本的に、割り込みが発生した場合、以下の流れで「現在の処理の中断→割り込み処理の実行→元の処理の再開」を行う。
1. 現在の実行中の命令を完了し、レジスタ値と条件コードを更新し、次に実行する命令の PC を確定する。
2. レジスタファイル内の全ての値、条件コード、PC の値をメモリに退避する。
3. PC に割り込み処理の命令列の最初のメモリアドレスを渡す。
4. PC に従って命令を読み出し、割り込み処理を行う。
5. 割り込み処理が終了したら、2. で退避したレジスタ値、条件コード、PC の値を書き戻す。
6. PC の値に従って命令を読み出し、1. で実行していた処理の続きを行う。

ハードウェアコストは増大するが、割り込み処理用の PC、レジスタファイル、条件コードを持ち、割り込み処理の開始／終了とともに切り替えるようにすることが考えられる。

3.3 サイクル数の予測

並列化をどれだけ行うかによって変化するが、並列化しない SIMPLE ではサイクル数は 5、p1/p5 の並列実行ではサイクル数は 4、p1/p3 と p2/p5 の並列実行ではサイクル数は 3、p1/p3/p5/と p2/p4 の並列実行ではサイクル数は 2、p1/p2/p3/p4/p5 の並列実行ではサイクル数は 1 となる。

4 考察

即値オペランドの強化、入出力命令の強化、性能の目標数値、拡張命令、並列化、パイプラインは配布された資料 simple_v40 を参考にした。他に考えられる改良としては分岐予測を実装、Jump And Link 命令を追加する、マルチコアプロセッサにするなどである。