

コンピュータアーキテクチャ 設計コンテスト2018

天野英晴

今年は性能重視

- 基本的なパイプライン処理: base
 - 基本的な5段パイプライン
 - 乗算命令を装備
- 改造した設計を基本設計に比べてどれだけ性能比が上がったかを競う。
- ただし、消費電力が40mW以内とする。これを越えると失格
- メモリは命令、データ共に2バンク利用可能とする。

コンテストデザインキット

- `tar xvf contest18.tar`
- `contest18/base`: base processorの設計キット→これは比較用に使うので直接改造しないこと。
- `cd contest18`
- `cp base contest -r` でディレクトリごとコピーし、`contest`のディレクトリで作業を行うこと

ディレクトリ構造

base,contest:
mipse.v, alu.v, rfile.vなどハードウェアの
ディレクトリ
verilog改造作業はここで行う
論理合成もここで行う

log: 合成結果

プログラム用ディレクトリ

prog_swap swap用

prog_norm norm用

プログラム変更作業はそれぞれのディレクトリで行う。ハードウェアはシンボリックリンクを張っているので、上の階層で変更すること

prog_norm

- 0番地から並んだ100個の内積を計算する
- 結果は400番地から書き出す
- Makefileのコマンド
 - make test: iverilogのコンパイル
 - make imem: norm.asmのアセンブル
- ./a.outで実行 表示された結果が21cfc8になれば正常動作している
- countで実行クロック数c(norm)、stallでストール数を出力
- 計算重視のプログラム

prog_swap

- 0番地から並んだ100個の数と400番地から並んだ100個の数を入れ替える。
- Makefileのコマンド
 - make test: iverilogのコンパイル
 - make imem: swap.asmのアセンブル
- ./a.outで実行 実行結果はresult.datのファイルに吐き出される
- diff result.dat answerで答えと同じになったら正常動作している
- countで実行クロック数c(swap)、stallでストール数を出力
- メモリ転送機能重視のプログラム

論理合成

- `dc_shell-t -f mipse.tcl | tee mipse.rpt`で合成してくださいませ。
- 電力 `log/mipse.power.log` → a
- クリティカルパスの遅延 `log/mipse.max.timing.log` → d
- 合成時のメッセージ `mipse.rpt`: エラーメッセージ

- それぞれのプログラム実行時間:
 - $t_{\text{norm}} = d \times c(\text{norm})$
 - $t_{\text{swap}} = d \times c(\text{swap})$
- baseの実行時間との比を取り、相乗平均を計算 → ポイント！
- $(t_{\text{norm}}(\text{base})/t_{\text{norm}}(\text{contest}) \times t_{\text{swap}}(\text{base})/t_{\text{swap}}(\text{contest}))^{1/2}$

設計条件

- baseのmipseの命令を削ってはならない
- 合成対象としないメモリは32ビットの命令メモリ2バンクとデータメモリ2バンクに制限
- 合成対象とするメモリは何を使っても良い(しかし、その分電力は消費する)
- 消費電力が40mWを越えてはならない。

提出

- 8月1日 18:00 厳守
- hunga4125@gmail.comに提出のこと、メールのあて先を間違えないように！
- コピーしたcontestのディレクトリをtarして添付すること
 - tar cvf contest.tar contest
 - ディレクトリの中にレポートをreport.txt中にまとめること
 - レポートには、7ページの式の各項を示し、ポイントを明記すること。
 - その他、高速化の手法でアピールすべきことを示すこと
- 質問は、tikezoe@am.ics.keio.ac.jpまで
- 内容についての質問には答えるが、デバッグは手伝わない→不公平になるため
- 授業のホームページにバグ情報が載るので注目のこと！