

コンピュータアーキテクチャ 設計コンテスト2017

天野英晴

今年は3部門を設ける

- パイプライン処理: base
 - 基本的な5段パイプライン
 - 最もハードウェアが分かりやすいがスピードアップには限界が、、、
- スーパースカラ部門: mipssess
 - 2命令同時発行静的スーパースカラ
 - 最も複雑なハードウェア、手を入れるのは至難の業、しかし性能は最も大きくなる可能性がある
- VLIW部門: mipsevl
 - 4命令分をまとめた長命令を持つVLIW
 - スケジュールだけでかなり高速化が期待できる
- 全ての設計で乗算命令(いんちゃん版)を装備
- どれか一つを選択: それぞれの基本設計に比べてどれだけ性能比 $\times 1/\text{面積比}$ (つまり面積当たりの性能) が上がったかを競う。

コンテストデザインキット

- `tar xvf contest17.tar`
- `contest16/base,mipsess,mipsevl`: 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 contest17`
- `cp base contest -r`
- `cp mipsess contest -r`
- `cp mipsevl 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

ディレクトリ構造

base,mipsess,mipsevl/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 answe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 XXX.tcl | tee XXX.log`で合成してくださいませ。
 - 面積 `log/mipse.area.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.log`: エラーメッセージ
-
- それぞれのプログラム実行時間: $d \times c(\text{norm})$, $d \times c(\text{swap})$

ポイント(面積当たりの性能比)の計算

- $\left(\frac{(d \times c(\text{norm}):base)}{(d \times c(\text{norm}):contest)} \times \frac{(d \times c(\text{swap}):base/d \times c(\text{swap}):contest)}{d \times c(\text{swap}):contest} \right)^{1/2} \times \frac{a:base}{a:contest}$

性能比の相乗平均

面積比率

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(現在のmipsevlが最大)
- 合成対象とするメモリは何を使っても良い(しかし、その分面積は消費する)

提出

- 8月4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、9ページの式の各項を示し、ポイント(面積当たりの性能比)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、mushak@am.ics.keio.ac.jpまで
- ただし、今年は絶対に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！