2018年度最終レポート

天野英晴

1. pocoにJALR命令を設ける

- JAL命令は、11bitのフィールドの範囲（－1024から＋1023まで）しか飛ぶことができない。
- そこで、レジスタ間接サブルーチンコール命令JALR（Jump and Link Register)を導入する。
- JALR rd 00000_ddd_000_11000
  - pc <- rd　r7 <- pc+1
  - R型命令、Functionは11000
- chuo18/poco/poco.vを改造
  - make
  - ./shapa jalrtst.asm –o imem.dat
- jalrtst.asmを実行してr3が９になれば正解
- 最後の書き込みで実行が終了してクロック数を表示してくれる
- 提出物：改造後のpoco.v

2.同じ命令をパイプラインプロセッサpocop.vに実装せよ

- chuo18/pocop/pocop.vを改造
  - make
  - ./shapa jalrtst.asm –o imem.dat
- jalrtst.asmを実行してr3が９になれば正解
- 提出物：改造後のpocop.v

3.性能評価

- １，２の設計をvivadoで合成し、動作周波数、電力、利用資源を評価せよ。授業と同じArtix-7の最小構成を利用せよ。
- 公平な比較を行うために、メモリを含めた合成をする
  - １はfpga.vを、2はfpgap.vをトップ階層としてaddし、その下にメモリもaddしてから合成する。
- 動作周波数は適当に設定すること
- jalrtst.asmを実行して答が出るまでに、どちらがどれだけ高速かを比較せよ。
- パイプライン化された方はプログラムをスケジュールして性能改善を試みよ。
- 提出物：両者のデータと簡単な解説