Ver. 2024-11-12

コンピュータアーキテクチャ演習(5)
Computer Architecture Exercise (5)

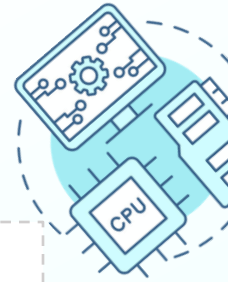
情報工学系　Berjab Nesrine

Computer Architecture support page :

https://www.arch.cs.titech.ac.jp/lecture/CA/

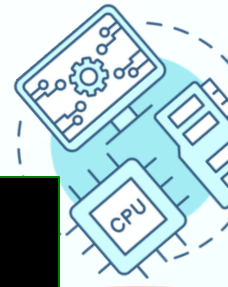
2024年度（令和6年）版

演習第五回の内容



- 最後のコンテストに向けて、高速なコンピュータ設計に取り組みましょう。

リファレンスデザインのプロジェクトのコピー (1/2)

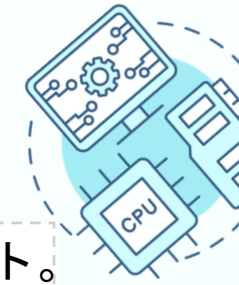



```
$ cd ~/ca2024/
$ cp /home/u_nesrine/ca2024/src/dram_proc_v1.zip .
```

❑ プロジェクトディレクトリの構成

- src/ ディレクトリ：このファイル群は、RISC-Vプロセッサのハードウェア記述と、テストや検証のためのシミュレーションファイルを含んでいる。
  - main.v: UART や RISC-V コアをインスタンス化する最上位モジュールを含むファイル。
  - top.v: シミュレーションのテストベンチを含むファイル。
  - sample1.bin: UARTで FPGA に送る src/bench ディレクトリのソフトウェアをビルドしたバイナリファイル。
  - sample1.mem: sample1.bin をテキスト形式にしたもの。
  - proc.v: mutli-cycle の RISC-V プロセッサのモジュールを含むファイル。
  - dram.v: DRAM の制御を行うモジュールを含むのファイル。

リファレンスデザインのプロジェクトのコピー (2/2)



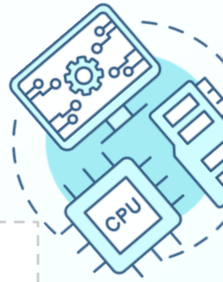
- Makefile: Verilatorを使ってVerilogコードをコンパイル・実行するためのビルドスクリプト。
  - make; make run でシミュレーションを行う。

- vivadoディレクトリ:
  - Vivado の Open Project で dram_proc.xpr ファイルを開くと Vivado のプロジェクトが開ける。

高速化を目指したコンピュータ設計の手法例 (1/2)

- 実行サイクル数を取得する
  - プロセッサの命令実行にかかるサイクル数を計測できるようにすることで、性能評価やボトルネックの特定が可能になる (Checkpoint 7)。
- コンピュータ設計をさらに高速化するための手法の一例：
  1) パイプラインプロセッサにする
     - パイプライン化により、複数の命令が同時に異なるステージ（フェッチ、デコード、実行など）で処理され、全体的な処理速度を向上させる。
  2) データキャッシュを追加する
     - 頻繁にアクセスするデータをキャッシュに保存し、メインメモリへのアクセス時間を短縮してパフォーマンスを向上させる。

高速化を目指したコンピュータ設計の手法例 (2/2)



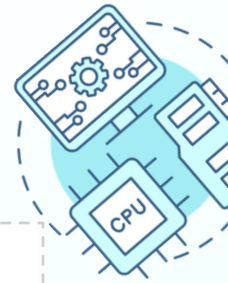
4) 分岐予測を導入する

- 分岐の際に次の命令を予測することで、パイプラインの停止を防ぎ、処理効率を高める。

5) その他の高性能化手法

- スーパースカラ; アウトオブオーダー; …

Additional comments



- vs0xx ～ vs5xx の 60台の高速 CPU を搭載したVMで、RISC-V のクロスコンパイラが利用可能
  - src/bench を自分でビルドできる。
  - 自作のアプリケーションをコンパイルして動作確認できる。