

# 自作 CPU 上で 例のレイトレを動かした話

第5回 自作CPUを語る会  
2025.04.12  
@htkymtks

# 自己紹介

- はたけやまたかし
- 株式会社永和システムマネジメント
  - RubyでWEBアプリケーション開発
- Twitter（現X）：@htkymtks

# 趣味

- 低レイヤプログラミング
  - 自作CPU
  - 自作RISC-Vシミュレーター
  - 自作コンパイラ

# 今日お話しすること

- 東大CPU実験とレイトレーシングについて
- CPUを自作してレイトレを動かすまでの流れ
- 苦労したことやハマりどころ

# 東大CPU実験とは？

- 東大情報科学科の名物実験
- 自分たちでCPU、コンパイラ、シミュレーターを作成
- その上でレイトレーサーを動かして速さを競う

# レイトレーサーとは？

- 物体と光源の光線の経路を追跡することで3D空間の見え方をシミュレートするソフトウェア
- MinCamlというプログラミング言語で書かれている

# レイトレーサーの流れ

# 始めたきっかけ

- CPUの創りかたのTD4
  - はんだ付け難しそう → FPGAならいけるかも → できた！
- 毎年見かけるCPU実験のブログ記事
  - TD4が作れたのなら、RISC-Vも作れるのでは？





# 作成した CPU

- 32ビット RISC-V
  - RV32IF（整数演算と単精度浮動小数点演算）
- 動作周波数：25MHz
- メモリ：256KB（FPGA ブロックRAM） + 64MB（SDRAM）
- ノイマンアーキテクチャ
  - プログラムメモリとデータメモリでアドレス空間を共有
- LUT数: 12000
  - TangNano 9Kには収まらない...



# MinCamlとレイトレーサー

- MinCamlコンパイラ
  - <https://github.com/esumii/min-caml>
  - CPU実験で作成するコンパイラのリファレンス実装
  - OCaml で書かれた OCaml のサブセット言語
  - 以下のアーキテクチャ向けのアセンブリを出力可能
    - PowerPC, UltraSPARC, x86
    - （ここに自分たちのアーキテクチャを追加するのがコンパイラ関係の仕事）

# MinCamlとレイトレーサー

- レイトレーサー
  - CPU実験のベンチマークプログラム
  - MinCamlで書かれている
    - <https://github.com/esumii/min-caml/blob/master/min-rt/min-rt.ml>
  - オブジェクト定義ファイルを読み込んでレイトレーシングを実行
    - 結果をPPMフォーマット(テキスト形式の画像フォーマット)で出力

# CPU開発環境

- FPGAボード: Radiona ULX3S 85F (Lattice ECP5 を搭載)
  - 2021年: 1万8000円
  - 2025年: 3万8000円 
- 開発言語: System Verilog
- 開発ツール: yosys + nextpnr
  - オープンソースな合成+配置配線ツール
  - インストールが容易
  - macOS で動く 

## 😊 今日話すこと

- TinyRubyの紹介
- コンパイラ作成Tips
- コンパイラはじめての一步

提供

株式会社 永和システムマネジメント

以下、ボツスライド



# 東大CPU実験の4つの班

- コア係（HDL を書いて CPU を作る）
- コンパイラ係（MinCamlコンパイラの移植）
- シミュレータ係（デバッグ用のシミュレータとアセンブラの作成）
- FPU係（浮動小数点演算器とライブラリ関数(sin, cos など)の作成）





# やったこと

- コア係 → ○ RISC-V CPUを作成
- コンパイラ係 → ○ MinCamlコンパイラを移植
- シミュレータ係
  - シミュレータ → 既存の RISC-V シミュレータ (Spike) を利用
  - アセンブラ・リンカ → RISC-V GNUツールチェーンのものを利用
- FPU係
  - FPU → 既存の FPU コアを利用
  - ライブラリ関数 → ○ 書いた