

プロセッサ・コンパイラ実験 最終発表（03/05）

完動班（5班）
富田、卜部、名取、濱中

最終発表内容

- 今回の実験にたいする取り組み
- (開発スタイルについて)
- アーキテクチャについて
- 余興
 - RTA, CPU プレゼン , OS
- (シミュレータ、FPU まわりについて)
- コンパイラについて
- レイトレ実演

完動班()の CPU 実験

- Real Time Attack
 - 速い CPU を作る→速< CPU を作る
- CPUExSolver
 - <https://github.com/furaga/CPUExSolver>
 - を用い、足掛け 11 日で 1st 完成 (09/18)
 - マルチクロックサイクル

開発スタイルについて

- イテレーティブな開発
 - いきなり目標を目指さない。到達可能な地点を目指す。
- フィードバックループ
 - 開発が正しい方向に進んでいるか常に確認
- テスト駆動開発

アーキテクチャ

- 1st→ 完成時点で min-rt 動作時間 110sec.
- 2nd→ パイプライン化
 - メモリまわり 5 段、FPU6 段、その他 4 段
 - Forwarding
 - 実際にハザードが発生するときのみストール
 - 分岐では遅延スロット(2段)を利用

余興

- 10 月 : RTA (前述)
- 11 月 : CPU プレゼン
↓
- 12 月 : OS 自作

※ フラクタルフレームは現状 2nd 仕様のシミュレータで動作せず

余興 - CPU プレゼン

- CPU でプレゼン

- FPGA に拡張基板を接続、グラフィックモジュールを利用してディスプレイに VGA 出力

- ・ 10er の hole さんありがとうございました

- mincaml でテキストデータから表示文字列と位置を読みとりスライドの形に整える

- キーボード入力 (ENTER でページ送り) も受け付ける

余興 - OS 自作

- CLI（入力：キーボード、出力：VGA）
- C コンパイラのサブセット、バイナリの実装
- 基本的な Unix コマンド
- テキストエディタ (like vi)

余興 - OS 自作

- Lisp インタプリタ (like Lisp 1.5)
- SD カード (FAT ファイルシステム) へのアクセス
 - IIDX 班にモジュールの一部をいただきました
- 複数プロセスのネスト
- 割り込みなしのシングルプロセス

FPU, シミュレータ

- FPU→CPUExSolver の出力 (ほぼ) そのまま
-fdiv 用 finv のみ最適化処理
- シミュレータ→ 1st 用は同上
2nd についてはいくつかの命令の削除
(mul,and,or,nor,muli...) や、余興の
ためのディスプレイなどに対応

コンパイラ

- データ構造等は mincaml のままで、最適化処理を付加
- K 正規形の代わりに A 正規形を利用
- 最終的な min-rt の発行命令数
 - 約 15.5 億 (1,551,366,996)

最適化処理

A 正規形に対して：

- 共通部分式除去
- 複数の if 文を 1 つにまとめる
- エイリアス解析とそれを用いた最適化
- インライン展開を非再帰関数に対して重点的に行うように
- if 文の各分岐先に後続の式をつなげる
- lambda lifting

最適化処理

クロージャ変換処理に対して：

- グローバル配列、グローバルクロージャのアドレスを静的に決定
- (タプルの平坦化)
(引数のタプルの展開)
(配列へのタプルの埋め込み)
(タプルの削除)
→ 命令数が増えてしまったので廃止

最適化処理

- グラフ彩色によるレジスタ割り当て
アセンブリコードに対して：
- 2 段ジャンプ、直後へのジャンプ等
不要なジャンプの除去
- 同じコード列を 1 つにまとめる
- 遅延スロットを埋める複数の if 文を
1 つにまとめる

レイトレ実演

謝辞

- @__furaga さん
 - CPUEx Solver
- @h013 さん
 - 拡張基板、グラフィックモジュール
- IIDX 班の皆様
 - SD カードへのアクセス機構