

完動班進捗報告 11/13

IS3年 完動班 卜部、富田、濱中、名取

そろそろ「完動」の名が
重くなってきたこのごろ

余興

The LLVM Compiler Infrastructure

<http://llvm.org>

- C/C++
- Java
- Python
- Ruby (Rubinius)
- Haskell

LLVM backend

LLVM 中間言語から自作命令セットへの変換方式を LLVM のツールを使って定義する。

```
; ModuleID = 'shift.c'
```

```
define i32 @shift(i32 %x)
nounwind uwtable {
    %1 = alloca i32, align 4
    store i32 %x, i32* %1,
align 4
    %2 = load i32* %1, align 4
    %3 = lshr i32 %2, 5
    ret i32 %3
}
```

```
define i32 @main() nounwind uwtable {
    %1 = alloca i32, align 4
    store i32 0, i32* %1
    %2 = call i32 @shift(i32 1024)
    ret i32 %2
}
```

```

.file "shift.s"
.text
.globl shift
.align 4
.type shift,@function
shift:                                # @shift
# BB#0:
    sti $r3, $r14, -4
    or $r1, $r0, $r3
    srai $r3, $r3, 5
    andi $r3, $r3, 134217727
    return
    sti $r1, $r14, -8
.Ltmp0:
    .size shift, .Ltmp0-shift

.globl main
.align 4
.type main,@function
main:                                # @main
# BB#0:
    mvlo $r1, 0
    sti $r1, $r14, -4
    mvhi $r1, 103
    ori $r3, $r1, 27005
    call shift
    return
.Ltmp1:
    .size main, .Ltmp1-main

```

進捗

完全に失敗しました…

ひと通り書き換えが終わったところで、CALL RETURN の扱いに根本的な問題があることが発覚。

面倒になったのと、FractalFlameが公開されたので断念。

Fractal Flame

DRAMうごかん

コンパイラうごかん

進捗

- コア
 - DRAMコントローラをMIGベースで作ろうとした
 - タイミング制約の問題が解決できない
- コンパイラ
 - 必要な構文等を実装
 - コンパイルは通る
 - gene ごとの実行はできる

今後の目標

- 問題のありかを特定しやすくするため、問題を分割してテストできるようにする
- コンパイラが生成したコードの正当性を保証する仕組みを考える
- キャッシュの構成を実験的に決める
- より効率のよいメモリ管理方法を考える
- DRAM!