

###ファイル構成 <アセンブリ> `assembler.cpp`に実装 fibを0,1の機械語に変換

<シミュレータ> 1 ファイルで実行する場合 最初の方のbool debug = true;をfalseに変えると実行速度が速くなる。  
trueにするとデバッグ用の出力が表示される。

**実行の仕方** `g++ sim_c++_改訂版.cpp -o sim_c++_改訂版` の後に `./sim_c++_改訂版` などをコマンドラインで入力。 別ファイルでも同様

\*がある行で s or n or mと出力される →sかnかmを入力

s:ステップ実行 n: 次の\*まで飛ぶ m: レジスタの中身表示（有限の値のレジスタのみを出力します。）

高速化\_with fpuフォルダ（エミュレートが入ったもの）でmake `./sim -f`で実行(fast\_mode)

-g(not fast\_mode) 統計情報取得モード ストールなども含めた実行命令数を出力 hit,miss率も出す

`sim_0206.cpp`; 1つにまとめたファイル

makeでコンパイル可能 -fをつけて実行するとfast\_modeに -gだとキャッシュなどの統計情報が出る -dでデバッグモード（1つのファイルであっても bool debugやbool fast\_modeを変えることによりモードを変えることができます）

Ack関数結果

```
2 1024
5 8189
6 8188
hit miss 0 0
clean dirty 0 0
実行命令数 402260398
duration = 3.79298sec.
```

## 注意点

```
void sld_to_ppm(){
    string filename ("contest.dat"); //asm_3
    vector<string> lines;
    string line;
    string s1 = "/Users/maimai/my-3A/cpu-simu/sim_contest.ppm";
```

という部分はstring s1の名前の変更が必要 入力ファイルcontest.dat(ppmファイルをdatファイルに変換したもの)を手元に置いてお使いください base.datに書き換えられていたらbase.datを使用して下さい

cppファイルと同じフォルダにdatファイルとアセンブリ(.s)ファイルを置いて実行

レジスタの使い方

```
x0...常に0を格納
x1...戻りアドレスを格納
x2...スタックポインタ
x3...ヒープポインタ
x5-x23...汎用レジスタ
x24...定数を一時的に格納するレジスタ
浮動小数点数用のレジスタ
f0-29... 汎用レジスタ
f30...ゼロレジスタ
f31...定数用レジスタ
```

x2の初期値はSIZE=1024 それ以外のレジスタは初期値0 'm'でレジスタ出力されたときに値が0のレジスタは出力されず、有限の値のレジスタのみを出力します。 <FPUのエミュレート> TARGET = fpu\_sim

## コンパイル対象のソースコード

```
SRCS = fsqrt_sim.cpp SRCS += simu_fmul.cpp SRCS += simu_fadd.cpp SRCS += fpu_common.cpp
```

###RISC-V命令について 疑似命令も一般的なもの処理できる

**デバッグ機能** -d: デバッグモード -dの後に print mem: メモリの中身をprint \*がついたアセンブリの行で break (一番最初の\*)

**実行時の流れ** 最初に一度ファイルを全て読み込んだ時点でpcの値などが流れて出力され、次にファイルを読み込んで実際の処理をするときは、\*がある行まではラベルのみが出力され\*の行にくとその行が出力されるとともに実行が一時停止され、s or n or m と出力されます。sかnかmを入力して下さい。

**更なる注意** ここにある仕様は一時的なもので、今後デバッグしていくうちに断りなく改変されることがあります。信じすぎないで困ったらコードを少し見るかすぐにslackなどで訊いて下さい。