コンパイラが吐き出した

32bitの命令を読み取ってそれからADDやらなんちゃらを読み取ってレジスタに格納というのをシミュレーションする

例えば

Addは

上から

0~5bit 31

6~10bit代入先レジスタ

11~15bit RA　一つ目レジスタ

16~20bit RB 二つ目レジスタ

21 OE addの種類を決める？

22~30 266

31 Rc

Special Registers Altered:

CR0 (if Rc=1)

SO OV (if OE=1)

よくわからんけどこうなってる

1stアーキテクチャ

初めの6つで命令を規定して残りの26個でレジスタなどを規定

ADDでは

31~26 命令の種類

25~21 代入先レジスタ

20~16 元レジスタ1

15~11 元レジスタ2

下は全て0

ADDI

31~26 命令の種類

25~21 代入先レジスタ

20~16 元レジスタ1

下はimm

<https://github.com/cpu-experiment-2018-2/assembly/blob/master/src/encode.ml>

32bitの命令列が改行区切りでずらっと並ぶ感じのテキストが出力される

* + 各ステップごとにレジスタの値を表示して欲しい
  + 仕様
  + %r3 - %r14 -> int ,%r15 - %r26 -> floatとする感じで
  + 呼び出し規約はPowerPcのものを模倣する感じにします
  + 以下は特別に表示して欲しい
  + %r0 ゼロレジスタ
  + %r1 スタックの先頭
  + %r2 スタックフレーム
  + %r31 リンクレジスタ
  + %r30 コンディションレジスタ

Make前にeval $(opam config env)忘れずに