

AL_Project1 説明文件

S1081535_110/04/21

設計理念

1. 首先從指定 `file` 中以行為單位讀入 `Instruction`，存入 `Inst_s` 陣列。

2. 檢查 `Instruction` 是否為 `Label` 之起始 `Instruction`。(`check_label ()`)

- 是：將 `Label` 的名字存入 `label_name` 陣列中；並在 `label_loc` 陣列相對應的位置存入 `instruction` 的位置，方便 `Label` 被呼叫時計算 `offset`。再整理 `Inst_s`，讓開頭沒有 `Label` 和空白。
- 否：整理 `Inst_s`，使其開頭非空白及 `tab`。

3. 將 `Instruction` 的 `operation` 單獨存放，方便判斷型態。

4. 重複(1)(2)(3)，直到所有 `Instruction` 都讀入。

5. (從第一個 `Inst_s` 開始) 若有 `operation` 則判斷是哪種指令，並進入指定的 `Instruction type function` 運算。

- `R-type` : `getvalue_r ()`
- `S-type` : `getvalue_s ()`
- `I-type (offset)` : `getvalue_io ()`
- `I-type (immediate)` : `getvalue_ii ()`
- `R-type (rv64i)` : `getvalue_rup ()`
- `SB-type` : `getvalue_sb ()`
- `U-type` : `getvalue_u ()`
- `UJ-type` : `getvalue_uj ()`

6. 進入指定的 `type function` 後，每個 `type function` 有以其形態所需之變數，例：`func3_(type)`、`rd_(type)`、`offset_(type)`等。分別存取 `RISC-V Code` 中對應之值。

7. 利用 `Binary ()` 可將由10進制的表示的數值轉為2進制；`fix ()`可補足不足格式的 `bits`；`find_address ()` 可用來找尋需呼叫之 `Label` 的 (相對位置 / 4)。

8. 待所有變數皆轉換為格式所需，依照格式輸出 `Machine Code` 至指定 `file` 中。

9. 重複(5) ~ (8)直到所有 `Inst_s` 皆輸出 `Machine Code`。

程式操作說明

- ◆ 編譯執行前，先輸入讀入和輸出的檔案。

```
6      #include<fstream>
7      using std::ifstream;
8      using std::ofstream;
9      ifstream infile("test.txt", ios::in);
10     ofstream outfile("ans.txt", ios::out);
```

- ◆ RISC-V Code 輸入格式皆依照 Green Card 上的格式。
 - EX : jalr x0, x1, 0
- ◆ Register 的表示方法皆使用 x0 ~x31 (X0 ~ X31)
- ◆ RISC-V Code 後盡量不要有多餘的字符