Readme : RISC-V Assembler

Praneeth Chamarthy && Gona Sanjana

September 8, 2024

1 Project Structure

```
project_root/
          src/
                main.c
                parser.c
                utils.c
          include/
                assembler.h
                parser.h
                utils.h
          tests/
                unit/
                      test_register_number.c
                      test_parse_rtype.c
                      test_pasre_itype.c
                      test_parse_stype.c
                integration/
                      simple_add.s
                      simple_add.o
                      branch_and_jump.s
                      branch_and_jump.o
                      arithmetic.s
                      arithmetic.o
                      load_store.s
                      load_store.o
                edge_cases/
                    immediate_bound.s
                    mixed_instructions.s
                    pseudo_instructions.s
                Error Hanadling/
                    invalid_immediate.s
                    invalid_instruction.s
                    invalid_register.s
          docs/
                report.pdf
          Makefile
          README.md
```

2 File Descriptions

2.1 Source Files (src/)

- `main.c`: Contains the main program logic for the assembler.
- `parser.c`: Implements parsing functions for different RISC-V instruction types.
- `utils.c`: Provides utility functions for the assembler.

2.2 Header Files (include/)

- `assembler.h`: Main header file with common definitions and structures.
- `parser.h`: Declarations for parsing functions.
- `utils.h`: Declarations for utility functions.

2.3 Test Files (tests/)

- `Unit tests`: Test individual functions (e.g., register number lookup, R-type instruction parsing).
- `Integration tests:` Test complete assembly of simple programs.
- `Edge case tests`: Verify assembler behavior with large immediates and maximum number of labels.
- `Error Handling tests`: Verifies the assembler behaviour when there is an error in the input file

3 Usage Instructions

1. **Compilation:**

```
make

```

   This will compile the assembler and create the executable in the `bin/` directory and the created objected will be present in `obj` directory

2. **Running the Assembler:**

```
make run

```

   This command will:

   - Copy the executable to the current directory
   - Copy the input file (`input.s`) from the `input/` directory
   - Run the assembler
   - Move the output file (`output.hex`) to the `output/` directory

3. **Cleaning the Project:**

```
make clean

```

   This removes all generated files, including object files and the executable.

4 Input and Output

- Input: Place your RISC-V assembly code in `input/input.s`.
- Output: The assembled machine code will be written to `output/output.hex`.

5 Supported Instructions

This assembler supports various RISC-V instructions, including:

- R-type: add, sub, sll, slt, sltu, xor, srl, sra, or, and
- I-type: addi, slti, sltiu, xori, ori, andi, slli, srli, srai, lb, lh, lw, lbu, lhu
- S-type: sb, sh, sw
- B-type: beq, bne, blt, bge, bltu, bgeu
- U-type: lui, auipc
- J-type: jal
- Special: jalr

Additionally, it supports several pseudo-instructions and RV64I instructions.

6 Testing

The `tests/` directory contains various test files:

- Unit tests validate individual components of the assembler.
- Integration tests check the assembler's performance on complete programs.
- Edge case tests ensure the assembler handles extreme scenarios correctly.