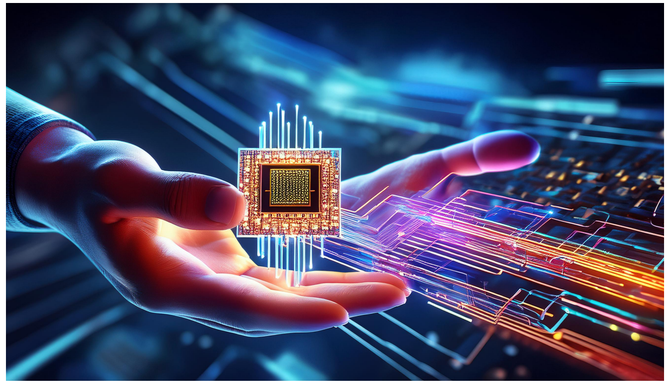
26.11.24 17:59

CSCI 250
Introduction to Computer Organisation
Lecture 5: Compiler Basics III



Jetic Gū

2024 Fall Semester (S3)

Overview

- Architecture: von Neumann
- Textbook: CO: 4.5
- Core Ideas:
    1. Assembler Requirement

P1
ASM

Lab 5

Assembler Requirement

P1
ASM

Assembler Directives

- Your assembler will support 1 directive, `.global`
- This line could appear anywhere in the assembly code, except for within the definition of a symbol
- Main function must be defined using `.global` as `_main`

Lab

P1
ASM

Symbols

- Your assembler will the definition of symbols
  - `_main` is mandatory
- Your assembler should also support other symbols, so your programmer can define other functions. Your programmer could try to name the symbol anything, but it has to meet the requirement for variable names for other programming languages
    - Exception: `x0-x6` or `X0-X6` should be reserved for register names
- Literal numbers need to be supported, including Hex and Bin. See LS17 for detail.

Lab

P1
ASM

B Instruction

- `B/BL` is the only way for the programmer to change `PC` (`R7`)'s value
  Your assembler should prevent `PC`'s value from being changed by other instructions, including by the programmer using `MOV`

- `BL` is not in the list of instructions that your CPU need to support, however your assembler will support it, by

  - Saving current `PC` to `R6` (LR: Link Register) using `MOV`

  - Use `B` to branch to target function, after its finished execution,

  - Use `MOV` to go back to original `PC` stored in `R6` (e.g. main function)

Lab

P1
ASM

Assembler Output

```
$ python3 asm_arm16.py programme.s -o programme.ic
$ ./asm_arm16 programme.s -o programme.ic
```

- Our development target is your ARM16 CPU computer, so your assembler should output in a way such that it can be loaded into the instruction cache of your ARM16 CPU.
- Your assembler will take 1 mandatory argument for the input filename (e.g. `programme.s`), and 1 optional flag `-o` for specifying the output filename
- Your assembler can be programmed using Python3 or C++
- How do you test it? You need to be able to copy the assembler's output to the instruction cache, and see it running.

Lab