FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Review report of a final thesis

Reviewer:	Ing. Tomáš Pecka
Student:	Bc. Ivo Strejc
Thesis title:	Tiny x86 - Architecture Simulator for Educational Purposes
Branch / specialization:	System Programming
Created on:	31 May 2021

Evaluation criteria

1. Fulfillment of the assignment

- ▸ [1] **assignment fulfilled**
  [2] assignment fulfilled with minor objections
  [3] assignment fulfilled with major objections
  [4] assignment not fulfilled

Assignment was fulfilled. I have no objections.

2. Main written part

70 / 100 (C)

The chapters see to be logically structured. However, I think the thesis could be longer; this is my main objection. Some chapters could be longer (e.g. chapters 1, 2). Also more examples and figures could be provided. For instance in chapter 2 (where CPU features are discussed) an example of a hazard or a wrong branch prediction would be beneficial for readers. Although not required, I also think that a class diagram (maybe also a sequential diagram) would be nice for better understanding of the VM design in chapter 5. Also the design choices are sometimes discussed only briefly.
I welcome the ISA documentation in the appendix. The documentation of the VM (how to run, etc.) could be more detailed.

There is a small error on fig. 4.1 where the order of IF and ID operations is reversed, however, the order is correct in the text. Apart from that, there are no inaccuracies.
There are several minor typographic errors (overflows, empty \item in itemize environment on p. 30, etc.). I think that there are better fonts for code samples and code identifiers than \textit (see e.g. p. 30 or p. 33).

I really appreciate the choice of the English language. However, the thesis contains (in my opinion) a lot of typos and some sentences make no sense (because of the grammar). I really recommend to use at least a spellchecker.

3. Non-written part, attachments 95 /100 (A)

The resulting VM in the tinyverse project turned out to be very good and seem to be working well. The tiny86 ISA and addressing modes are designed well. I welcome the DBG instruction (I can imagine the use-case for students projects.). The provided interface for building a tiny86 program is simple and intuitive.
There are some minor drawbacks:
- Unfortunately the CPU caches were not implemented. The data are always read/written from/to the RAM.
- The VM is tested only by several tests. These tests check whether a run of a tiny86 program results with a certain value. This is of course good, however, I'd welcome some unit tests for individual components of the VM.
- The compiler from tinyC language into tiny86 still needs some work. Some language constructs are not translated properly (e.g. ` for(int i = 0; i < 5; i++) body;` gets stuck in an infinite loop, bug using the prefix incrementation it works just fine). However, I think the VM was the main part of the project so I don't see this as a big deal. The (working) compiler could have been used for some high-level tests.
- The commit history could be improved. Many commits contain just generic messages ("update", "wip", "fixes", ...). Sometimes it is really hard to understand the motivation behind the commit without any description.

4. Evaluation of results, publication outputs and awards 100 /100 (A)

I believe the tiny86 VM is good and I hope it will prove useful in courses like NI-GEN.
I, for one, hope that the tinyverse project will continue and new features will be added. I'd like to see a debugger for the CPU of the VM. I can imagine this project being used in other courses dealing with CPUs and/or compilers (BI-APS maybe?).

The overall evaluation 90 /100 (A)

The written part of the thesis has some shortcomings. However the core of the thesis was the (non-trivial) implementation of a VM and it was done very well.
I evaluate the thesis with 90 points (A).

Questions for the defense

- Can the VM be extended to run multithreaded programs with critical sections? What needs to be added and/or changed in the VM (and ISA)?
- The semester is almost over; have you collected any feedback from students (apart from the things already mentioned in section 6.3)? Are there any important issues/comments/feature requests?

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 26/2017, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.