

Flow-Sensitive Typing for the SUS Language

Master Thesis

At a glance

- Research existing flow-sensitive typing approaches in Literature and other compilers, such as TypeScript
- Get to know the style and semantics of SUS code
- Get to know the existing infrastructure of the SUS compiler codebase
- Evaluate and implement possible approaches in the existing codebase

At PC2 we currently develop a new language and compiler, named “SUS”, that aims to help in the design and development of high-frequency FPGA accelerators. It does this by explicitly encoding pipelining information in the type of each wire, and using it both to automatically balance pipelines, as well as participating in type inference of submodule parameters.

A major feature of SUS is its “bounded integers”. In hardware design (as opposed to its software counterpart), the ranges of integer wires within a design are usually well-known, and SUS encodes this in the type system. Operations on these integers result in new bounds. $(0:20) + (5:10) \rightarrow (5:24)$ for instance. Another common operation on integers is first a boundscheck, and then use of the integer. For instance, currently in order to compile correctly, the below code would need to be patched with a `UnsafeIntCast#(FROM: 0, TO: 5)(idx)`. Such Unsafe casts are unwieldy and error-prone, and would be eliminated with flow-sensitive typing.

```
module IntegerNarrowing {
  int#(FROM: 0, TO: 256) idx

  bool[5] myArr = 5'b01001

  when idx < 5 {
    | bool _x = myArr[idx]
  }
}
```

Recommended Background:

- Experience with one or more system-level programming language. Preferably Rust
- Interest in Compiler Design
- Experience with Hardware Design is a plus, but not a necessity.

Further reading:

<https://github.com/pc2/sus-compiler>

https://en.wikipedia.org/wiki/Flow-sensitive_typing

<https://ayazhafiz.com/articles/22/why-dont-more-languages-offer-flow-typing>

Contact:

Lennart Van Hirtum, Phone: 05251/60-4542 E-Mail: lennartv@mail.uni-paderborn.de

For information on theses offered by the HPC group or PC2 please visit:
<https://en.cs.uni-paderborn.de/hpc/teaching/open-theses>