Verifying A Process to Build Critical Digital Systems.

Vincent Iampietro, David Andreu, David Delahaye

Critical Digital Systems.

Figure: Examples of Critical Digital Systems.

```
If BUG or FAILURE Then
    Death;
    Injuries;
    Natural Catastrophes;
    Financial Losses;
    ...
```

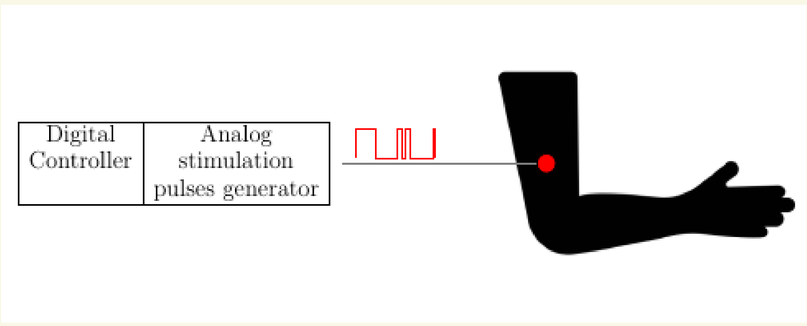

Figure: A Critical Digital System: Neurinnov's Neuroprothesis [1].

How to build them?

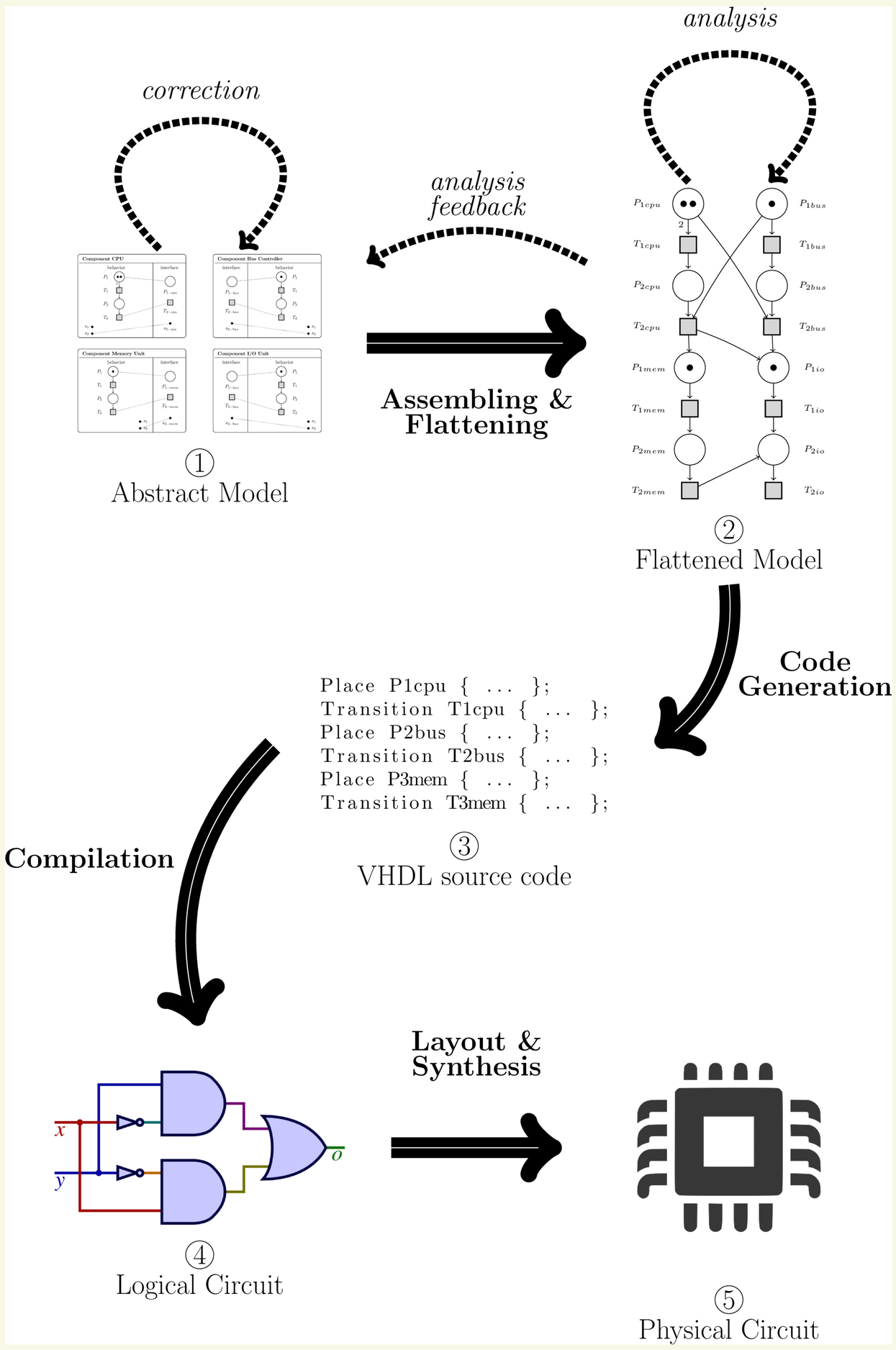

Figure: HILECOP: A Methodology to Build Critical Digital Systems [1].

The Petri Net (PN) formalism.

- A formal model used in HILECOP to describe the behavior of digital systems.
- HILECOP PNs are Synchronously executed, extended, generalized, Interpreted, Time Petri Nets with priorities and macroplaces (SITPNs).

Is It Safe? Verifying HILECOP

HILECOP models are safe thanks to analysis, but are model properties preserved through transformations?

What do we want to verify?

We want to formally prove that the transformation from the *Flattened Model* ② to *VHDL source code* ③:

- Preserves the structure of the model.
- Preserves the *semantics* of HILECOP Petri nets (PNs).

Proof Assistant.

Proofs will be conducted with the Coq proof assistant [3], which is:

- A Generic Functional Programming Language.
- A language for proof verification.

Proof steps.

Inspired by the works on *CompCert*, the certified C compiler [2]:

1. Model HILECOP PNs semantics.
2. Model VHDL language semantics.
3. Implement transformation and establish proofs of structural and semantical preservation.

Petri Nets Semantics.

The set of rules regulating the evolution of PNs.

That's one small step for the proof...

What we have done so far:

- Model the structure of Synchronously executed Petri Nets (SPNs).
- Model SPNs semantics.
- Implementing a token player program sound and complete regarding SPN semantics.

References.

D. Andreu, D. Guiraud, and S. Guillaume.
A distributed architecture for activating the peripheral nervous system.
Journal of Neural Engineering, 6(2):18, Feb. 2009.

X. Leroy.
Formal Verification of a Realistic Compiler.
Communications of the ACM (CACM), 52(7):107–115, July 2009.

The Coq Development Team.
Coq, version 8.9.0.
Inria, Jan. 2019.
http://coq.inria.fr/.