Design and Verification of Speed-Independent Multiphase Buck Controller

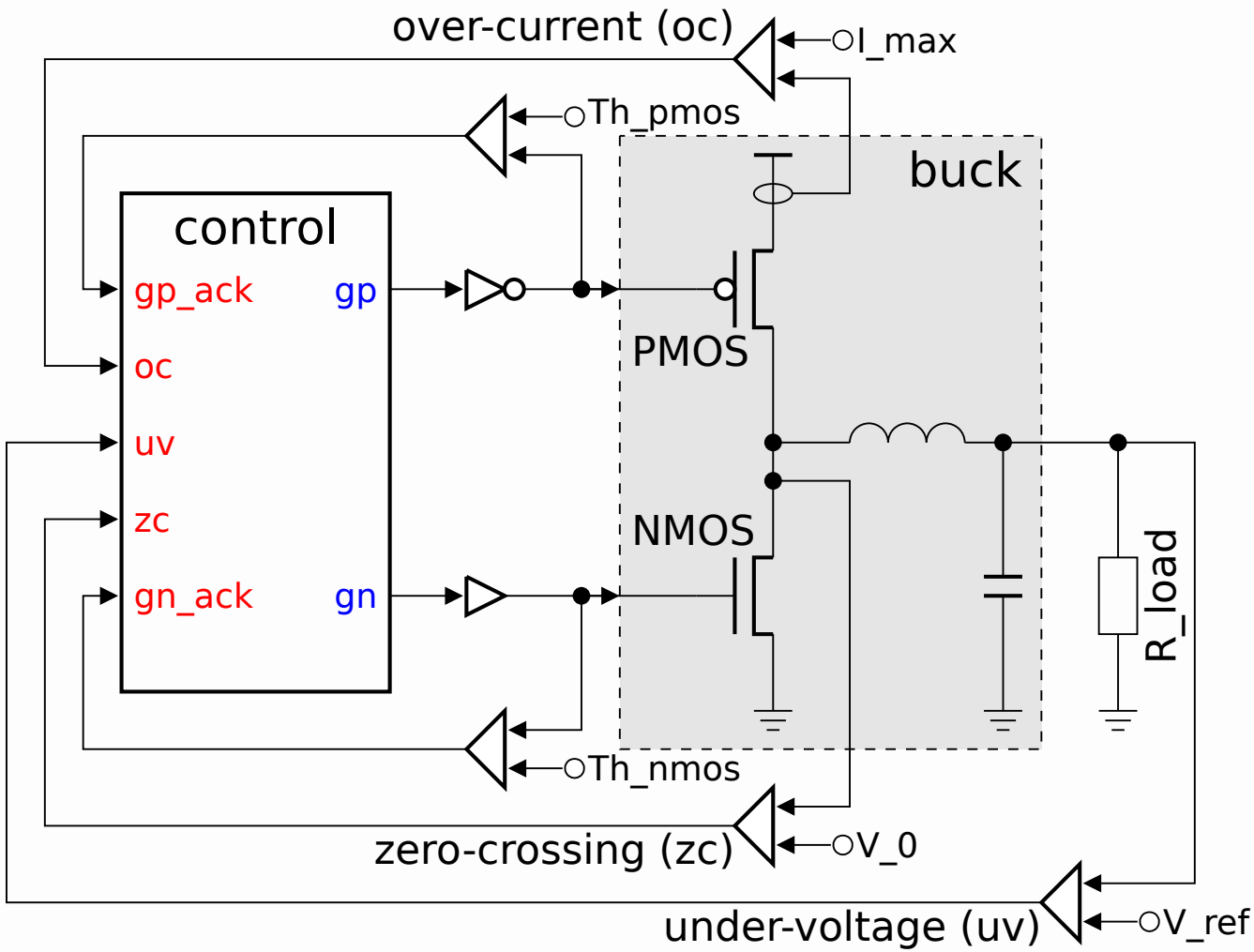
Danil Sokolov[1], Victor Khomenko[1], Andrey Mokhov[1], Alex Yakovlev[1], David Lloyd[2]

[1] *Newcastle University, UK;* [2] *Dialog Semiconductor, UK*

Motivation

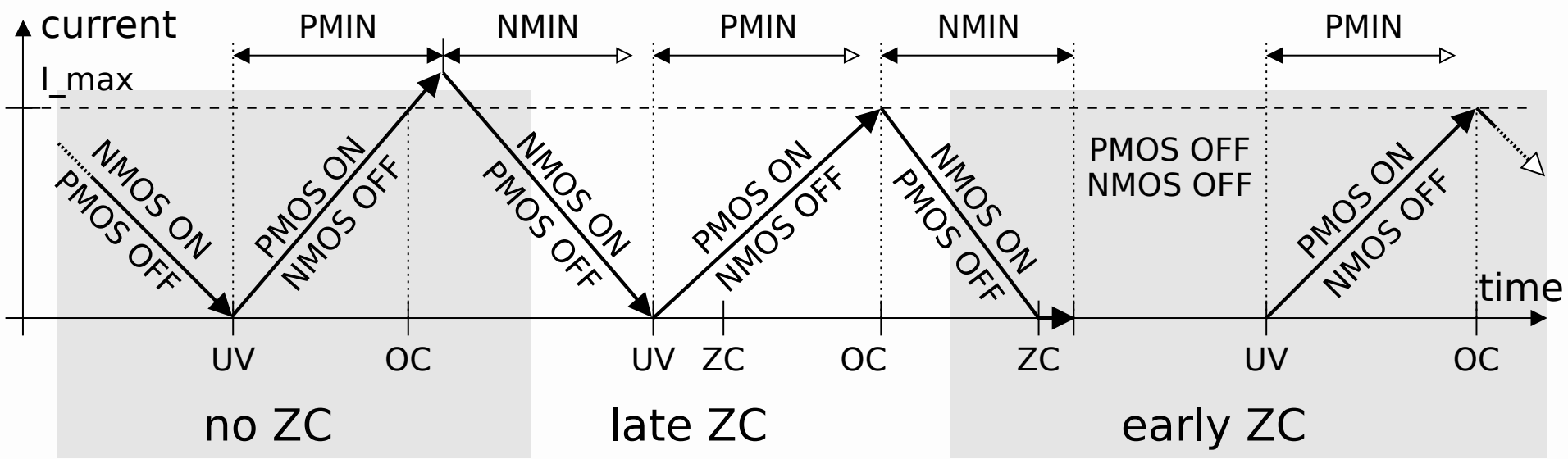
- Efficient implementation of power converters is paramount
  - Extending the battery life of mobile gadgets
  - Reducing the energy bill for PCs and data centres (5+3% of global electricity production)
- Need for responsive and reliable control circuitry - *little digital*
  - Millions of control decisions per second for years
  - An incorrect decision may permanently damage the circuit
- Poor EDA support
  - Synthesis is optimised for data processing - *big digital*
  - *Ad hoc* solutions are prone to errors and cannot be verified

Basic buck converter: Schematic



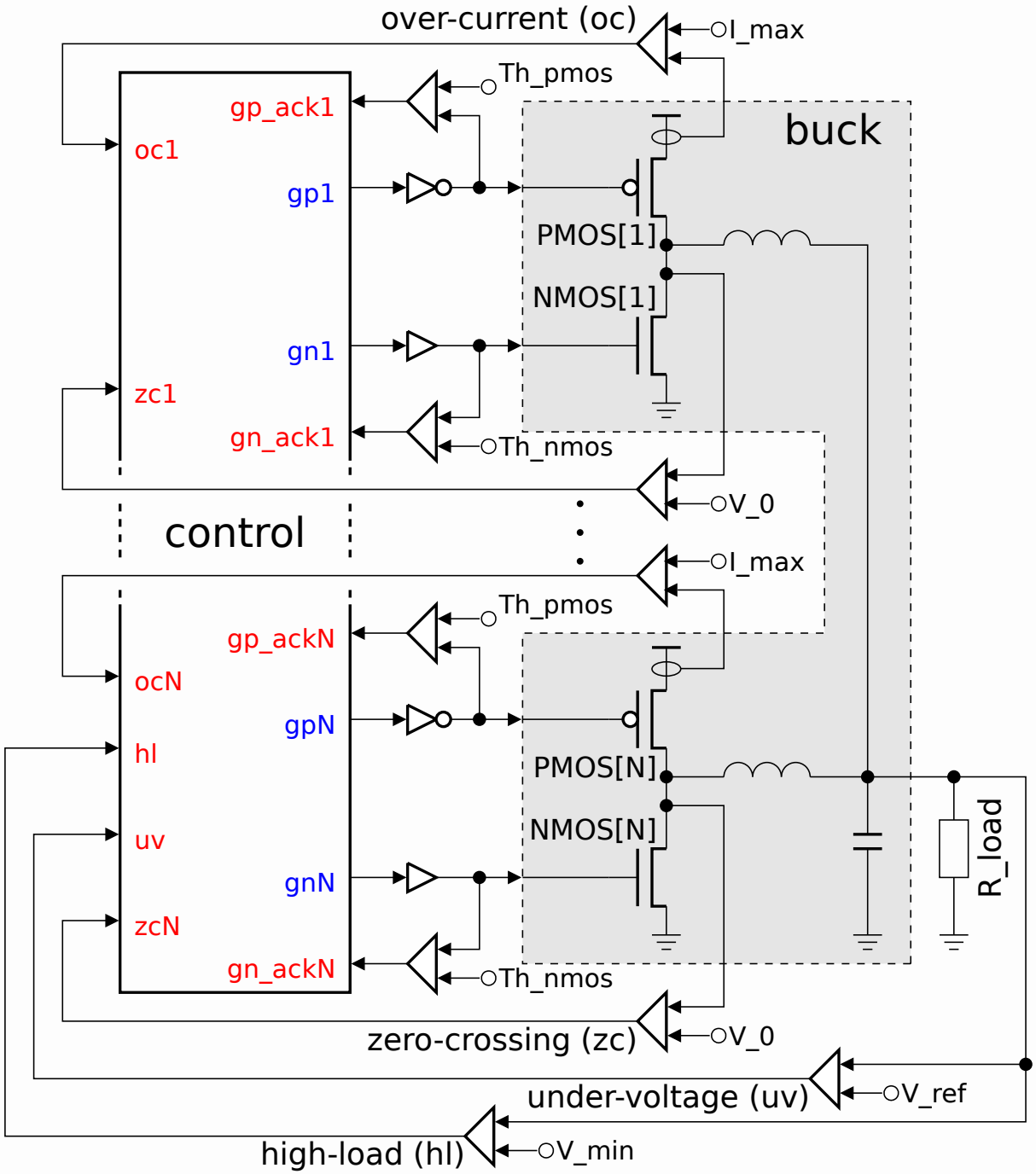
- In the textbook buck a diode is used instead of NMOS transistor

Basic buck converter: Informal specification



- **no ZC** – under-voltage without zero-crossing
- **late ZC** – under-voltage before zero-crossing
- **early ZC** – under-voltage after zero-crossing

Multiphase buck converter: Schematic



Multiphase buck converter: Informal specification

- Normal mode
  - Phases are activated sequentially
  - Phases may overlap
- High-load mode
  - All phases are activated simultaneously
- Benefits
  - Faster reaction to the power demand
  - Heat dissipation from a larger area
  - Decreased ripple of the output voltage
  - Smaller transistors and coils

Synchronous design

- Two clocks: phase activation (~5MHz) and sampling (~100MHz)
  - ☺ Easy to design (RTL synthesis flow)
  - ☹ Response time is of the order of clock period
  - ☹ Power consumed even when idle
  - ☹ Non-negligible probability of a synchronisation failure
- Manual ad hoc design to alleviate the disadvantages
  - ☹ Verification by exhaustive simulation

Asynchronous design

- Event-driven control decisions

  ☺ Prompt response (a delay of few gates)

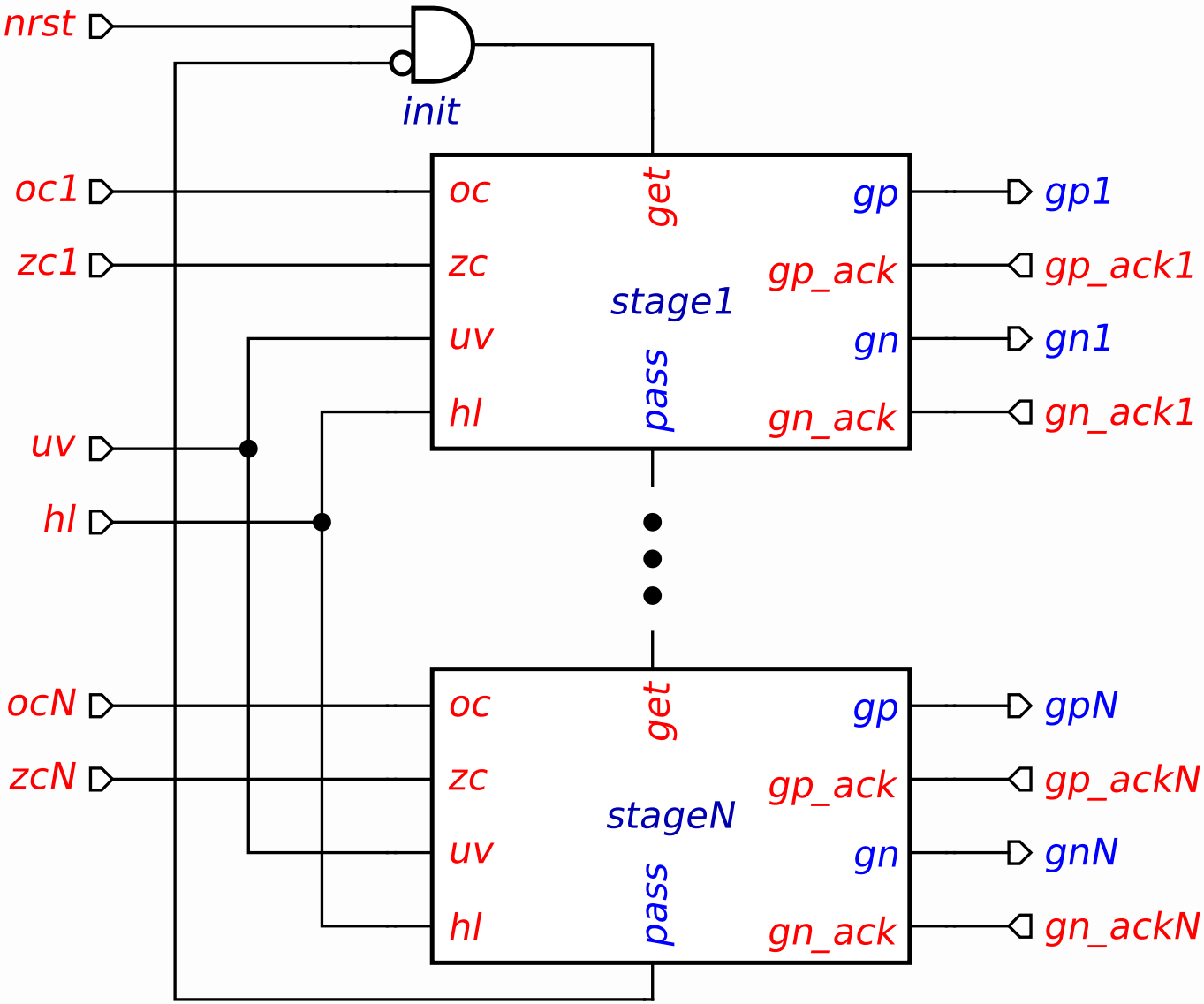
  ☺ No dynamic power consumption when the buck is inactive

  ☺ Other well known advantages

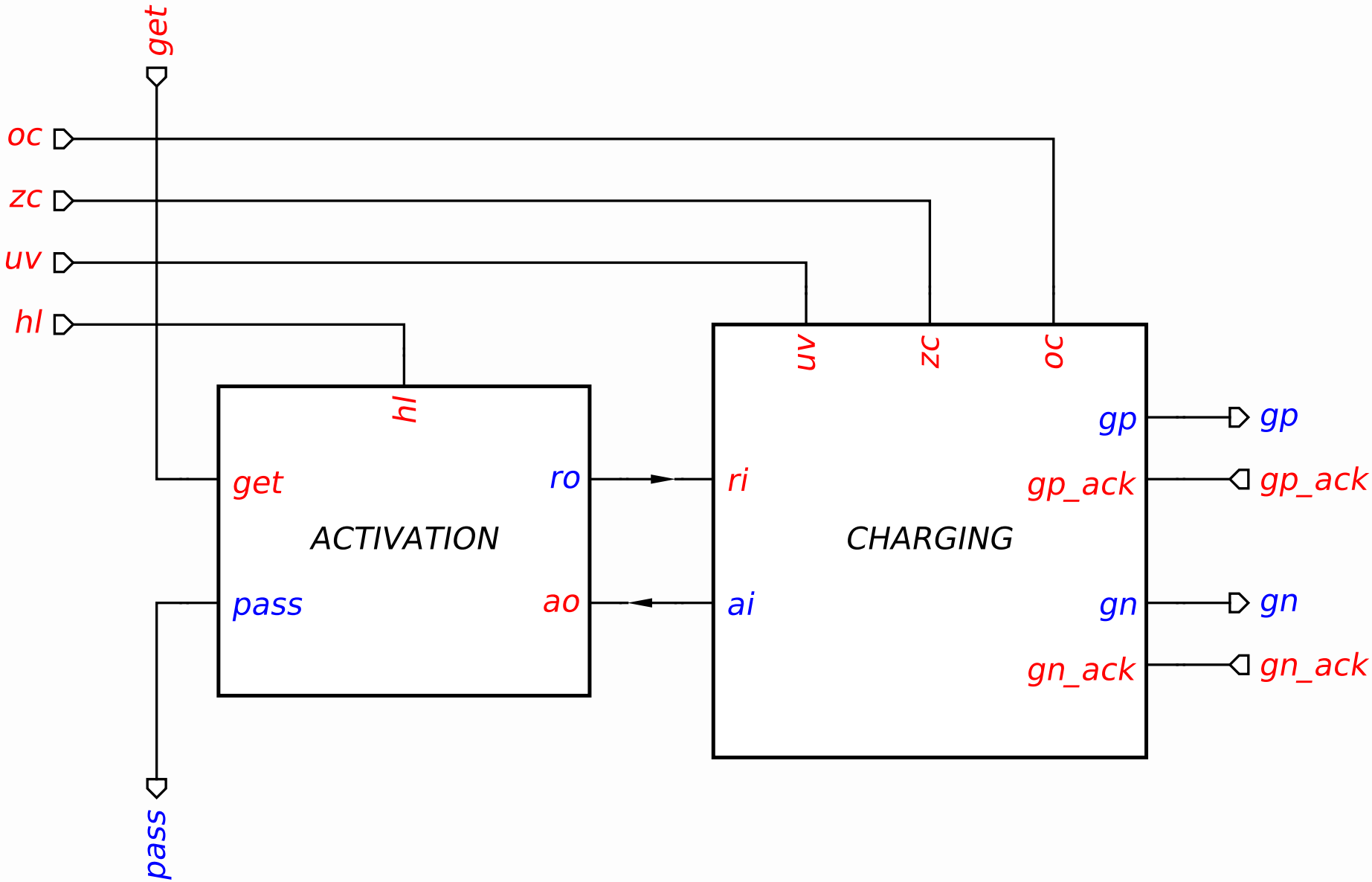
  ☹ Insufficient methodology and tool support

- Our goals
  - Formal specification of power control behaviour
  - Reuse of existing synthesis methods
  - Formal verification of the obtained circuits
  - Demonstrate new advantages for power regulation (power efficiency, smaller coils, ripple and transient response)

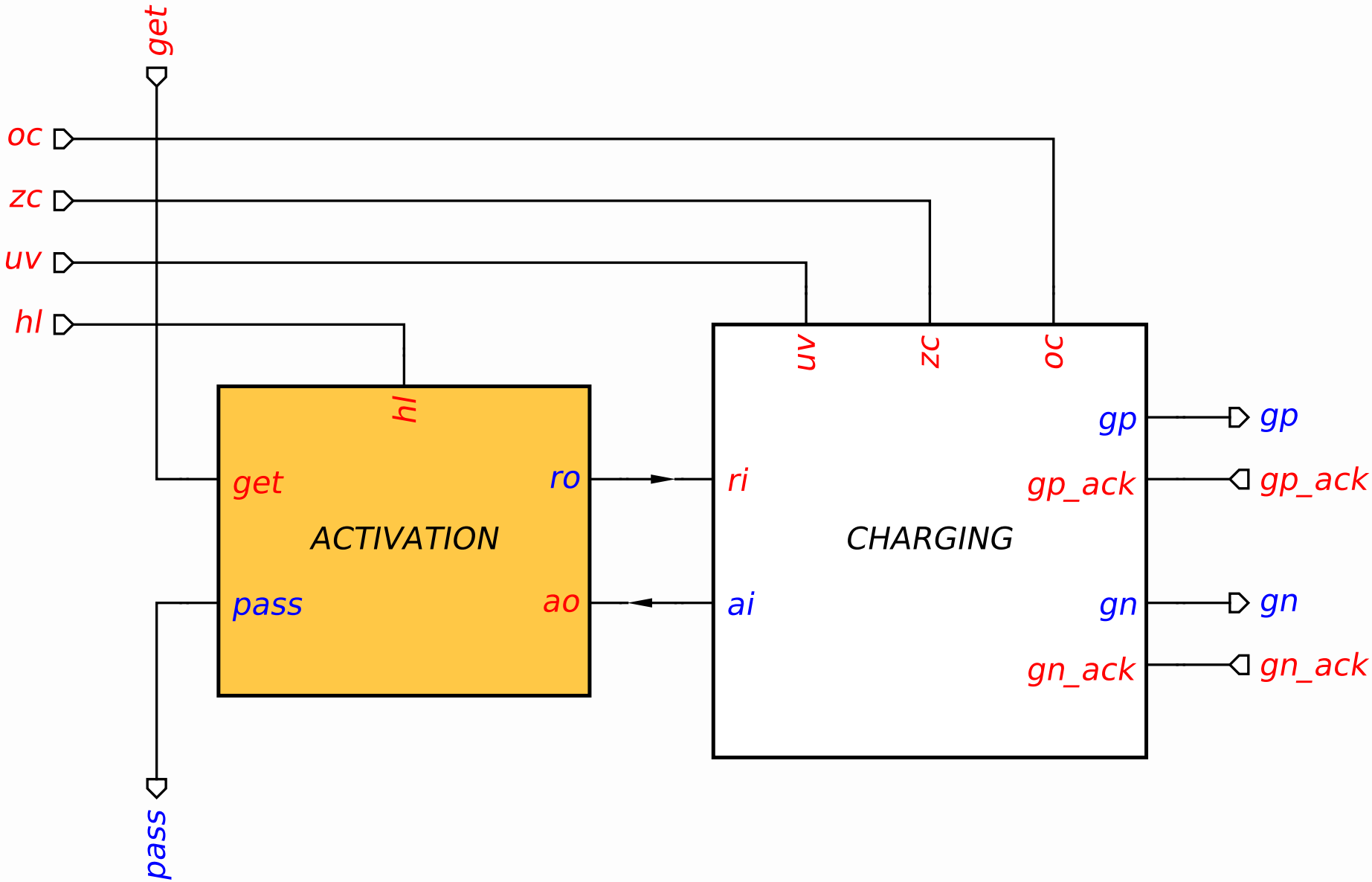
High-level architecture: Token ring



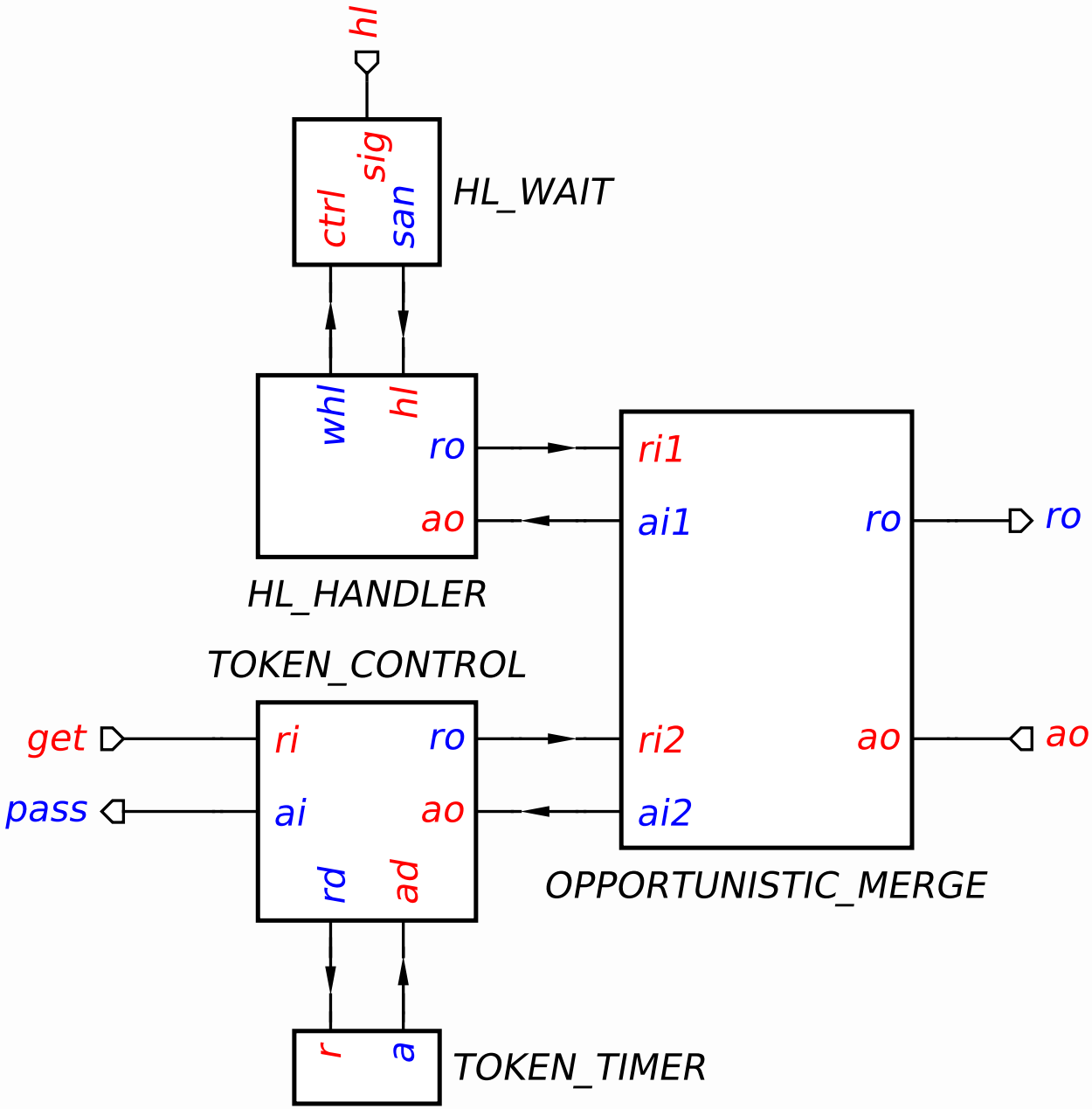
High-level architecture: Stage



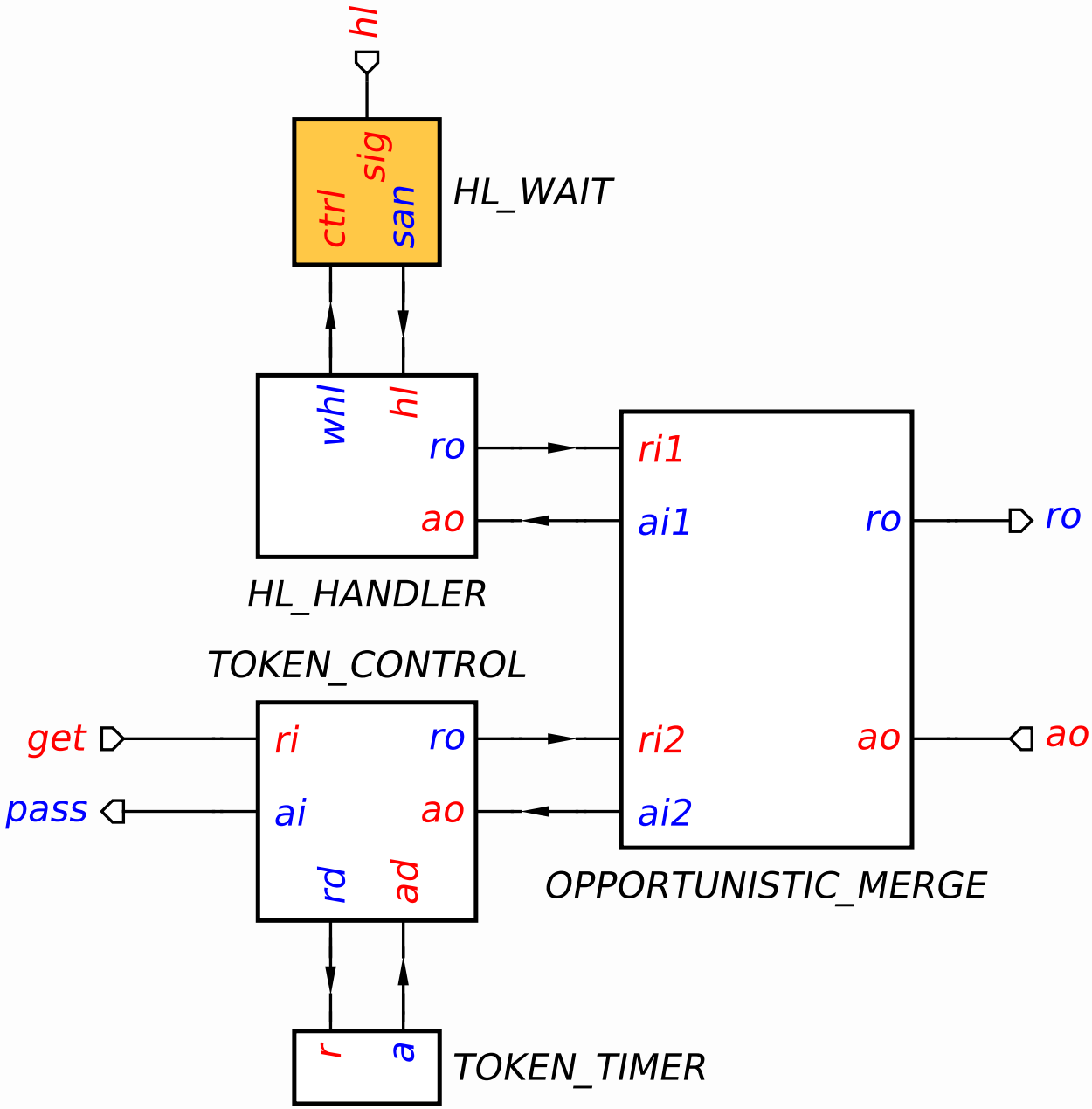
High-level architecture: Activation



High-level architecture: Activation

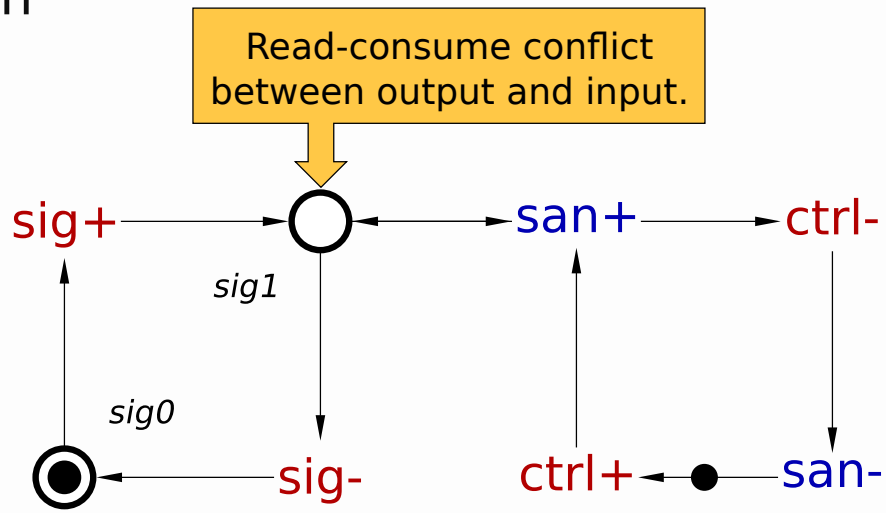


WAIT element

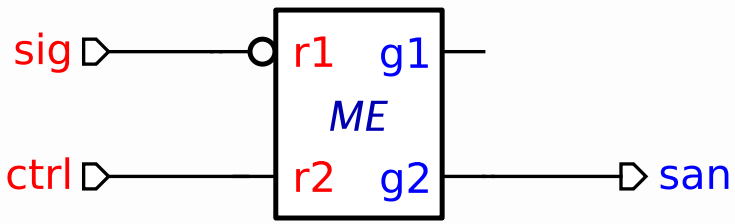


WAIT element

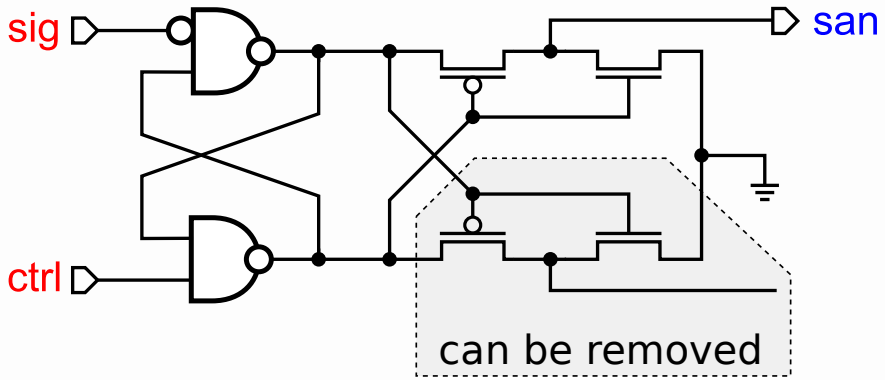
- STG specification



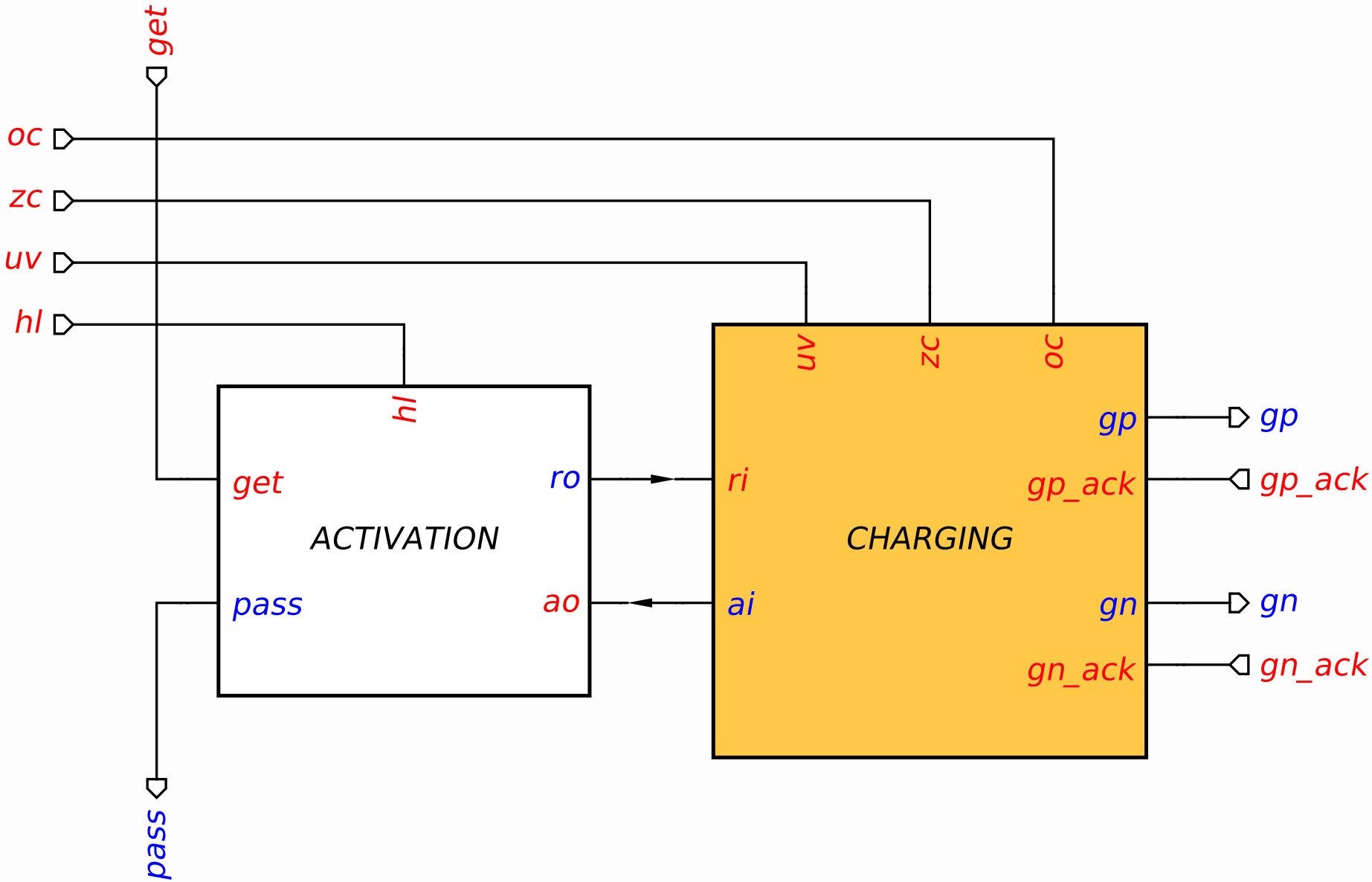
- ME-based solution



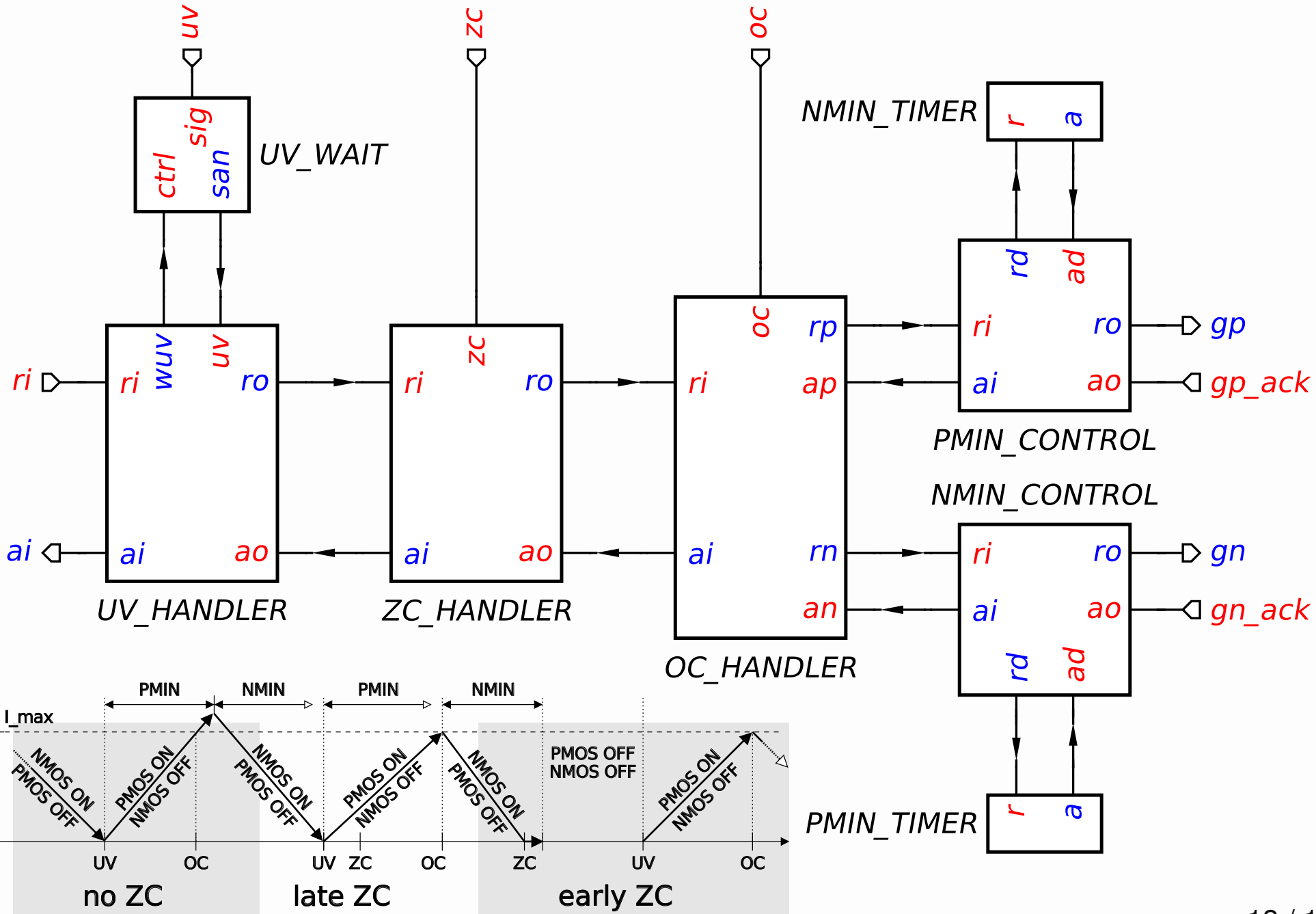
- Gate-level implementation



High-level architecture: Charging



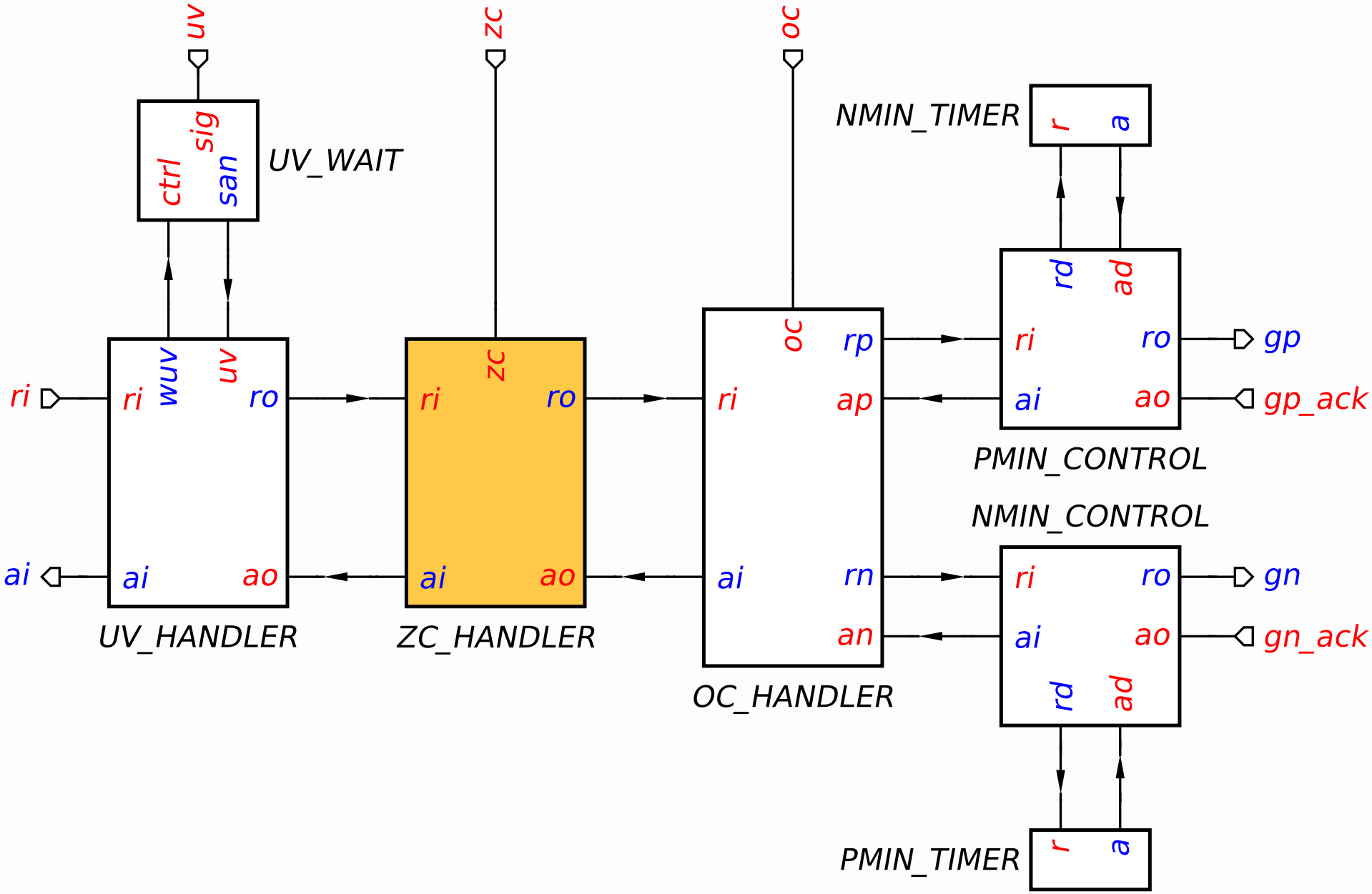
High-level architecture: Charging



Synthesis flow

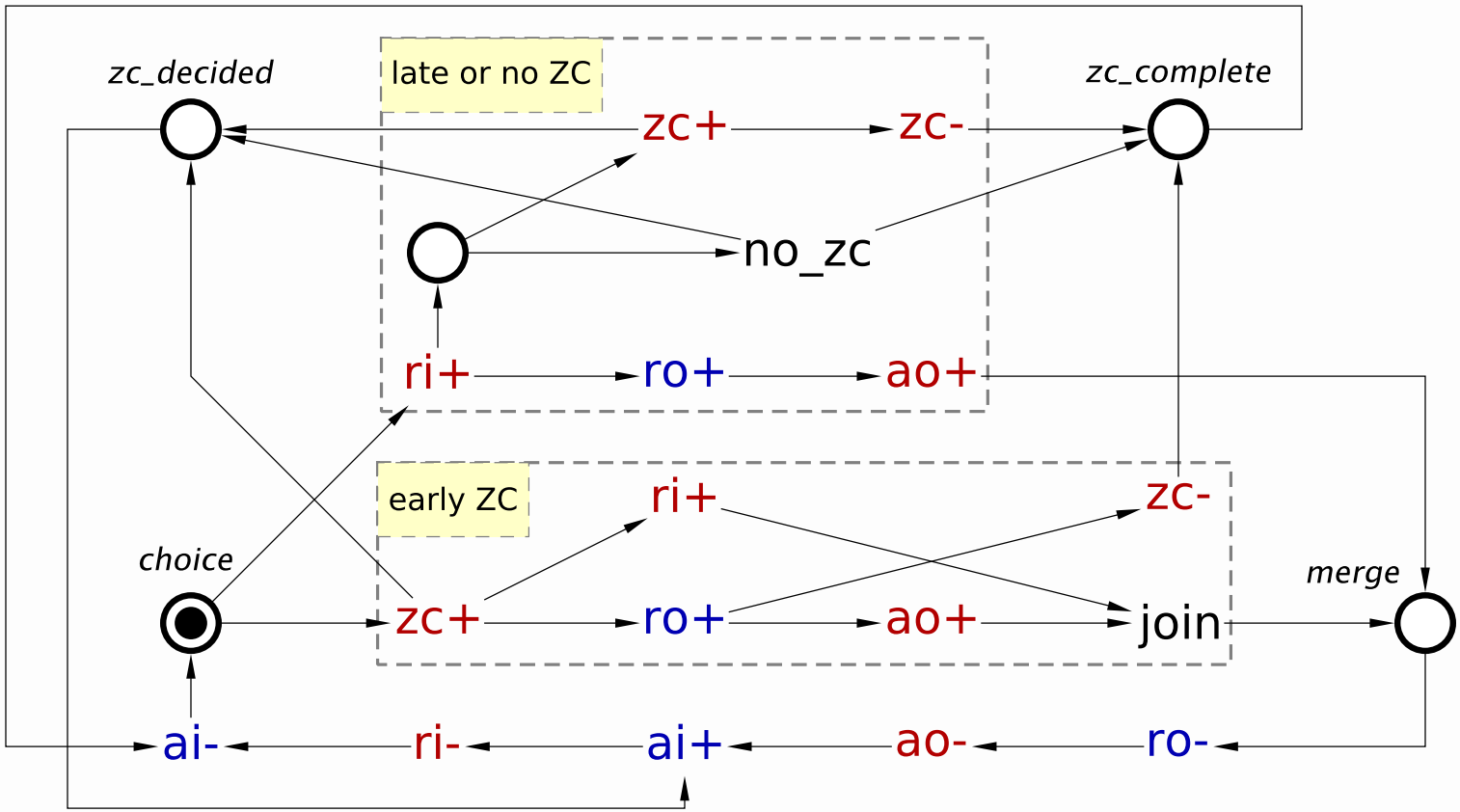
- Manual decomposition of the system into modules
  - To create formal specification from informal requirements (feedback loop with engineers)
  - To simplify specification and synthesis
  - Some modules are reusable
  - Some modules (WAIT and OPPORTUNISTIC MERGE) are potential standard components
- Each component is specified using STGs
- Automatic synthesis into speed-independent circuits

ZC_HANDLER module



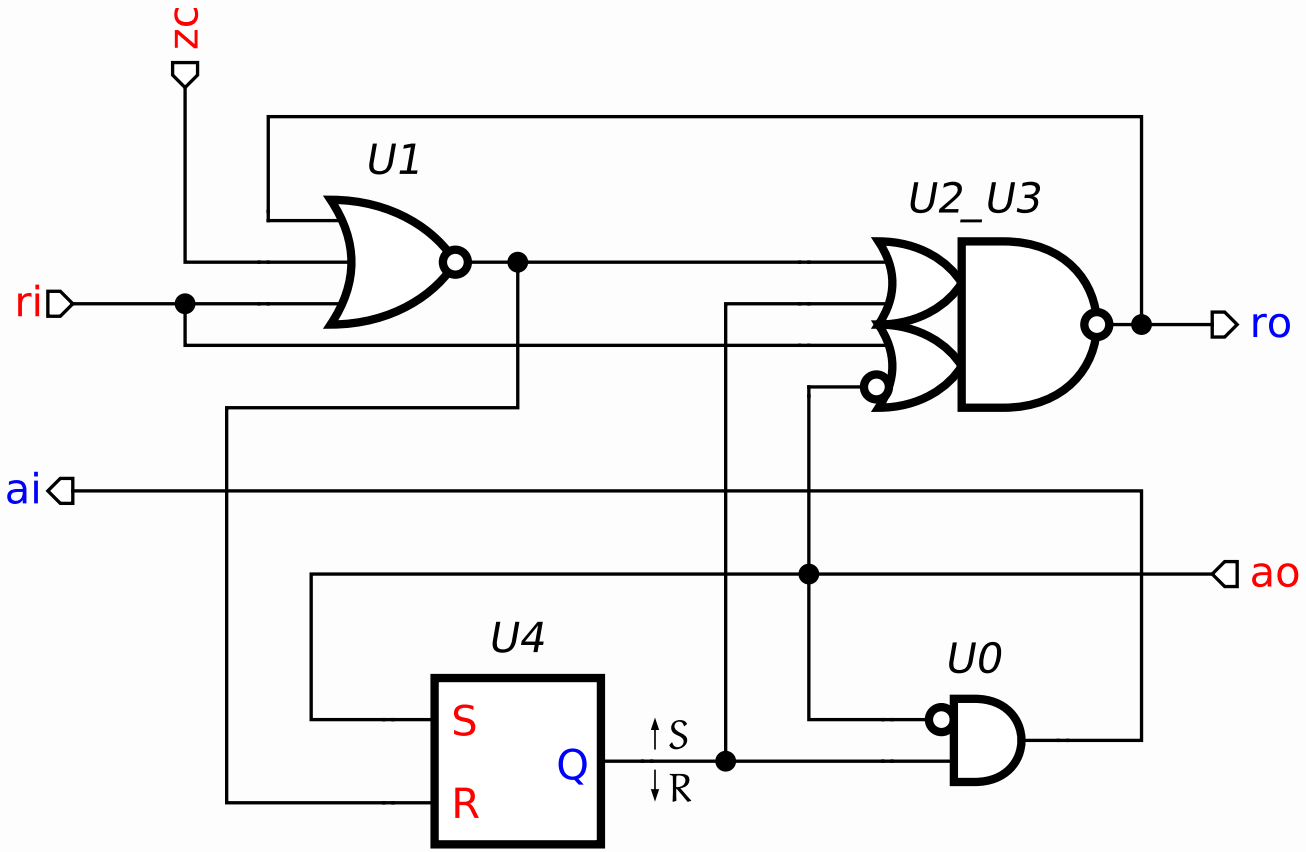
ZC_HANDLER module

- STG specification



ZC_HANDLER module

- Speed-independent implementation



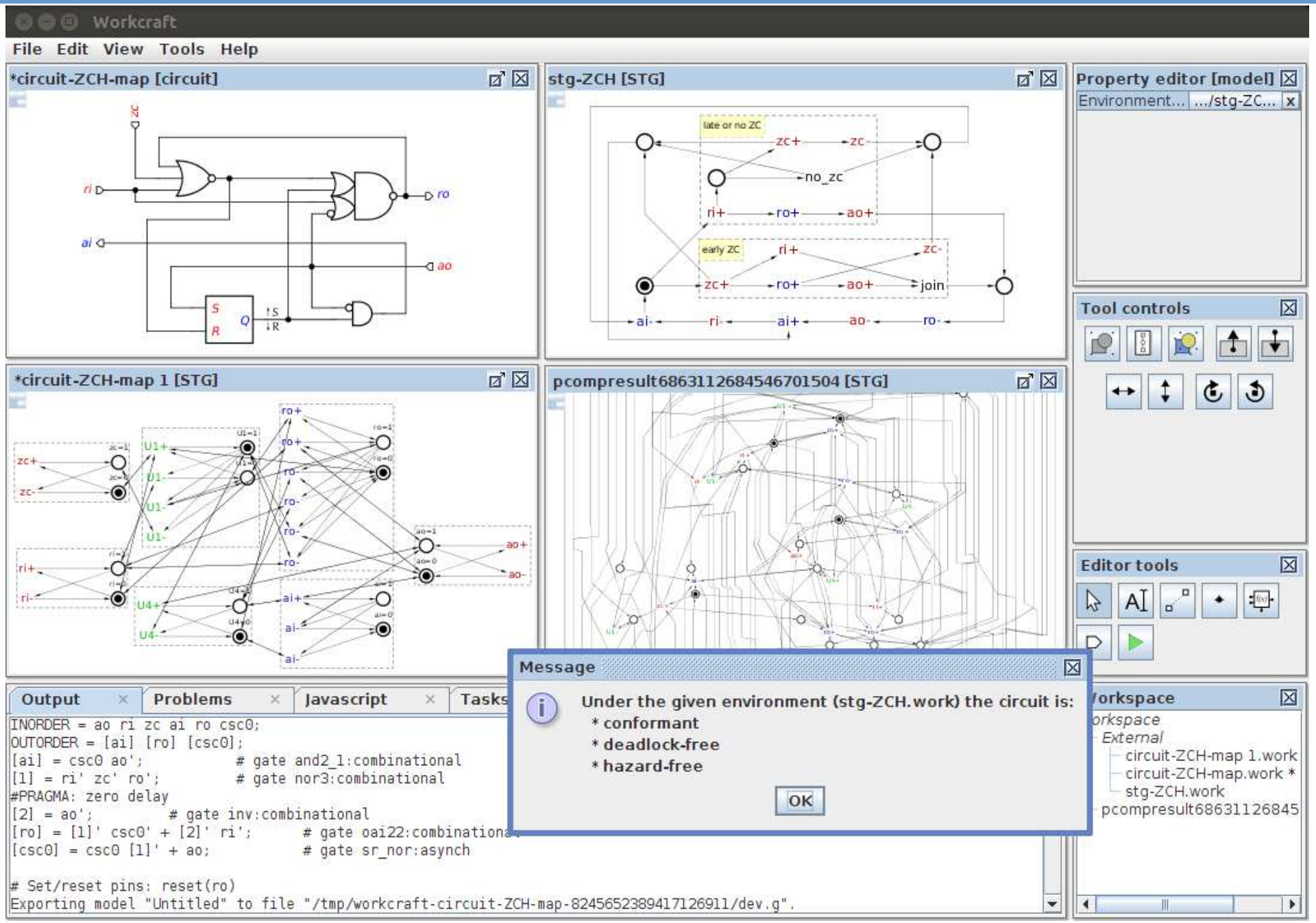
Formal verification

- STG verification
  - All standard speed-independence properties
  - PMOS and NMOS are never ON simultaneously (no short circuit)
  - Some timers are used in a mutually exclusive way and can be shared
- Circuit verification
  - Conforms to the environment
  - Deadlock-free and hazard-free under the given environment

Tool support: WORKCRAFT

- Framework for *interpreted graph models* (STGs, circuits, FSMs, dataflow structures, etc.)
  - Interoperability between models
  - Elaborated GUI
- Includes many backend tools
  - PETRIFY – STG and circuit synthesis, BDD-based
  - PUNF – STG unfolder
  - MPSAT – unfolding-based verification and synthesis
  - PCOMP – parallel composition of STGs

Tool support: WORKCRAFT



Conclusions

- Fully asynchronous design of multiphase buck controller
  - Quick response time: few gate delays, all mutexes are outside the critical path
  - Reliable: no synchronisation failures
- Design flow is automated to large extent
  - Automatic logic synthesis
  - Formal verification at the STG and circuit levels
- New standard components (WAIT and OPPORTUNISTIC MERGE)
- Future work
  - Measurements!
  - Development of WORKCRAFT to support hierarchical design
  - Co-simulation and co-verification of digital/analogue circuits
  - Better integration with the Synopsys and Cadence flows