Benefits of Asynchronous Control for Analog Electronics: Multiphase Buck Case Study

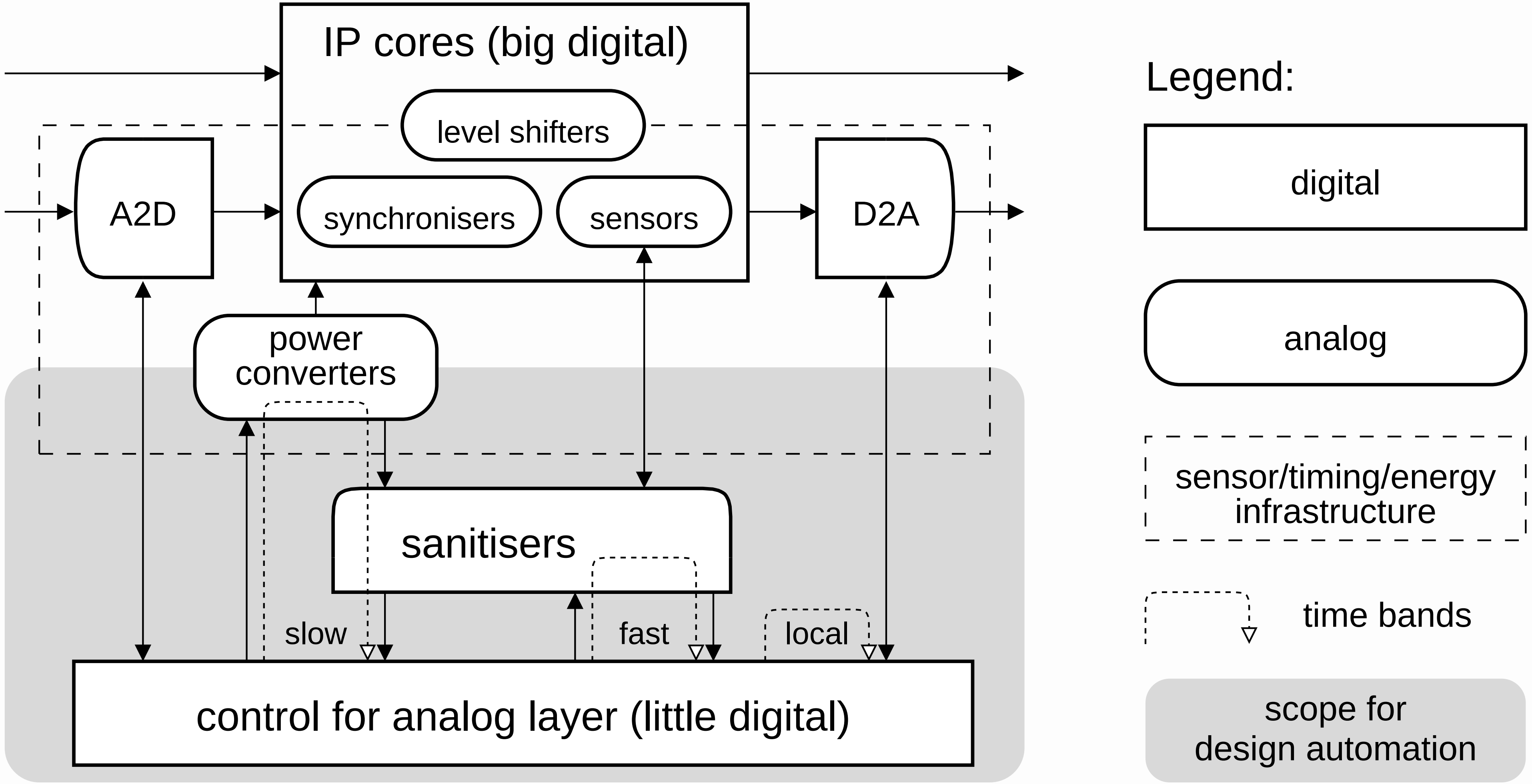
Danil Sokolov, Vladimir Dubikhin, Victor Khomenko,
David Lloyd[★], Andrey Mokhov, Alex Yakovlev

Newcastle University, UK   [★] *Dialog Semiconductor, UK*

Outline

- Motivation
- Basic buck converter
- A4A design flow
- A2A components
- WORKCRAFT design automation
- Design of multiphase buck converter
- Experimental results

Motivation: Little digital control for analog electronics

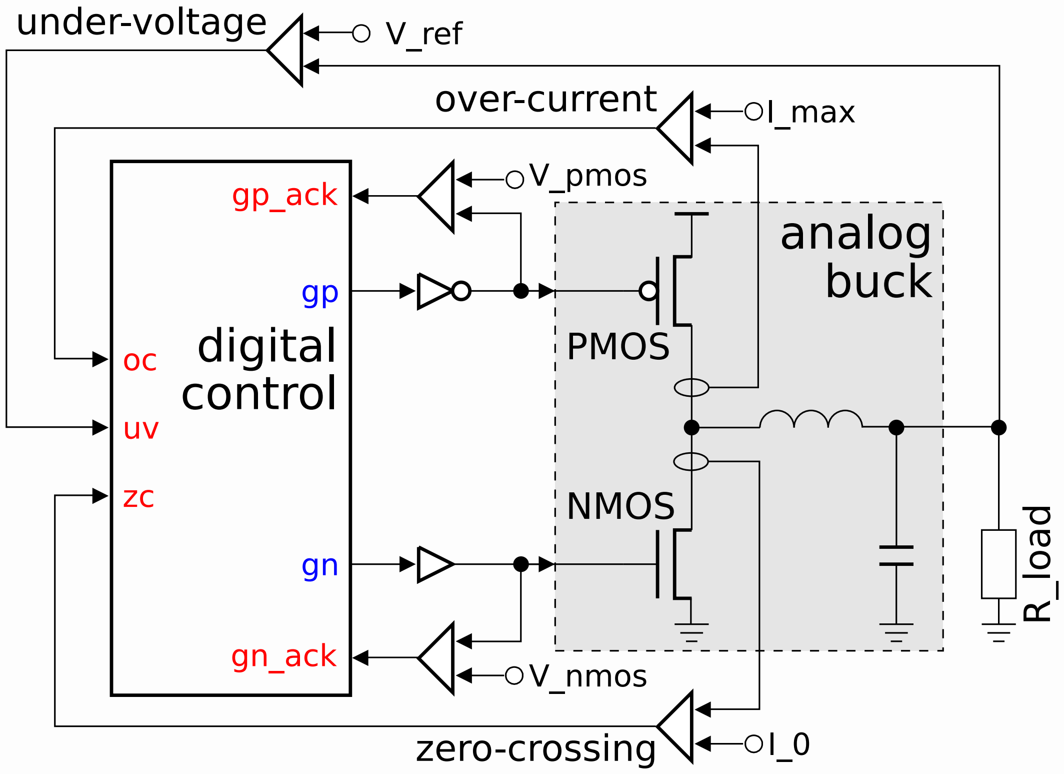


- Analog and digital electronics are becoming more intertwined
- Analog domain becomes more complex and itself needs digital control

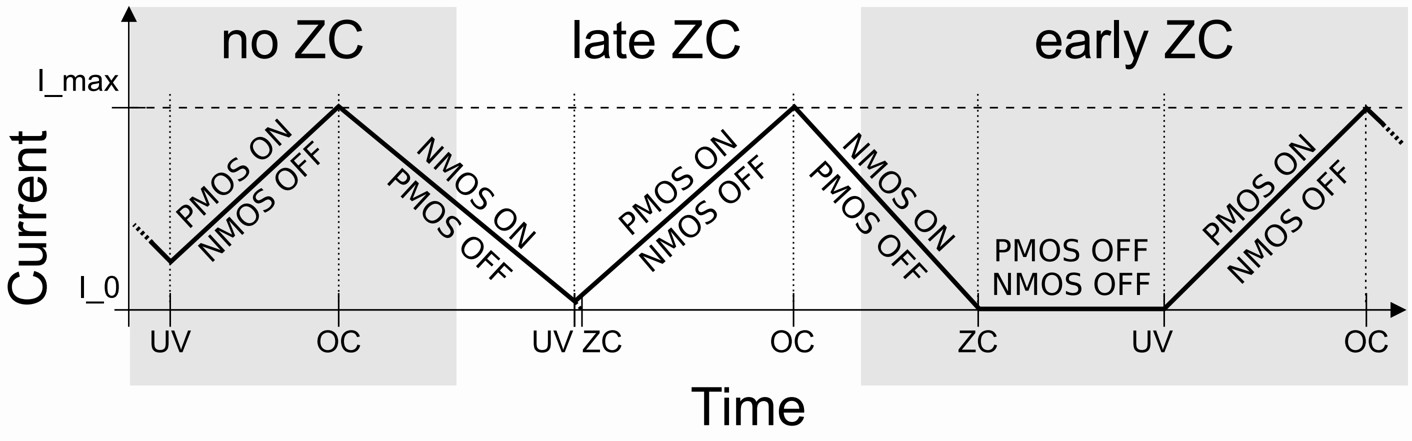
Motivation: Power electronics context

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

Basic buck converter



Phase diagram specification:



Buck conditions:

- under-voltage (UV)
- over-current (OC)
- zero-crossing (ZC)

Operating modes:

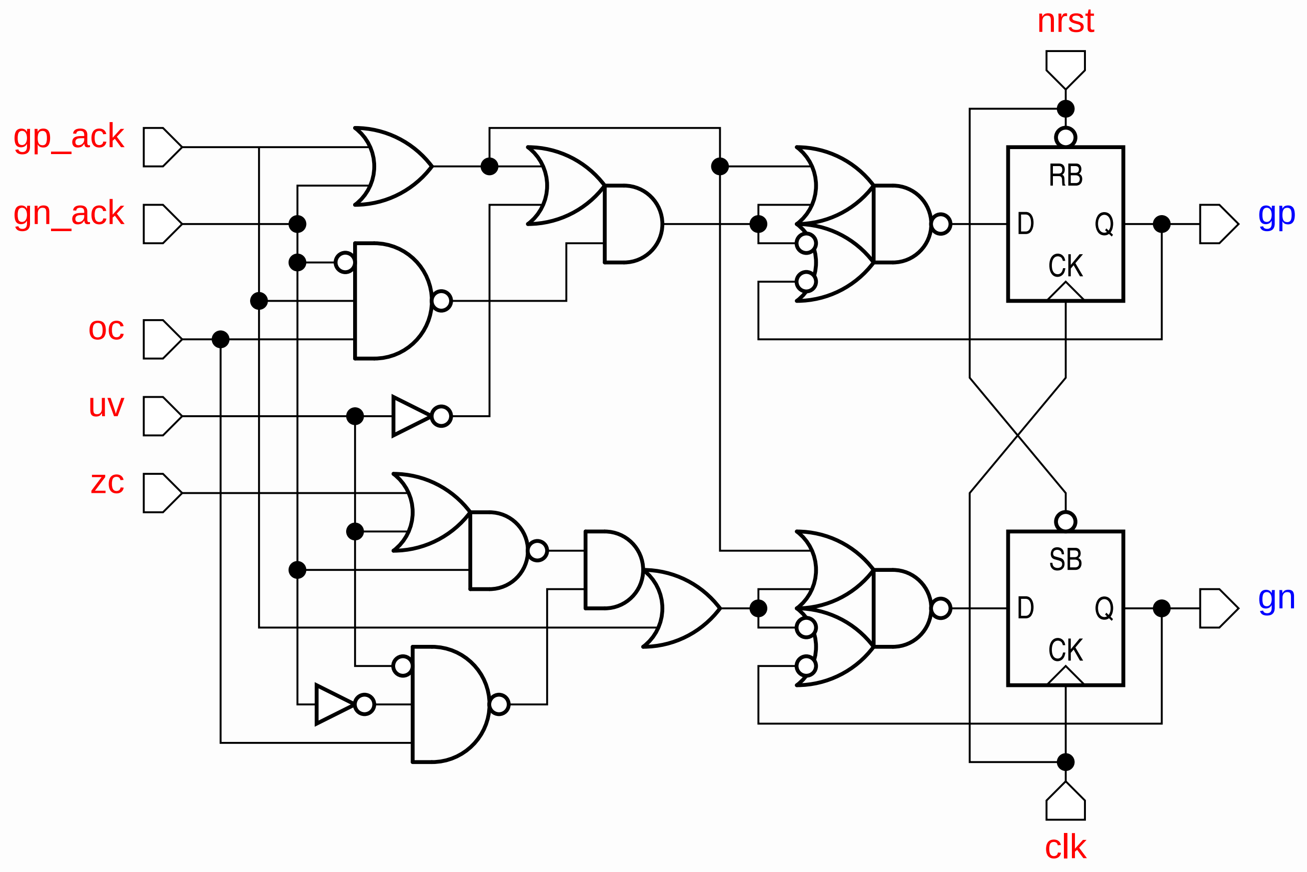
- no zero-crossing
- late zero-crossing
- early zero-crossing

Basic buck converter: Synchronous design

RTL description:

```
module control (clk, nrst, oc, uv, zc, gp_ack, gn_ack, gp, gn);
  input clk, nrst, uv, oc, zc, gp_ack, gn_ack;
  output reg gp, gn;
  always @(posedge clk or negedge nrst) begin
    if (nrst == 0) begin
      gp <= 0; gn <= 1;
    end else case ({gp_ack, gn_ack})
      2'b00: if (uv == 1) gp <= 1; else if (oc == 1) gn <= 1;
      2'b10: if (oc == 1) gp <= 0;
      2'b01: if (uv == 1 || zc == 1) gn <= 0;
    endcase
  end
endmodule
```

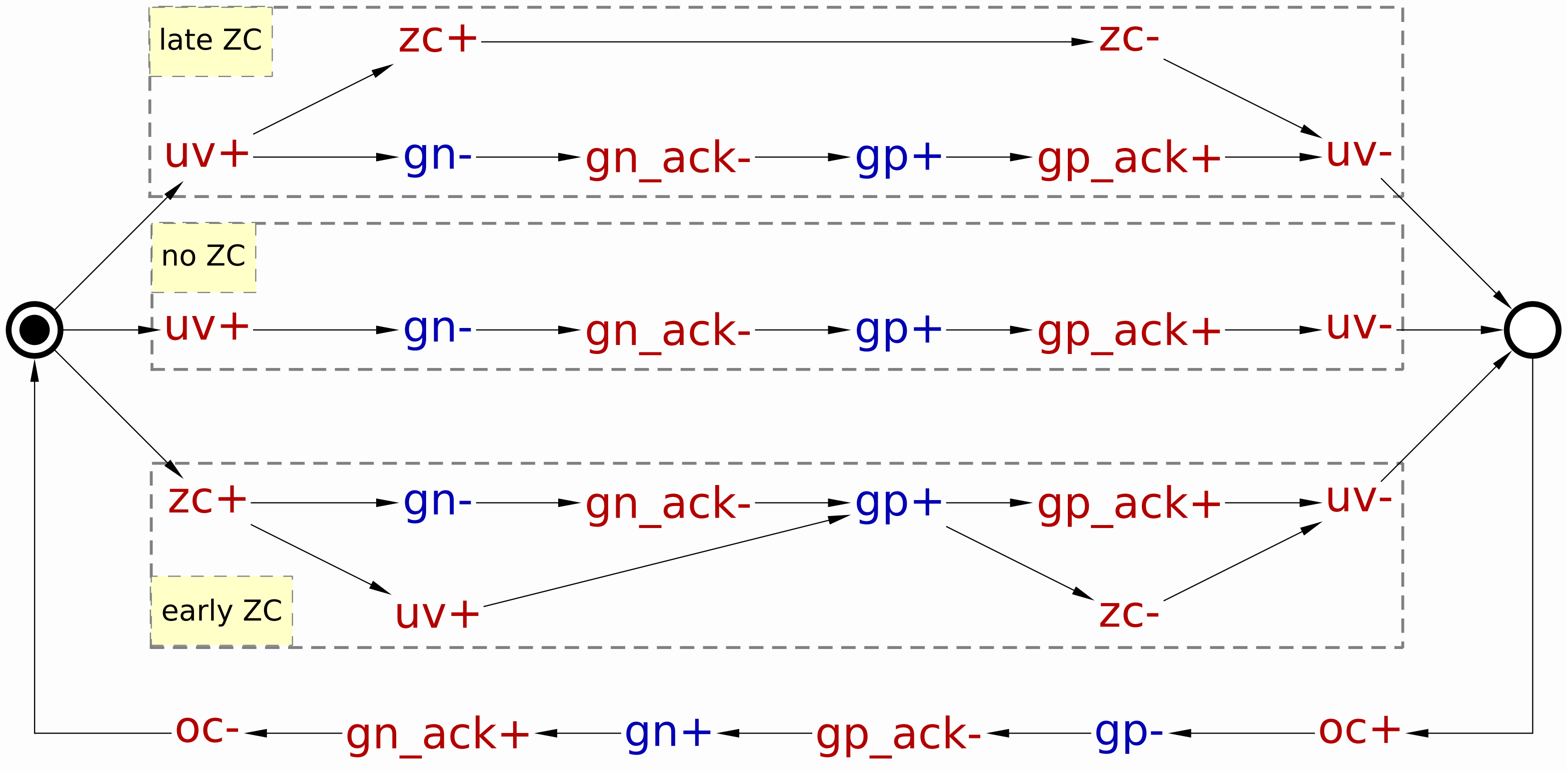
Design Compiler synthesis result:



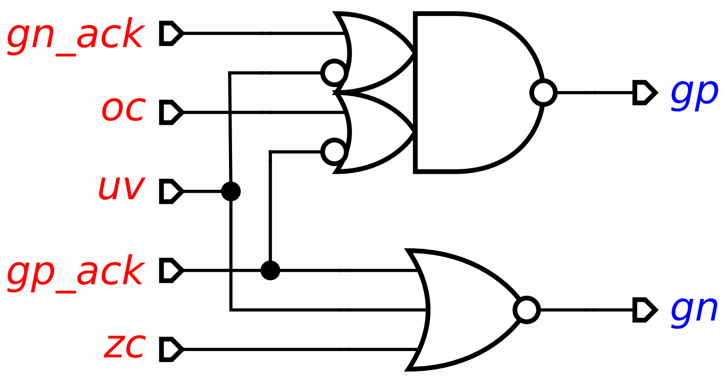
- Asynchronous inputs need to be synchronised (latency penalty)
- If clock is slow, the control is unresponsive to the buck changes
- If clock is fast, it burns energy when the buck is inactive

Basic buck converter: Asynchronous design

Formal specification:

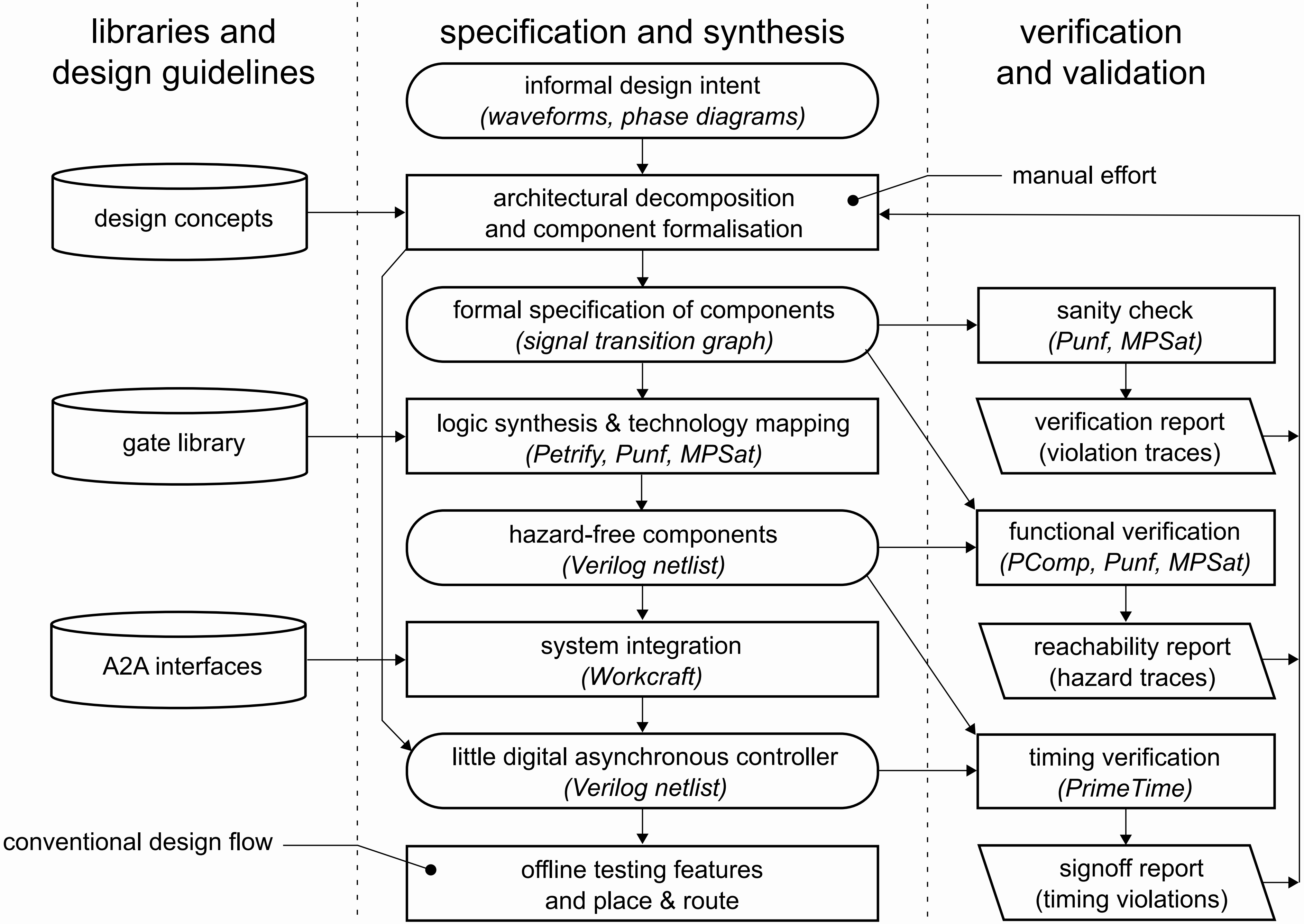


Hazard-free implementation:



- Formal specification using Signal Transition Graphs – similar to Petri nets
- Verifiable hazard-free implementation – correct for any gate delays
- Prompt reaction time to the buck changes – latency of a complex gate

Asynchronous control for Analog electronics (A4A) design flow

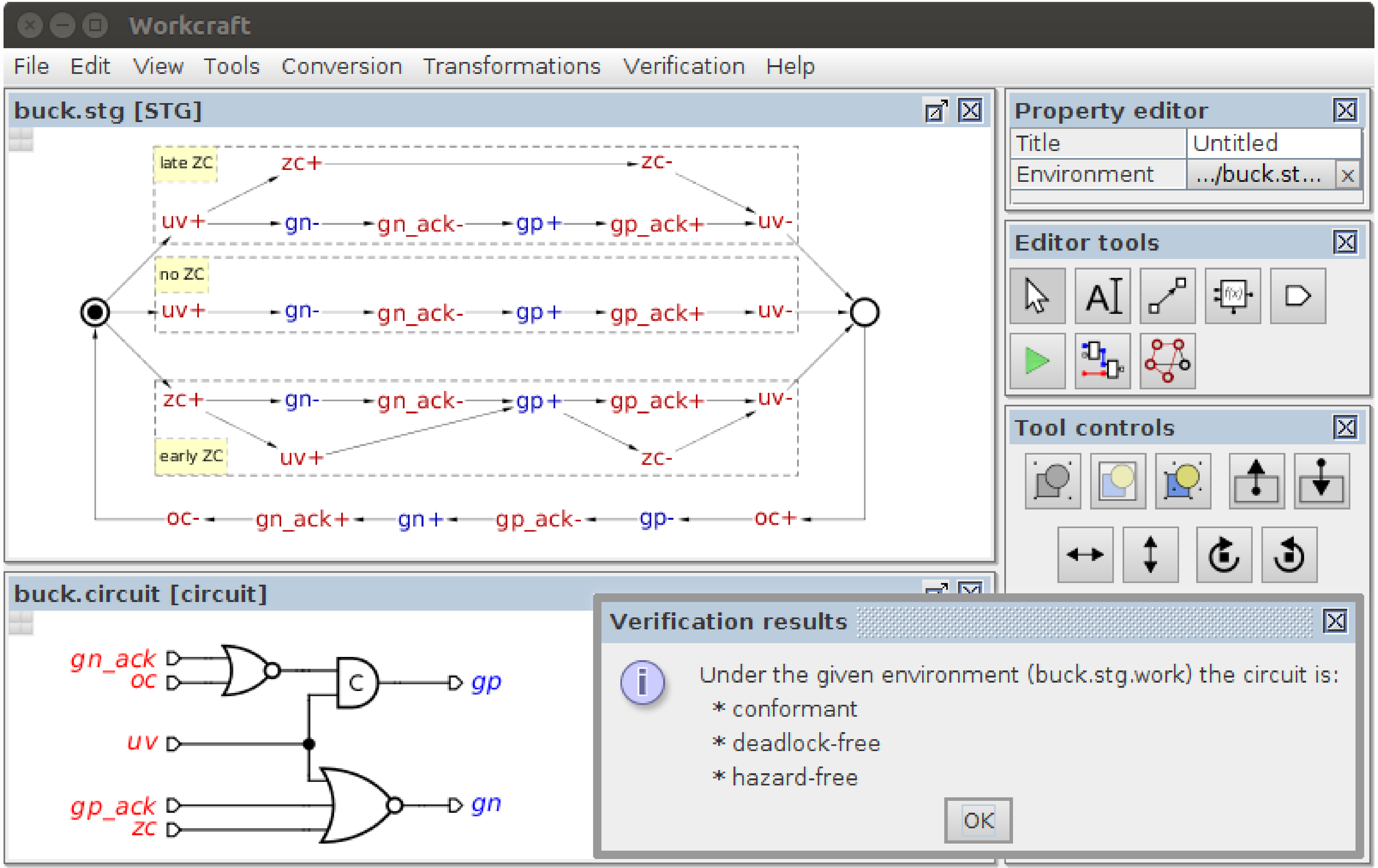


libraries and design guidelines
specification and synthesis
verification and validation
informal design intent (waveforms, phase diagrams)
design concepts
architectural decomposition and component formalisation
manual effort
formal specification of components (signal transition graph)
sanity check (Punf, MPSat)
gate library
logic synthesis & technology mapping (Petrify, Punf, MPSat)
verification report (violation traces)
hazard-free components (Verilog netlist)
functional verification (PComp, Punf, MPSat)
A2A interfaces
system integration (Workcraft)
reachability report (hazard traces)
little digital asynchronous controller (Verilog netlist)
timing verification (PrimeTime)
conventional design flow
offline testing features and place & route
signoff report (timing violations)

Library of analog-to-asynchronous (A2A) interface components

- Interface analog world of *dirty* signals
- Provide hazard-free *sanitised* digital signals
- Basic A2A interface components
  - **WAIT** / **WAIT0** – wait for analog input to become high / low and latch it until explicit release signal
  - **RWAIT** / **RWAIT0** – modification of WAIT / WAIT0 with a possibility to persistently cancel the waiting request
  - **WAIT01** / **WAIT10** – wait for a rising / falling edge
- Advanced A2A interface components
  - **WAIT2** – combination of WAIT and WAIT0 to wait for high and low input values, one after the other
  - **WAITX** – arbitrate between two non-persistent analog inputs
  - **WAITX2** – behaves as WAITX in the rising phase and as WAIT0 in the falling phase

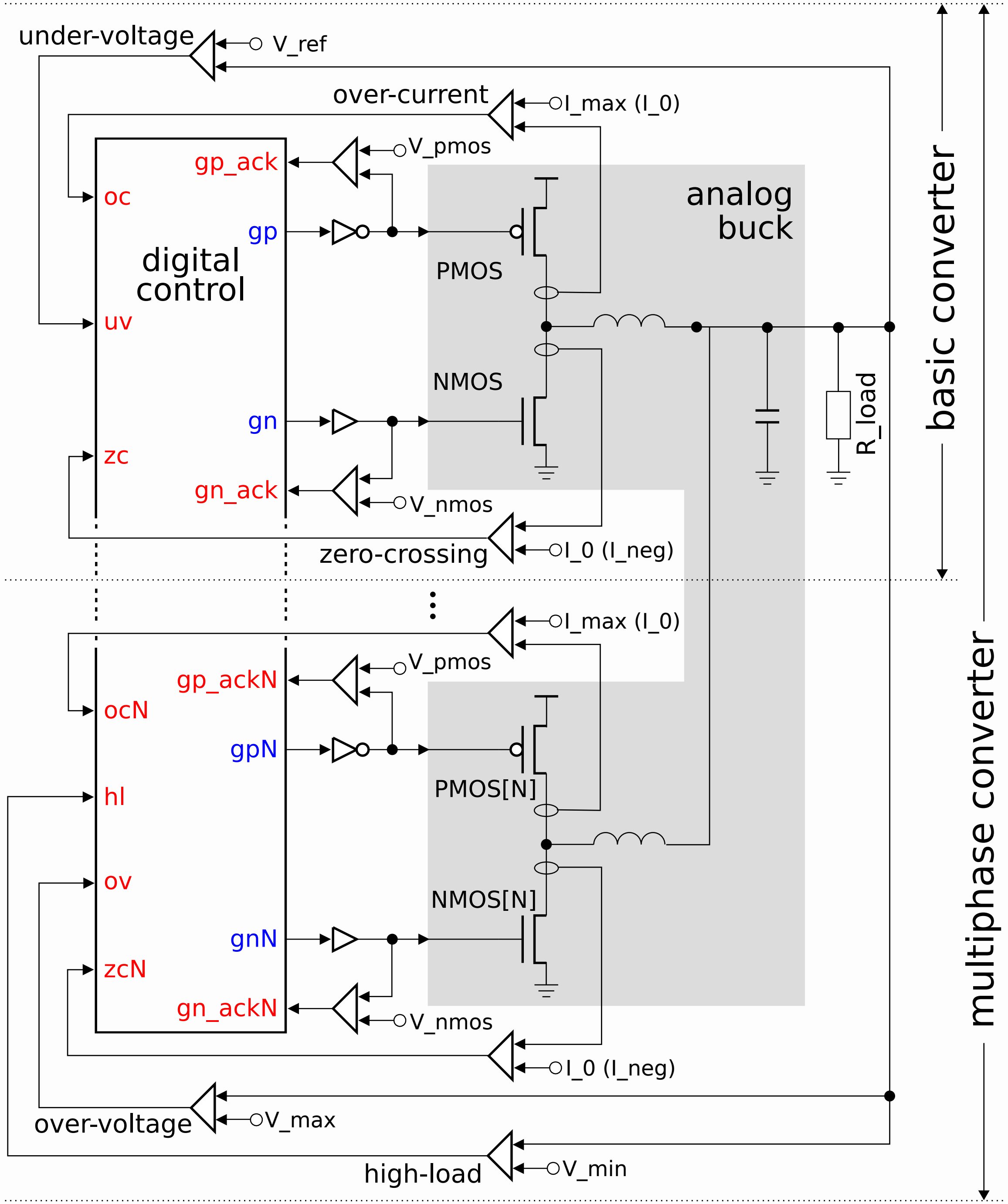
WORKCRAFT design automation – http://workcraft.org/



Under the given environment (buck.stg.work) the circuit is:

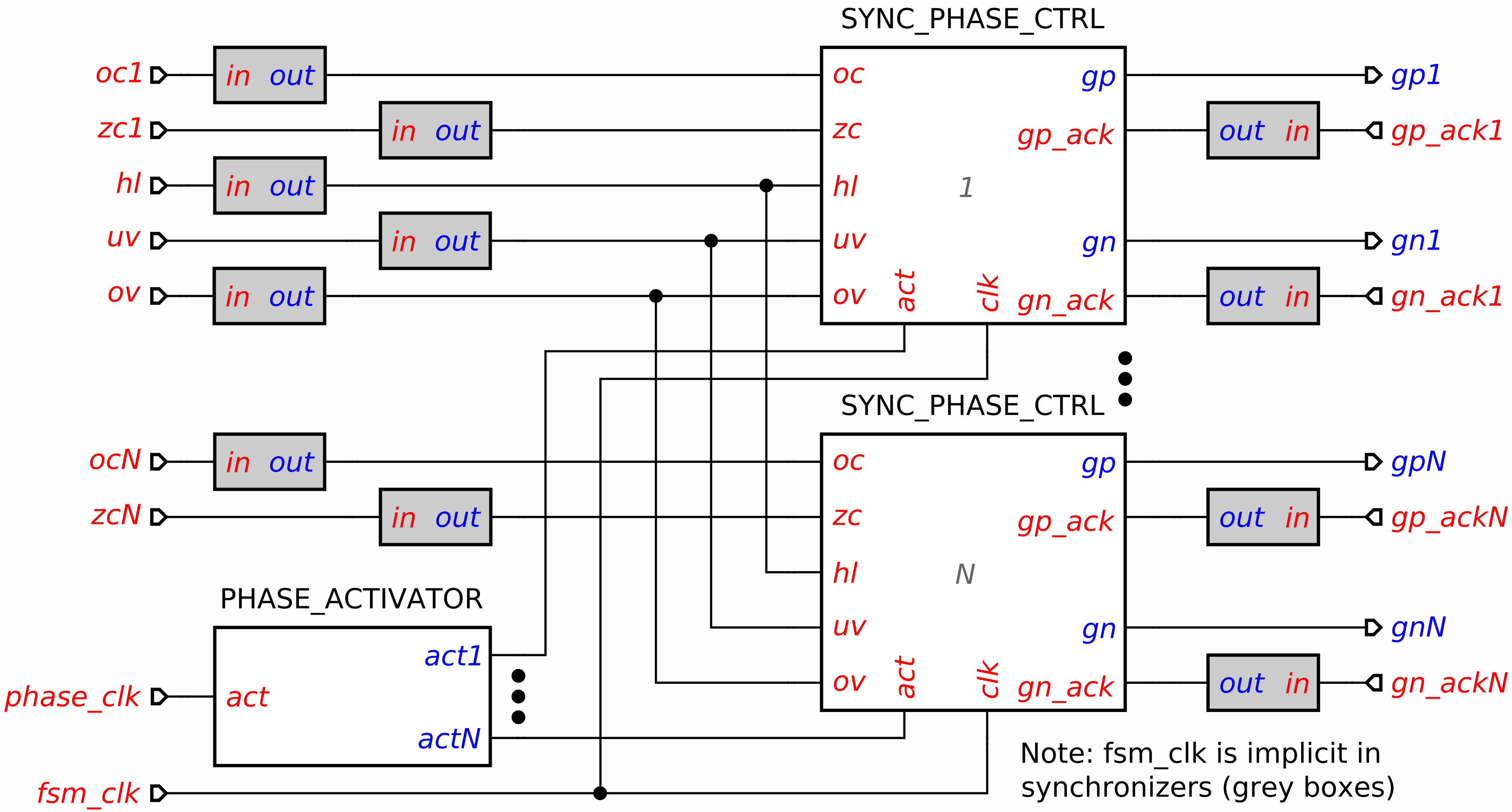
- conformant
- deadlock-free
- hazard-free

Multiphase buck converter



- Phases – pairs of power regulating transistors
  - Each phase operates as a basic buck
  - Phases are activated sequentially
  - Active phases may overlap
- More operating modes
  - Over-voltage (OV) – sink energy excess
  - High-load (HL) – boost the power
- Transistor min ON times
  - PMOS transistor: PMIN delay
  - NMOS transistor: NMIN delay
  - PMOS at first cycle: PMIN+PEXT delay

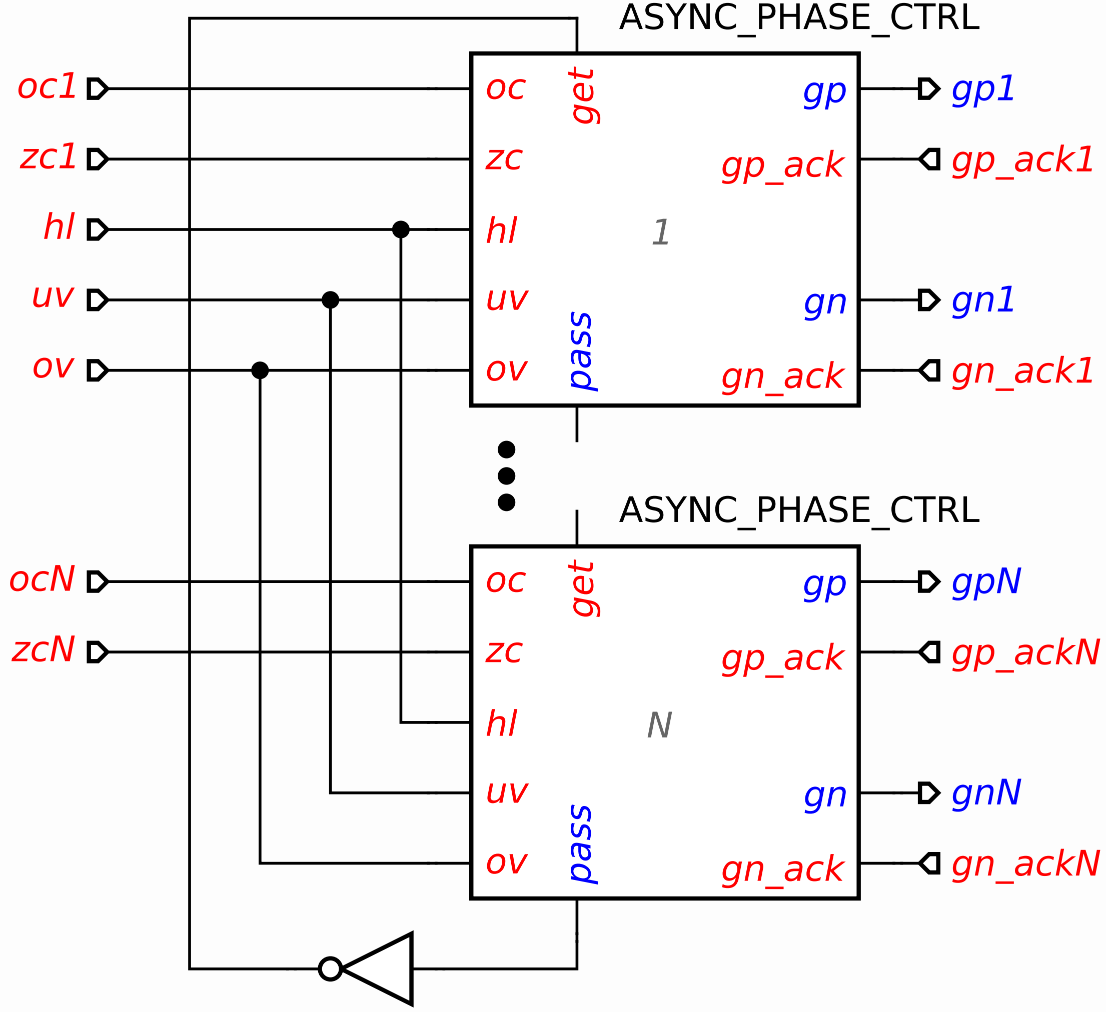
Multiphase buck converter: Synchronous control



Note: fsm_clk is implicit in synchronizers (grey boxes)

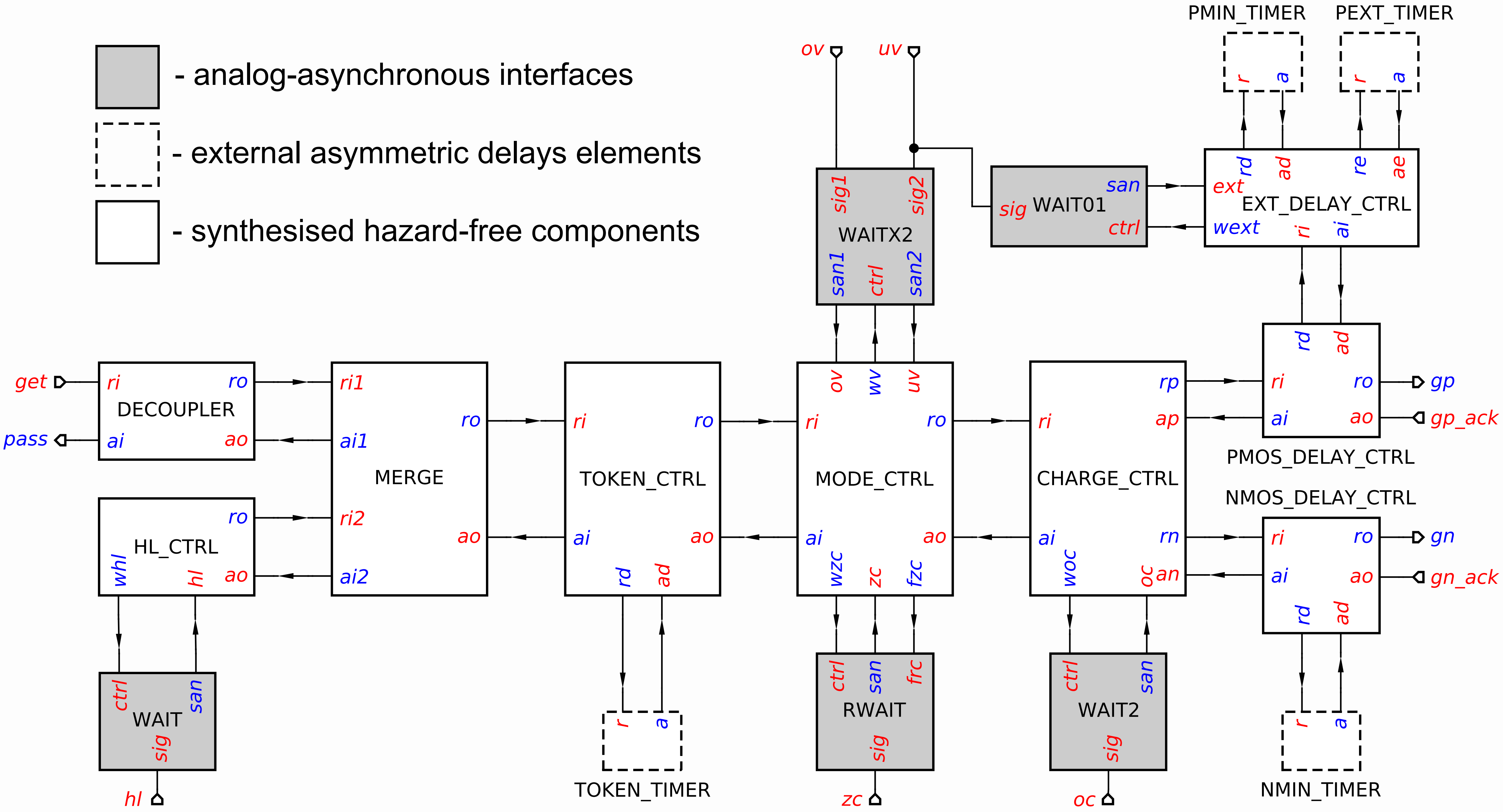
- Two clocks: phase activation (slow) and sampling (fast)
- Need for multiple synchronizers (grey boxes) – latency overheads, risk of metastability
- Conventional RTL design flow for `PHASE_ACTIVATOR` and `SYNC_PHASE_CTRL` components

Multiphase buck converter: Asynchronous control

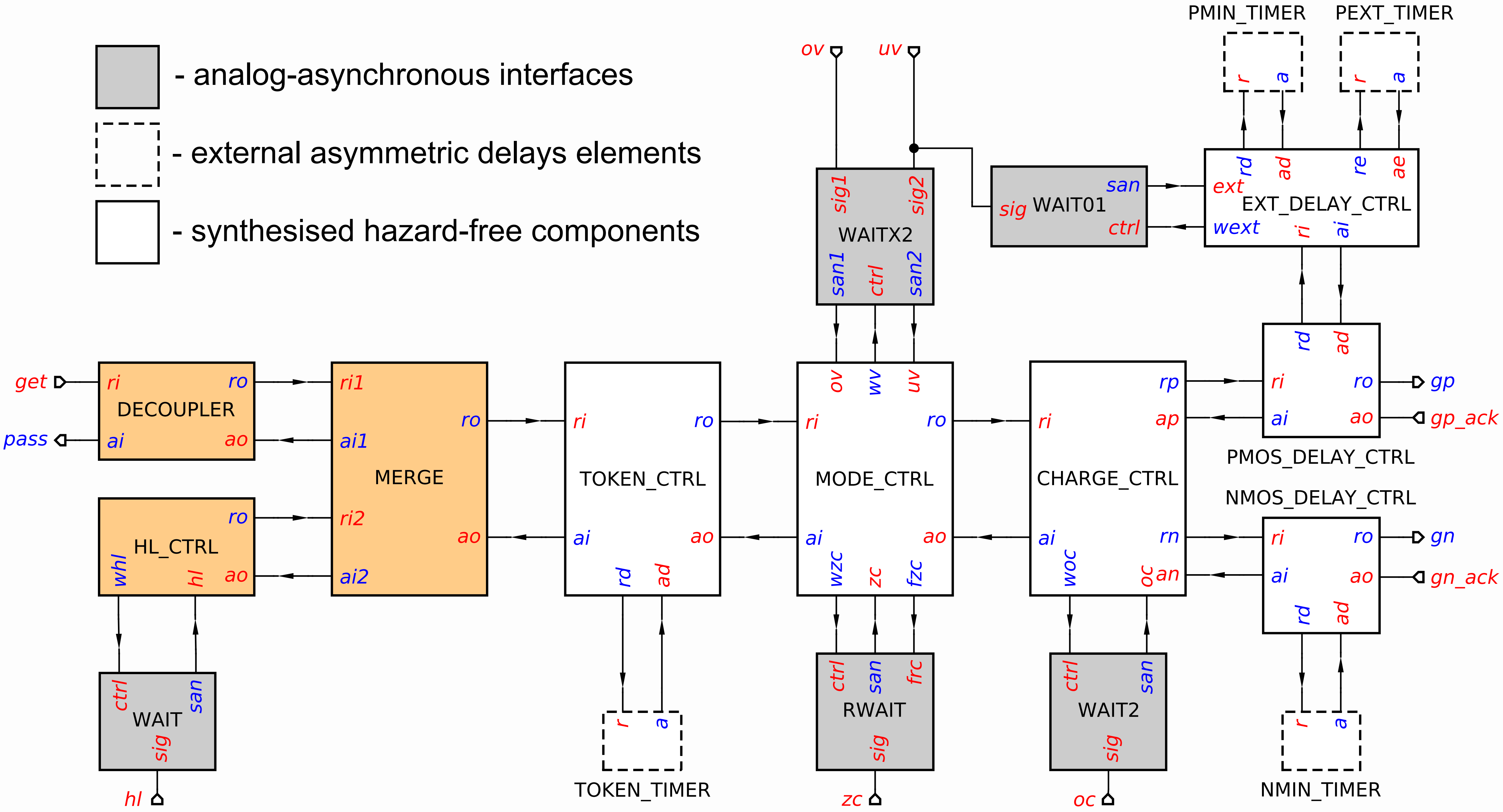


- Token ring architecture, no need for phase activation clock
- No need for synchronisers – all signals are asynchronous
- A4A design flow for `ASYNC_PHASE_CTRL` component

Multiphase buck converter: Asynchronous phase control



Multiphase buck converter: Asynchronous phase control

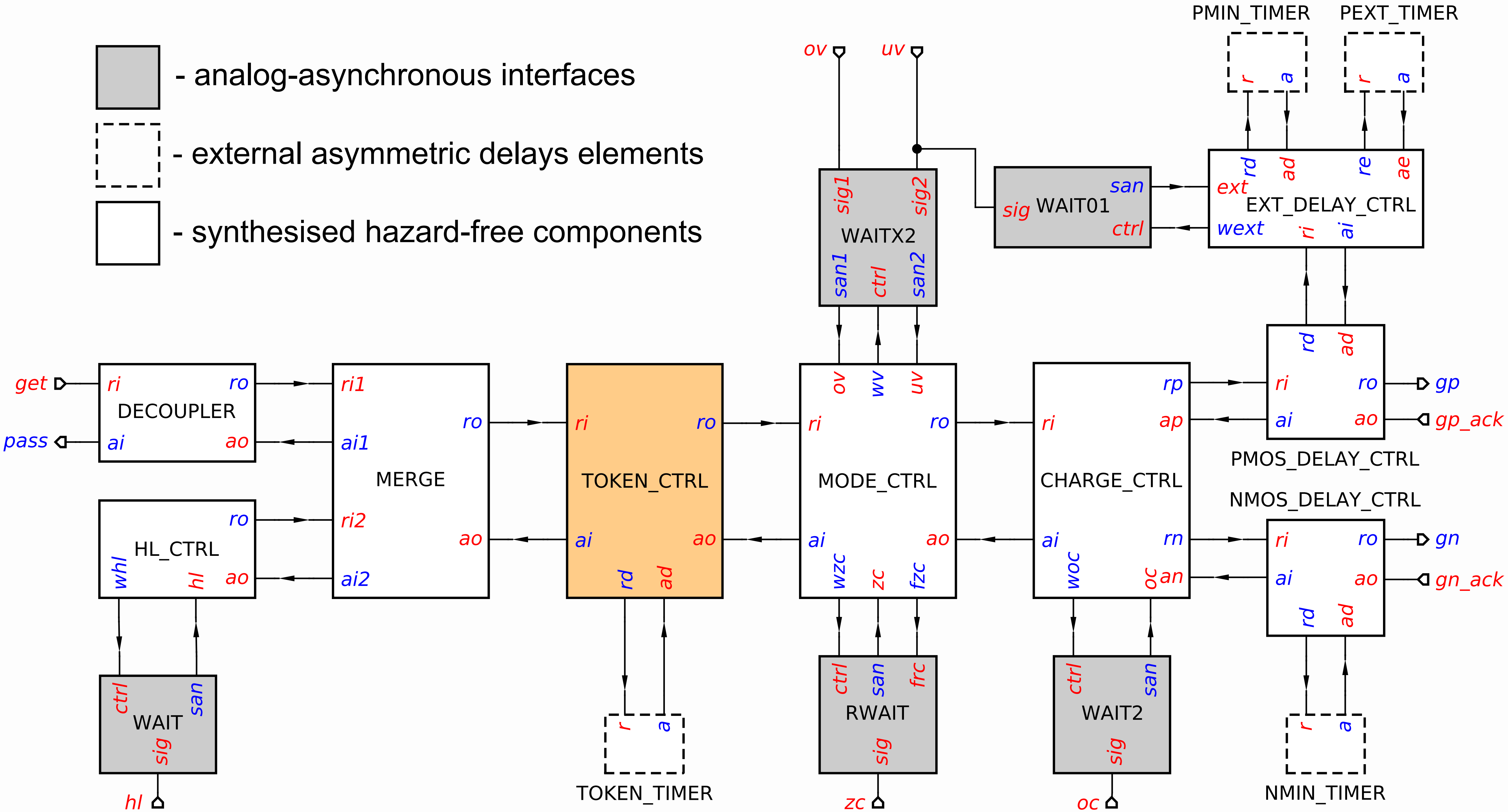


▨ - analog-asynchronous interfaces

⬚ - external asymmetric delays elements

□ - synthesised hazard-free components

Multiphase buck converter: Asynchronous phase control

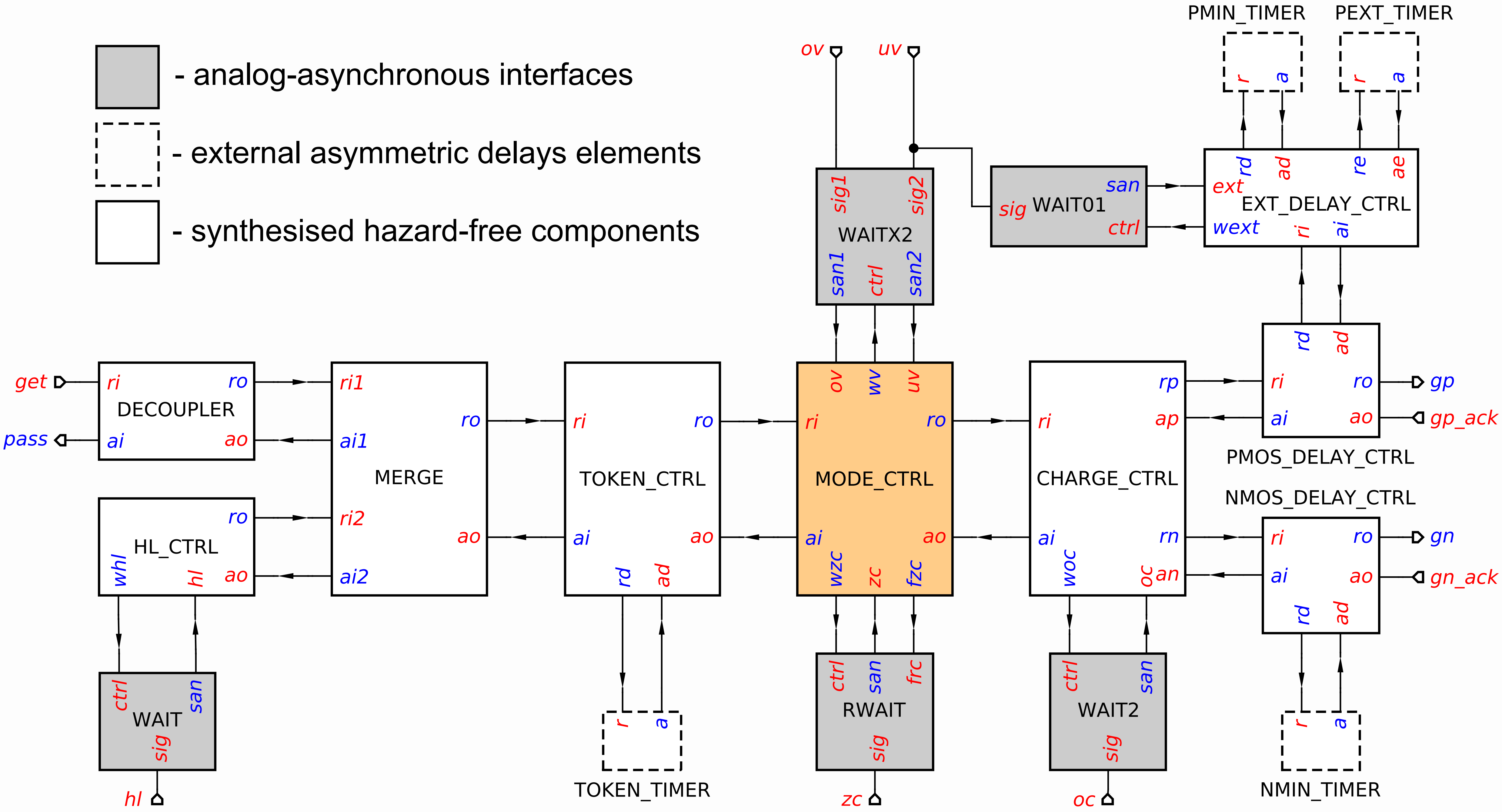


▨ - analog-asynchronous interfaces

⬚ - external asymmetric delays elements

□ - synthesised hazard-free components

Multiphase buck converter: Asynchronous phase control



▨ - analog-asynchronous interfaces

⬚ - external asymmetric delays elements

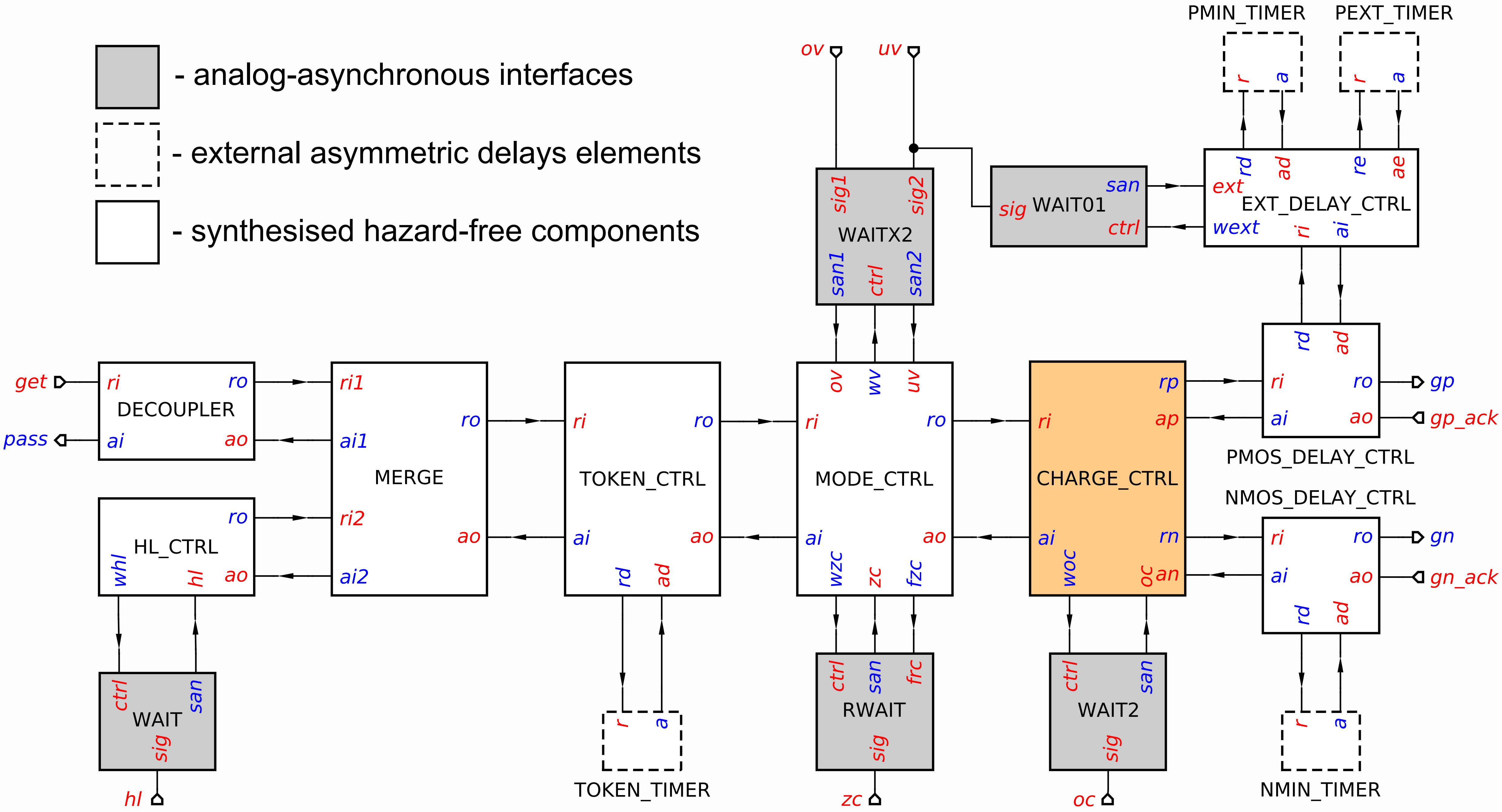
□ - synthesised hazard-free components

Multiphase buck converter: Asynchronous phase control

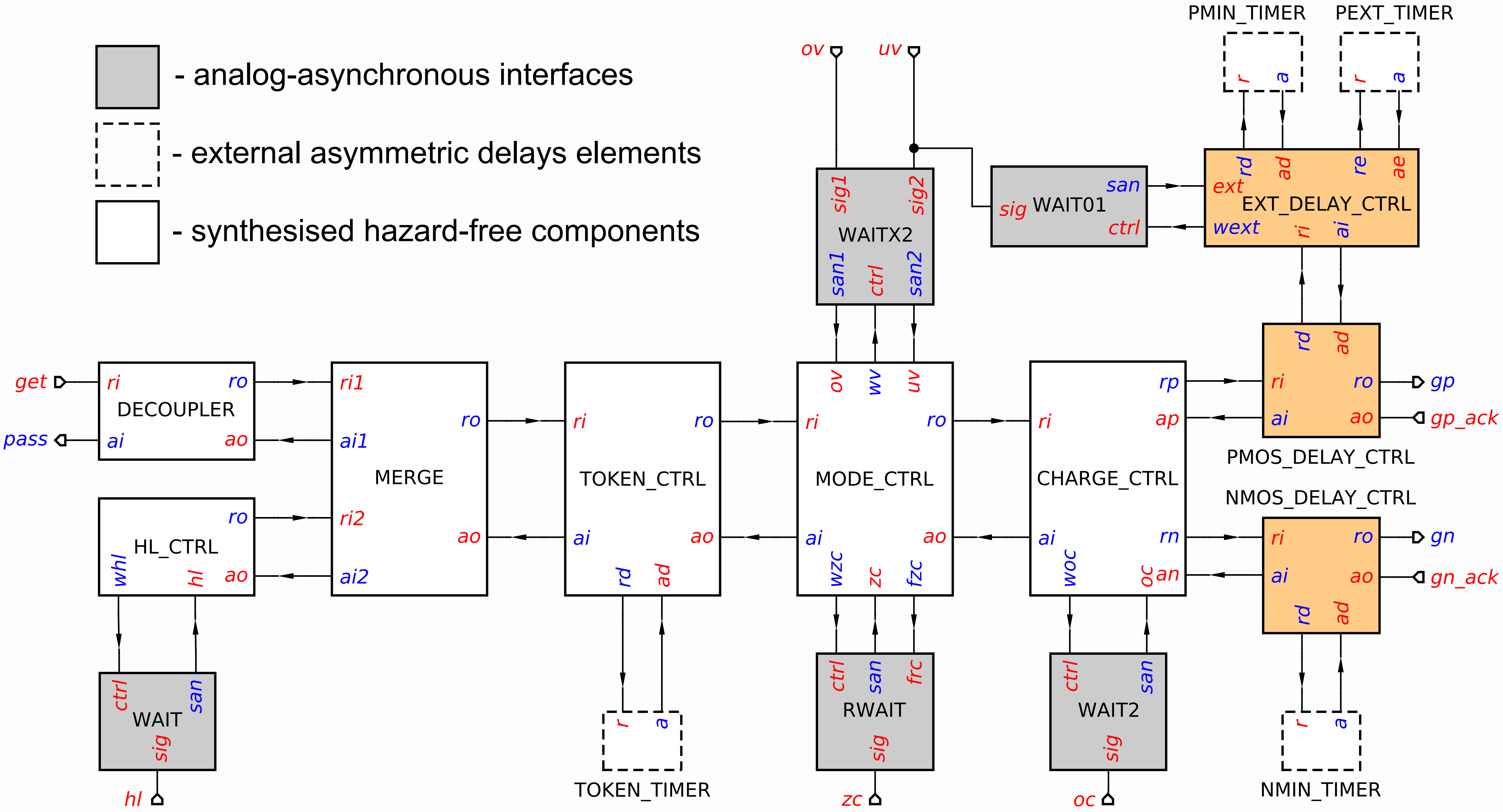
▨ - analog-asynchronous interfaces

⬚ - external asymmetric delays elements

□ - synthesised hazard-free components



Multiphase buck converter: Asynchronous phase control



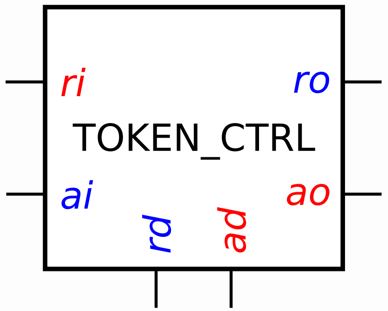
▨ - analog-asynchronous interfaces

⬚ - external asymmetric delays elements

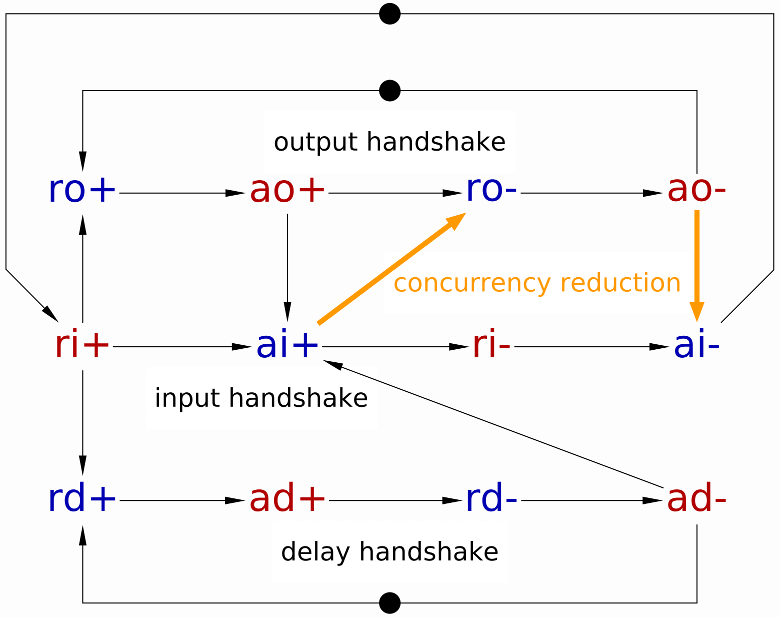
□ - synthesised hazard-free components

Multiphase buck converter: Design of asynchronous components

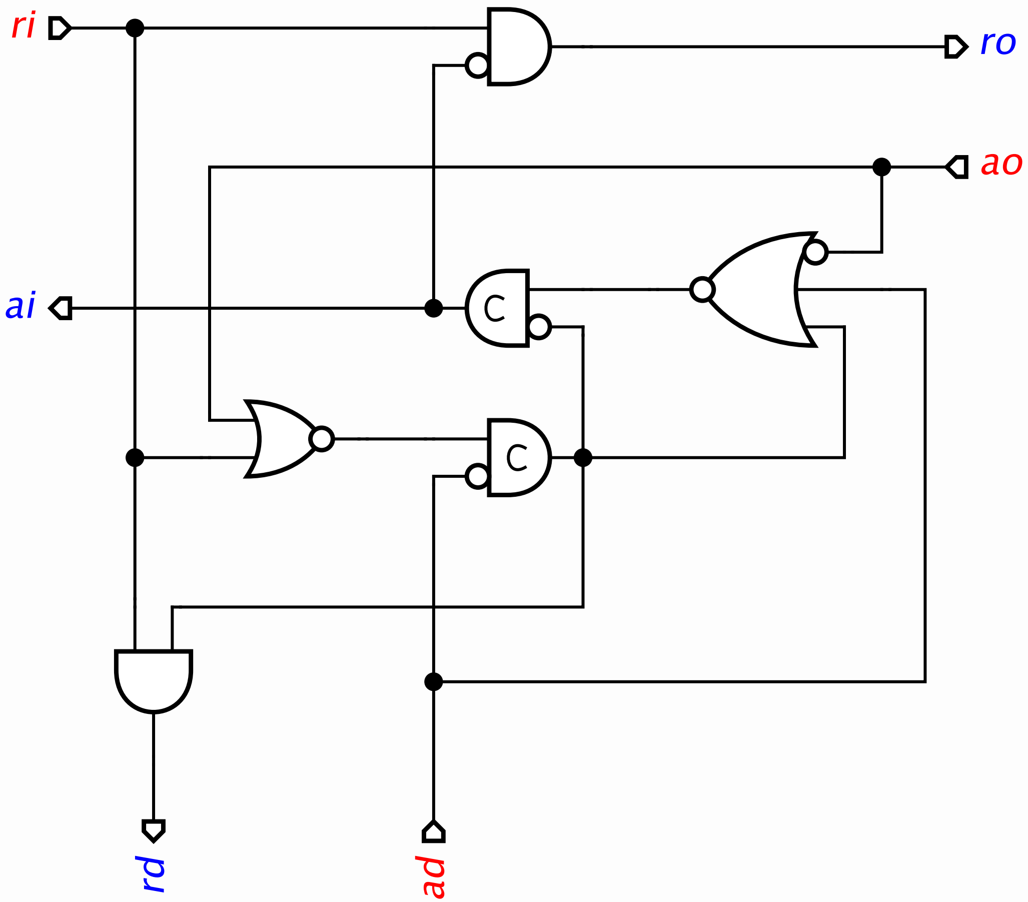
- Token control



- Formal specification



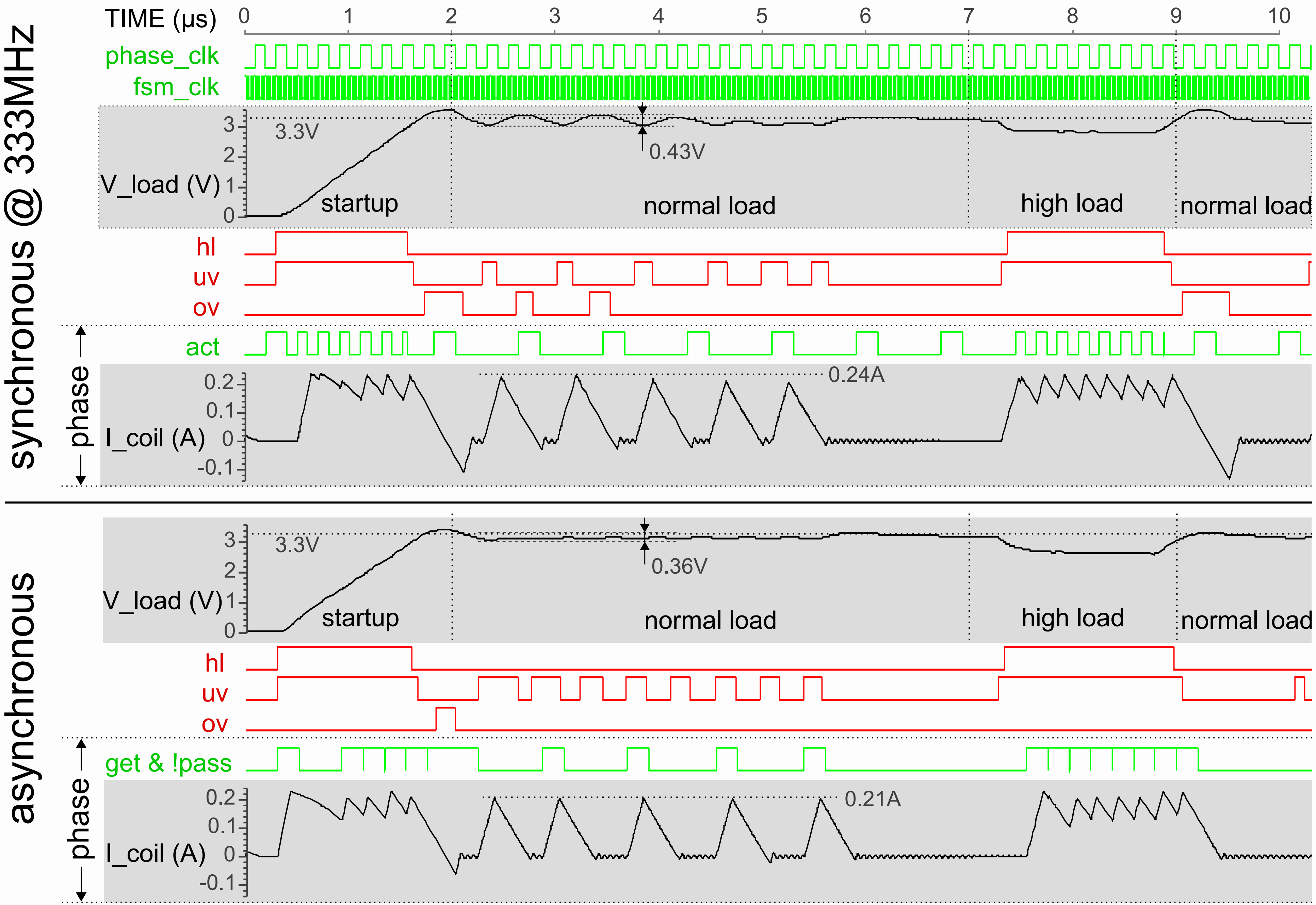
- Hazard-free implementation



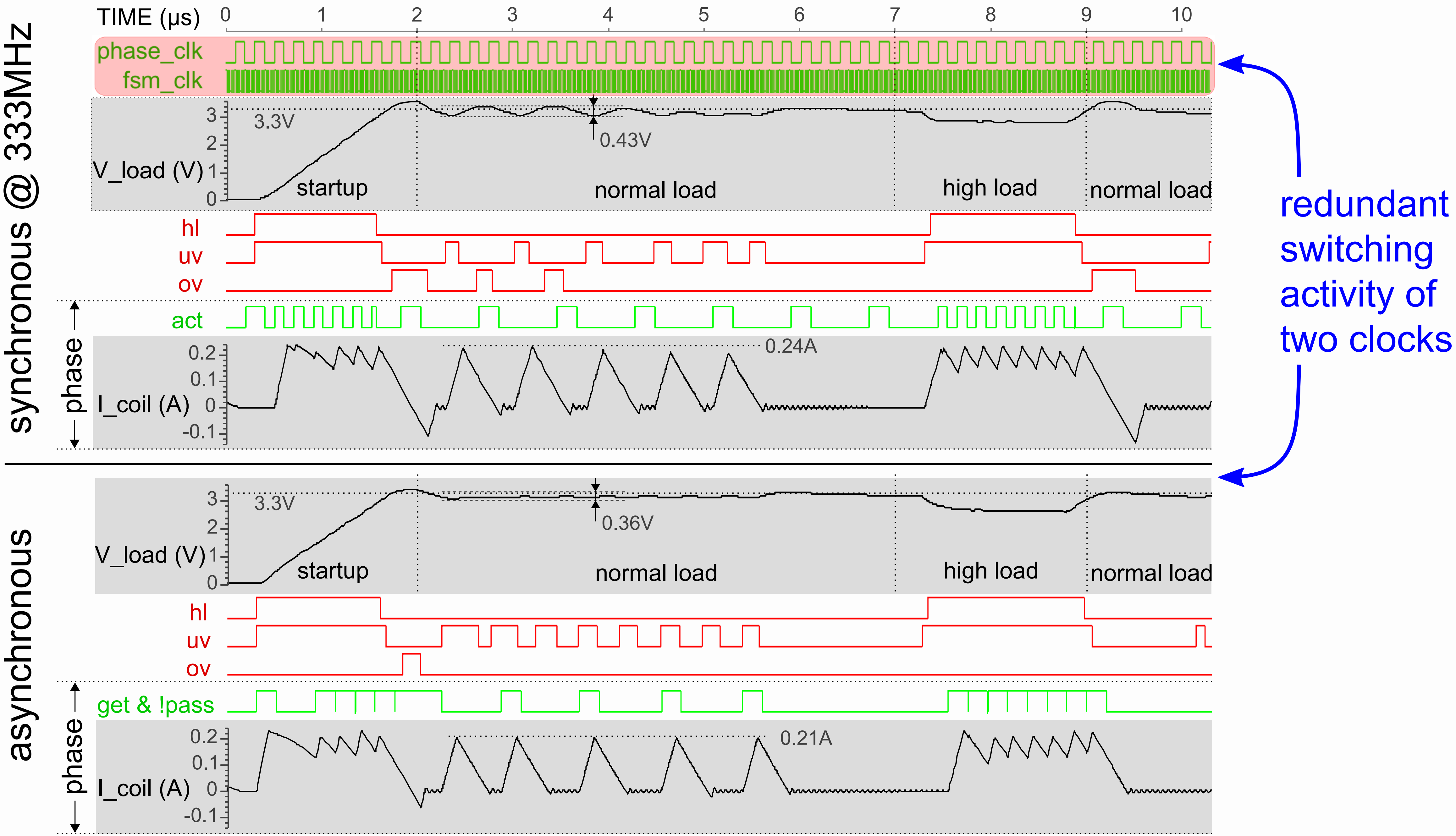
Experimental results: Simulation setup

- Verilog-A model of the 4-phase buck
- Control implemented in TSMC 90nm
- AMS simulation in CADENCE NC-VERILOG
- Synchronous design
  - Phase activation clock – 5MHz
  - Clocked FSM-based control – 100MHz, 333MHz, 666MHz, 1GHz
  - Sampling and synchronisation
- Asynchronous design
  - Phase activation – token ring with 200ns timer (= 5MHz)
  - Event-driven control (input-output mode)
  - Waiting rather than sampling (A2A components)

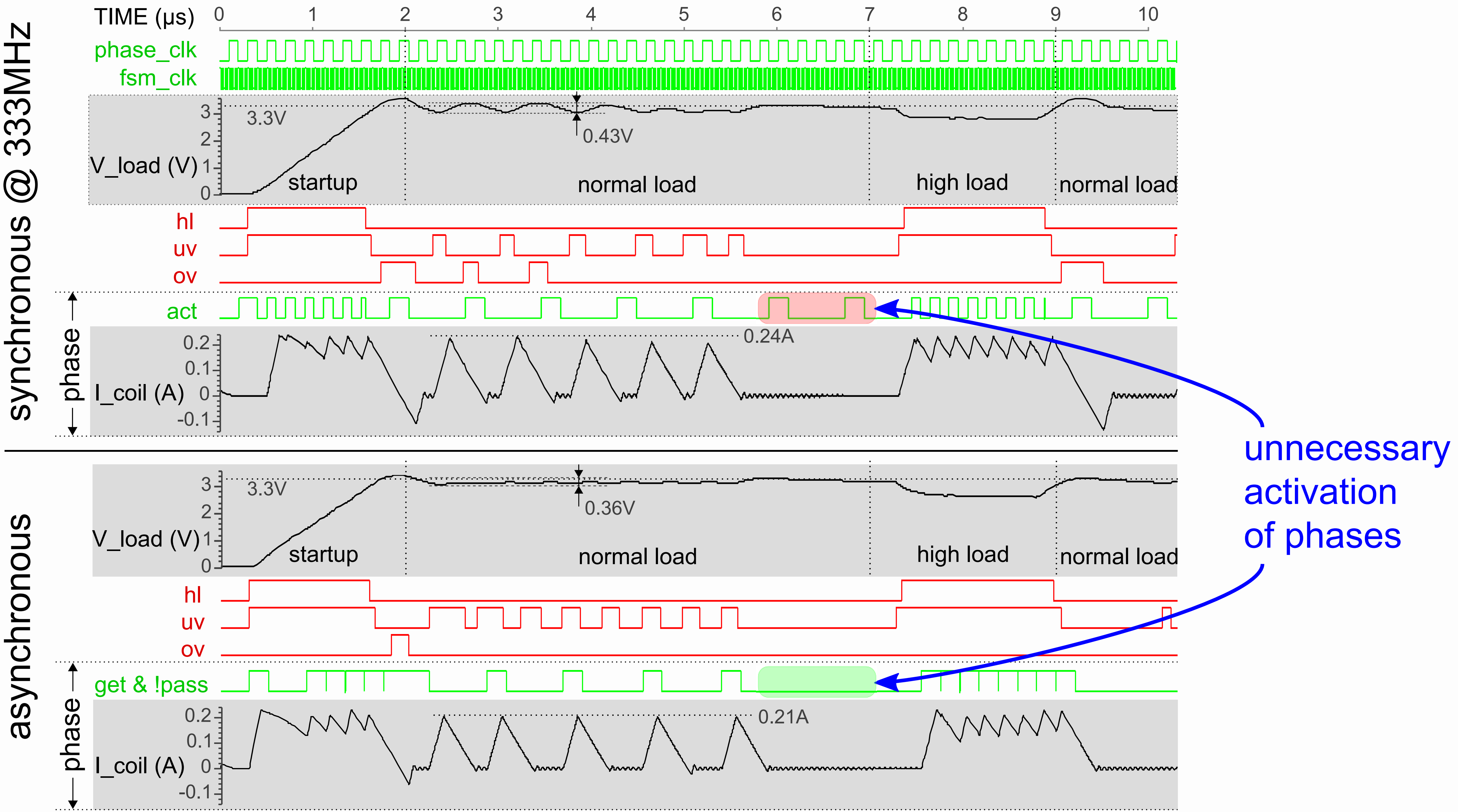
Experimental results: Simulation waveforms



Experimental results: Simulation waveforms

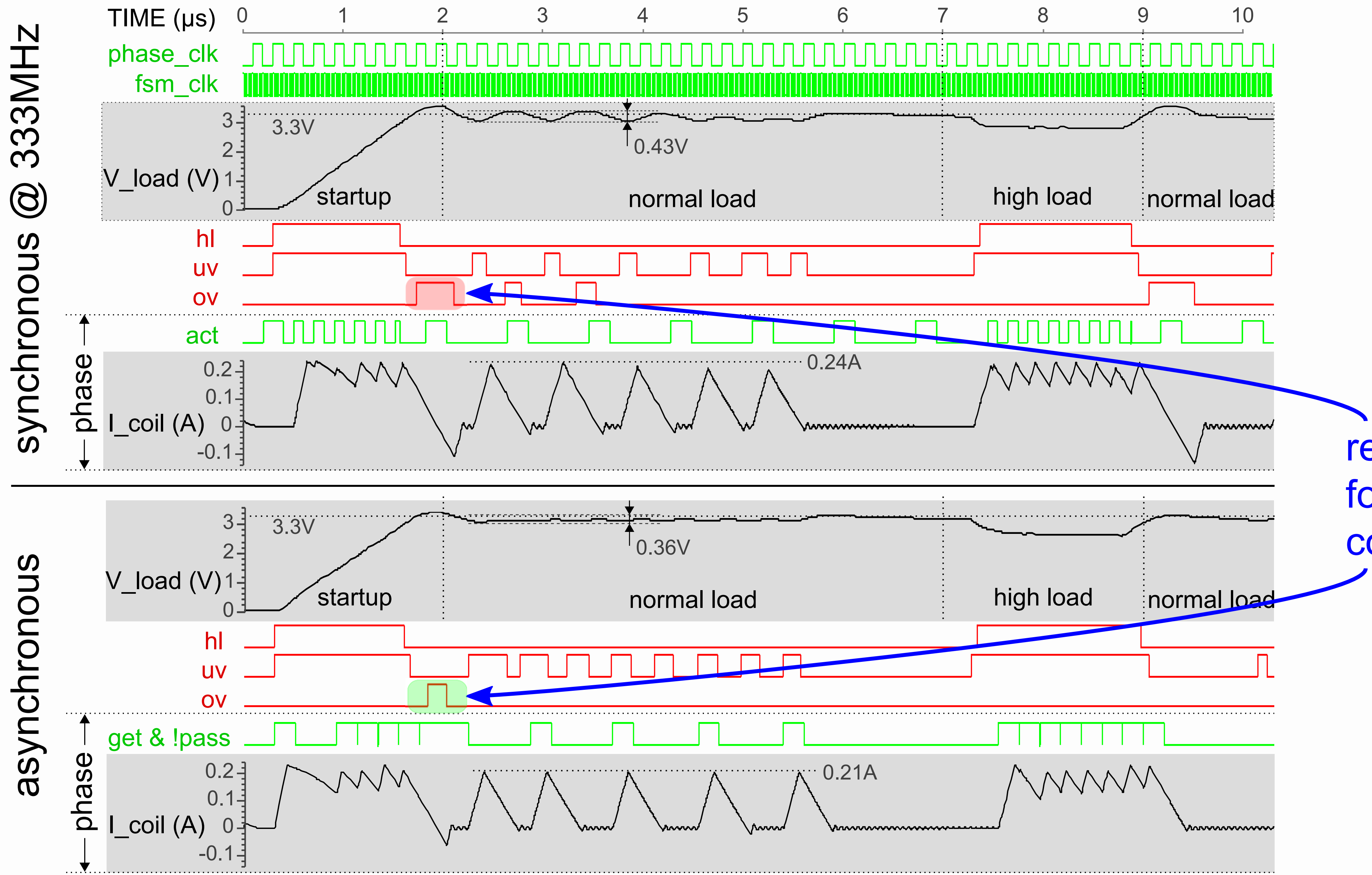


Experimental results: Simulation waveforms



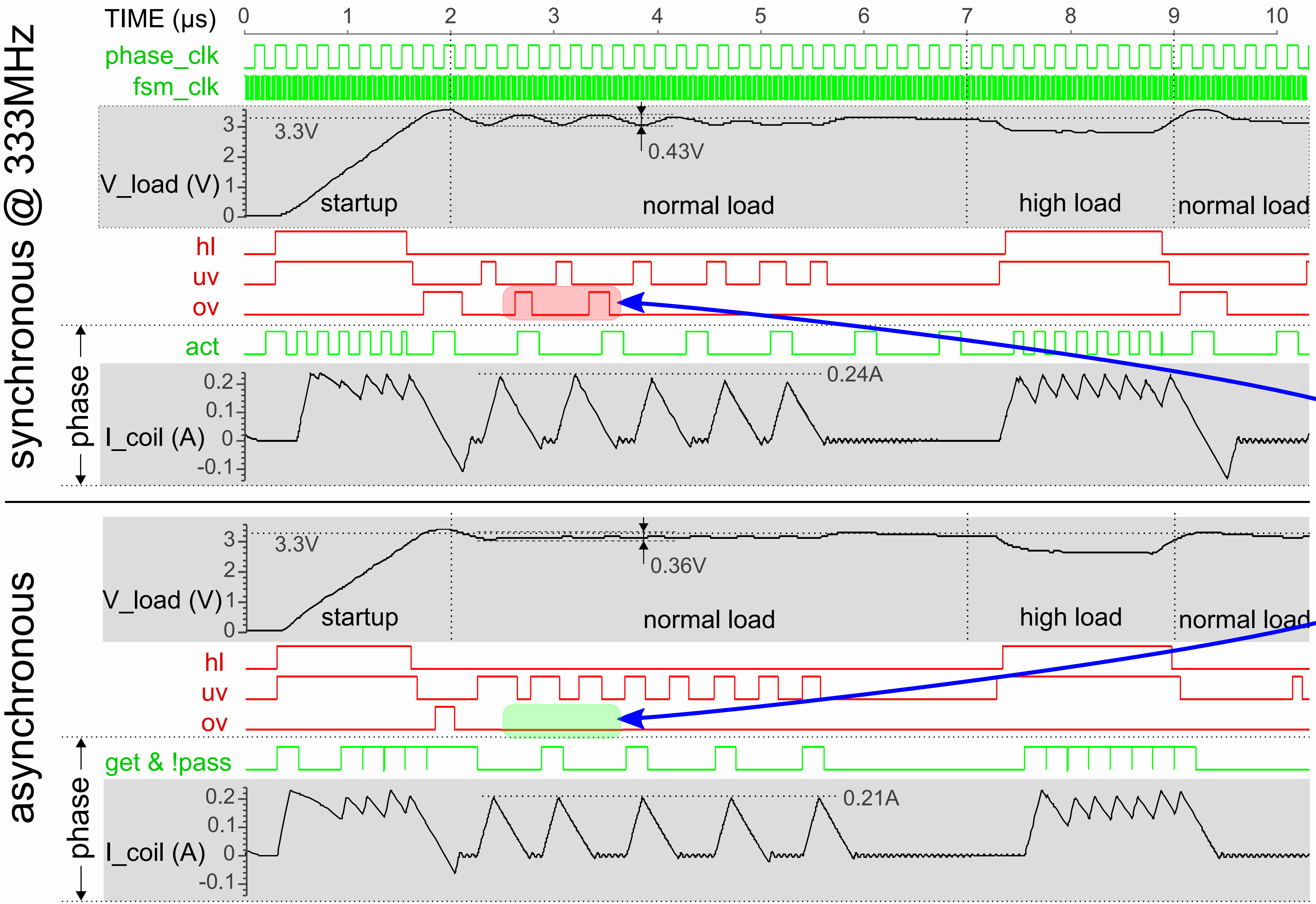
unnecessary activation of phases

Experimental results: Simulation waveforms



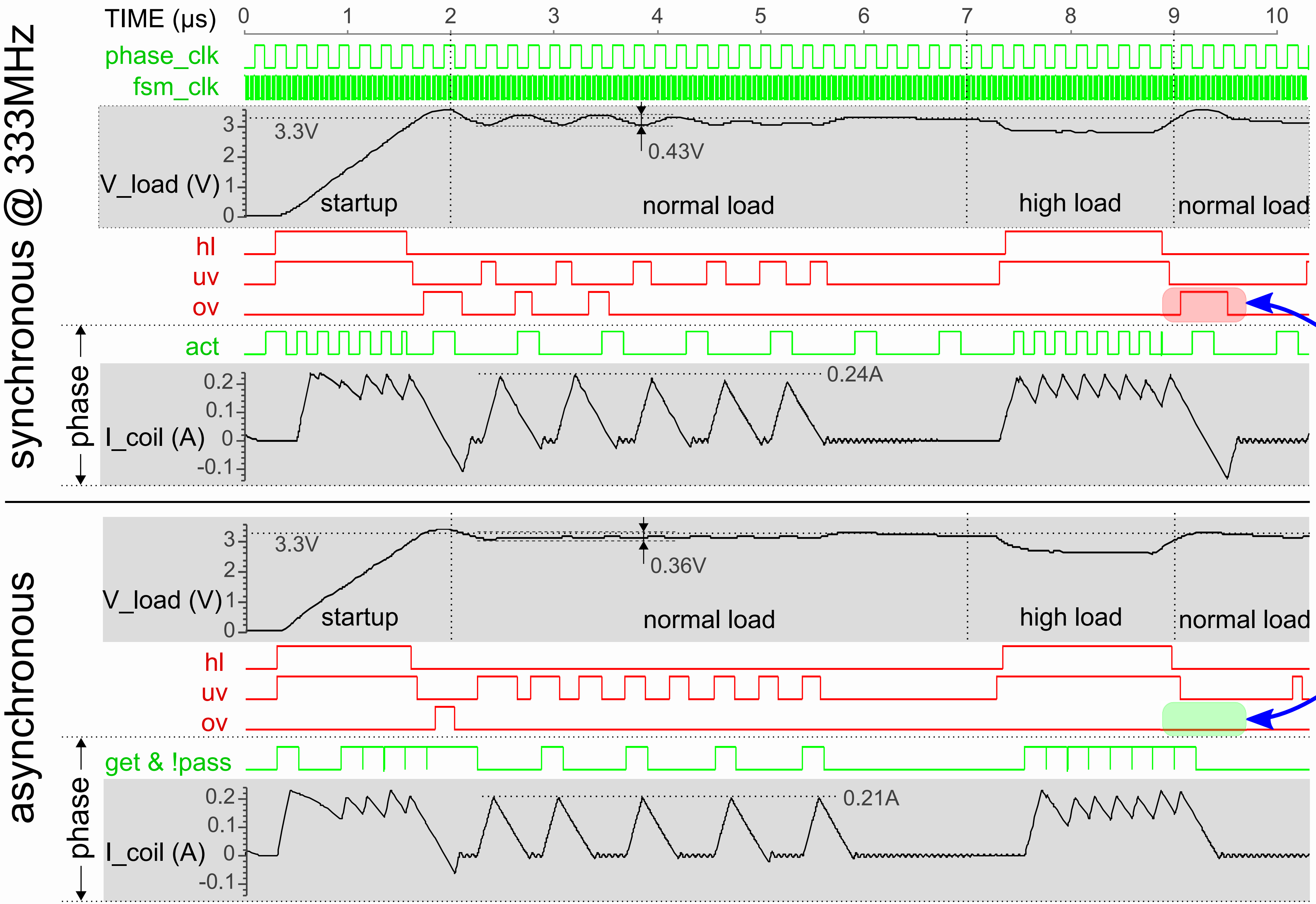
resolution time for over-voltage condition

Experimental results: Simulation waveforms



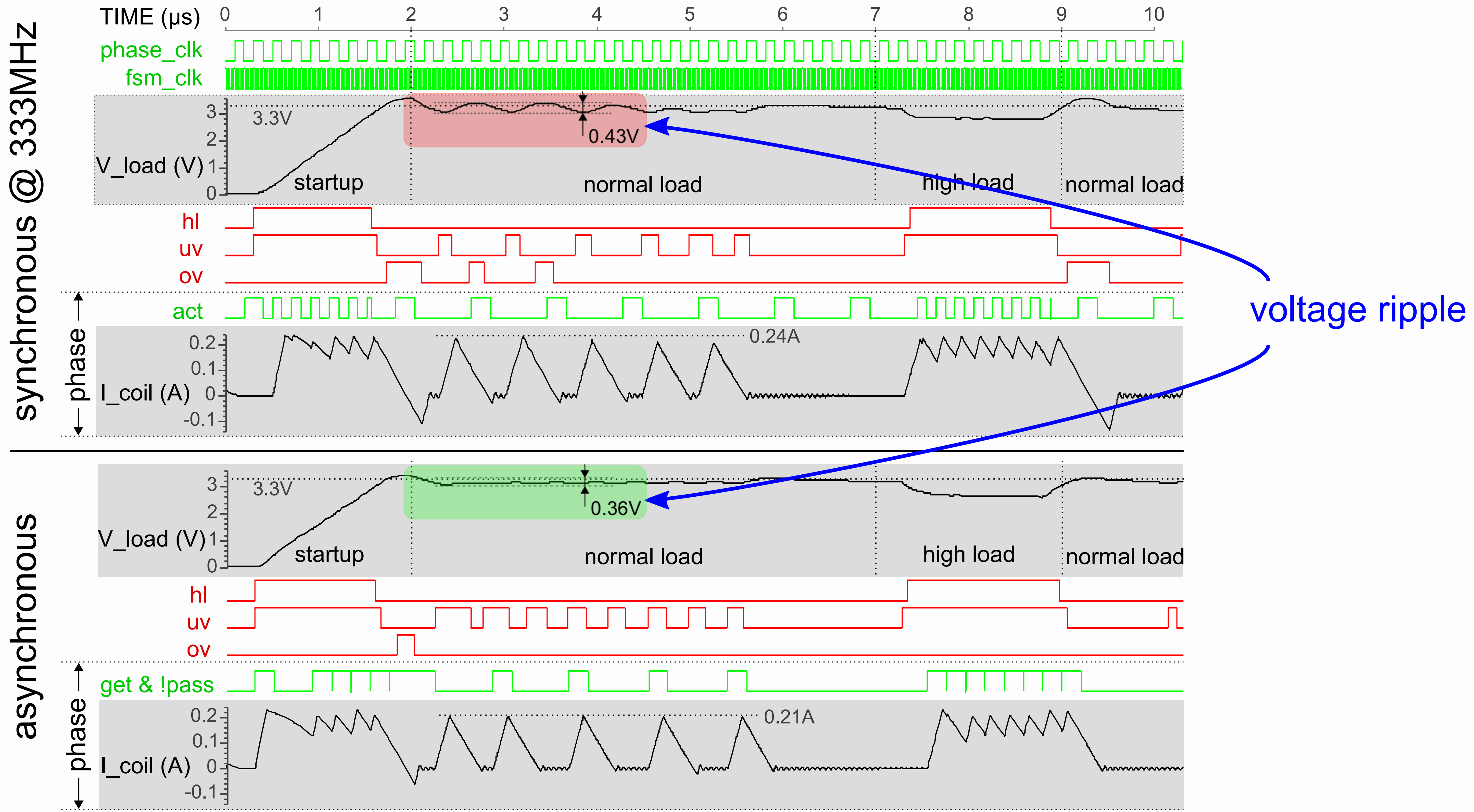
recuring over-voltage condition

Experimental results: Simulation waveforms



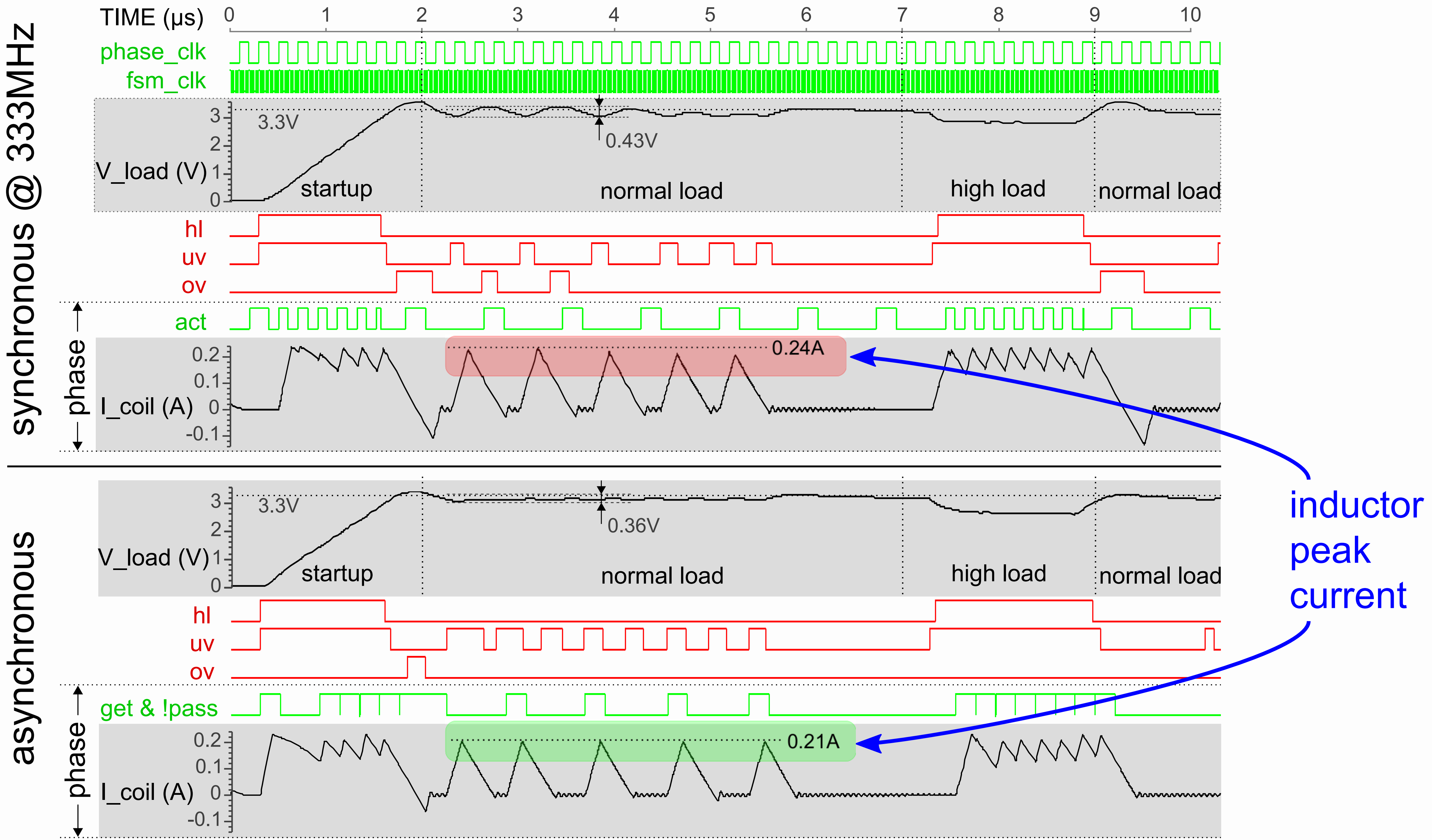
overshoot after high-load condition

Experimental results: Simulation waveforms



voltage ripple

Experimental results: Simulation waveforms

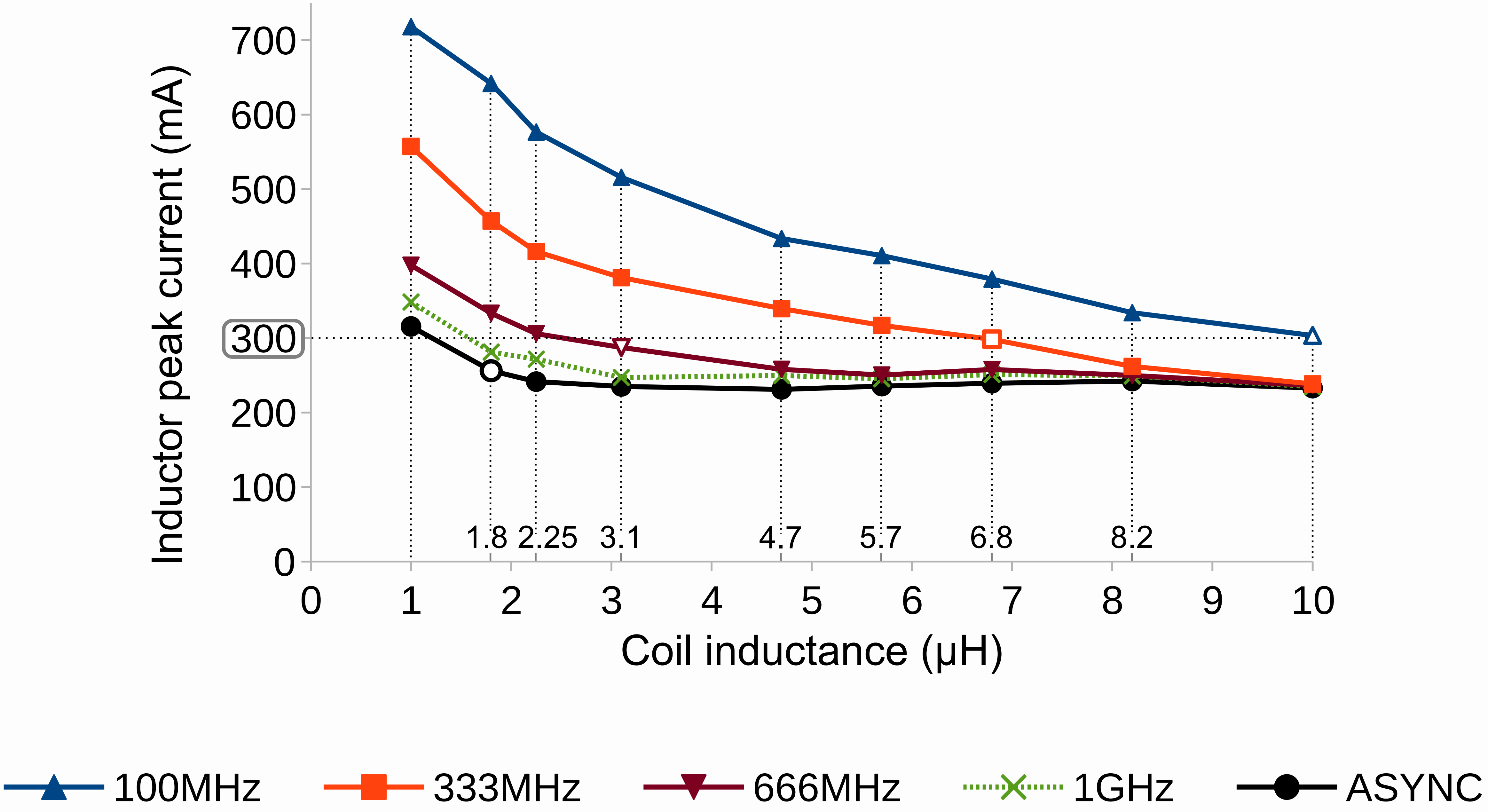


Experimental results: Reaction time

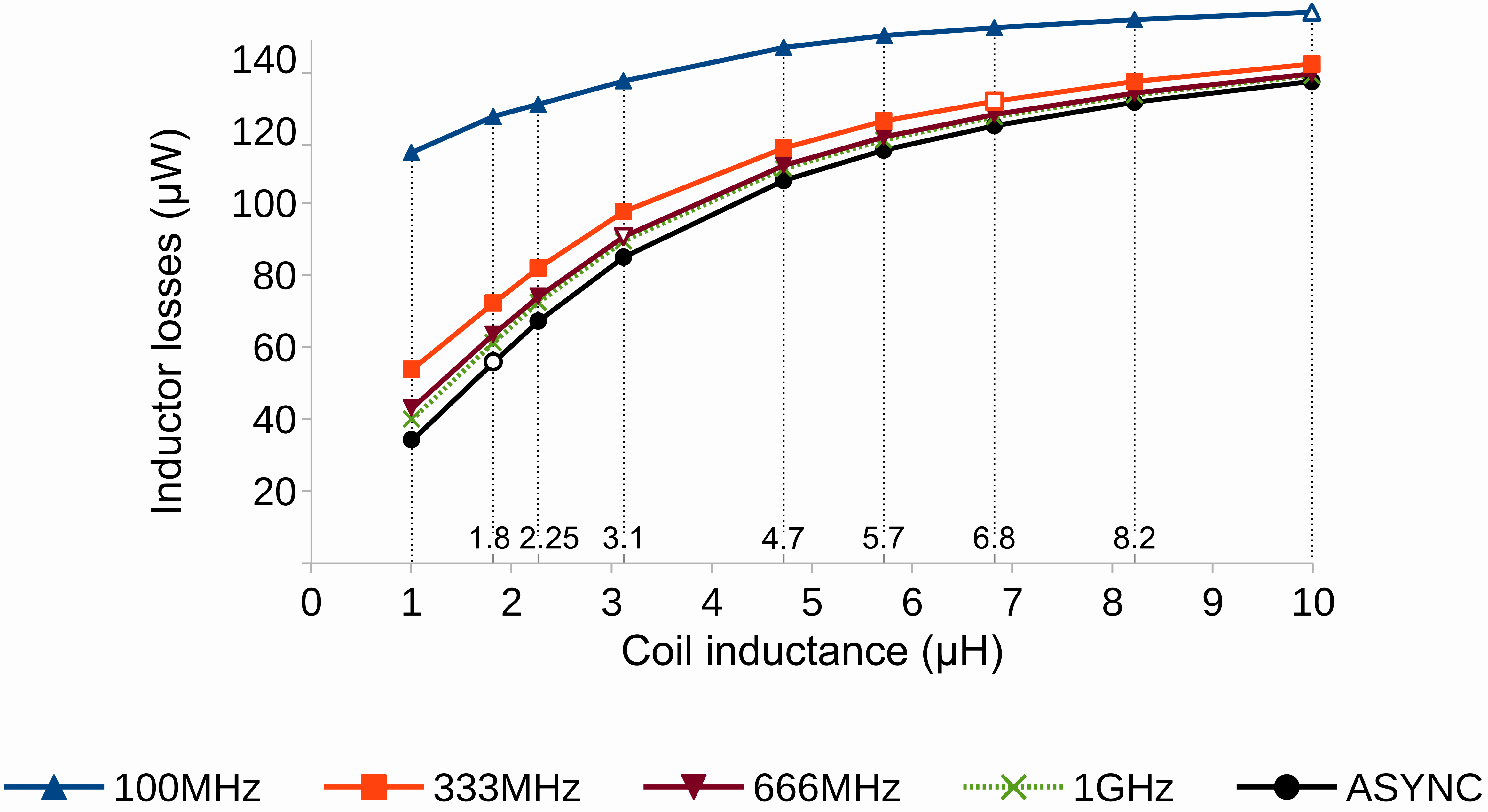
Buck controller	HL (ns)	UV (ns)	OV (ns)	OC (ns)	ZC (ns)
SYNC @ 100MHz	25.00	25.00	25.00	25.00	25.00
SYNC @ 333MHz	7.50	7.50	7.50	7.50	7.50
SYNC @ 666MHz	3.75	3.75	3.75	3.75	3.75
SYNC @ 1GHz	2.50	2.50	2.50	2.50	2.50
ASYNC	1.87	1.02	1.18	0.75	0.31
Improvement over 333MHz	4x	7x	6x	10x	24x

Synchronous buck controllers exhibit latency of 2.5 clock cycles.

Experimental results: Peak current



Experimental results: Inductor losses



Conclusions

- A4A design flow is automated in Workcraft framework
  - Library of A2A components
  - Automatic synthesis of hazard-free circuits
  - Formal verification at the specification and circuit levels
- Benefits of asynchronous multiphase buck controller
  - Reliable, no synchronisation failures
  - Quick response time (few gate delays)
  - Reaction time can be traded off for smaller coils
  - Lower voltage ripple and peak current
- A4A flow, A2A interfaces, and WORKCRAFT tools are used at Dialog Semiconductor