A Workflow for the Design of Mixed-signal Systems with Asynchronous Control

Vladimir Dubikhin, Danil Sokolov, Alex Yakovlev, Chris J. Myers

AMS Trends & Challenges

Key Drivers

- Internet of Things
- Mobile computing
- Automotive electronics

Trends

- Technology scaling
- Multiple power and time domains
- Analog and digital integration

Challenges

- Tighter reliability margins
- Concurrent analog and digital analysis
- Short development cycle

Based on slide from DAC2014 by ANSYS

What this means for AMS?

- **Achieving better verification of a*nalog and digital blocks***
- ***Verifying the increasing amount of digital logic in analog designs***
- **Creating a *higher level of abstraction* for analog and mixed signal blocks**
- ***Automating* the manual custom design steps**
- **Adopting *circuit analytics* that tell *why* and *where* the circuit is failing to perform**

Based on slide from ISQED2013 by Mentor Graphics

Why Asynchronous Logic?

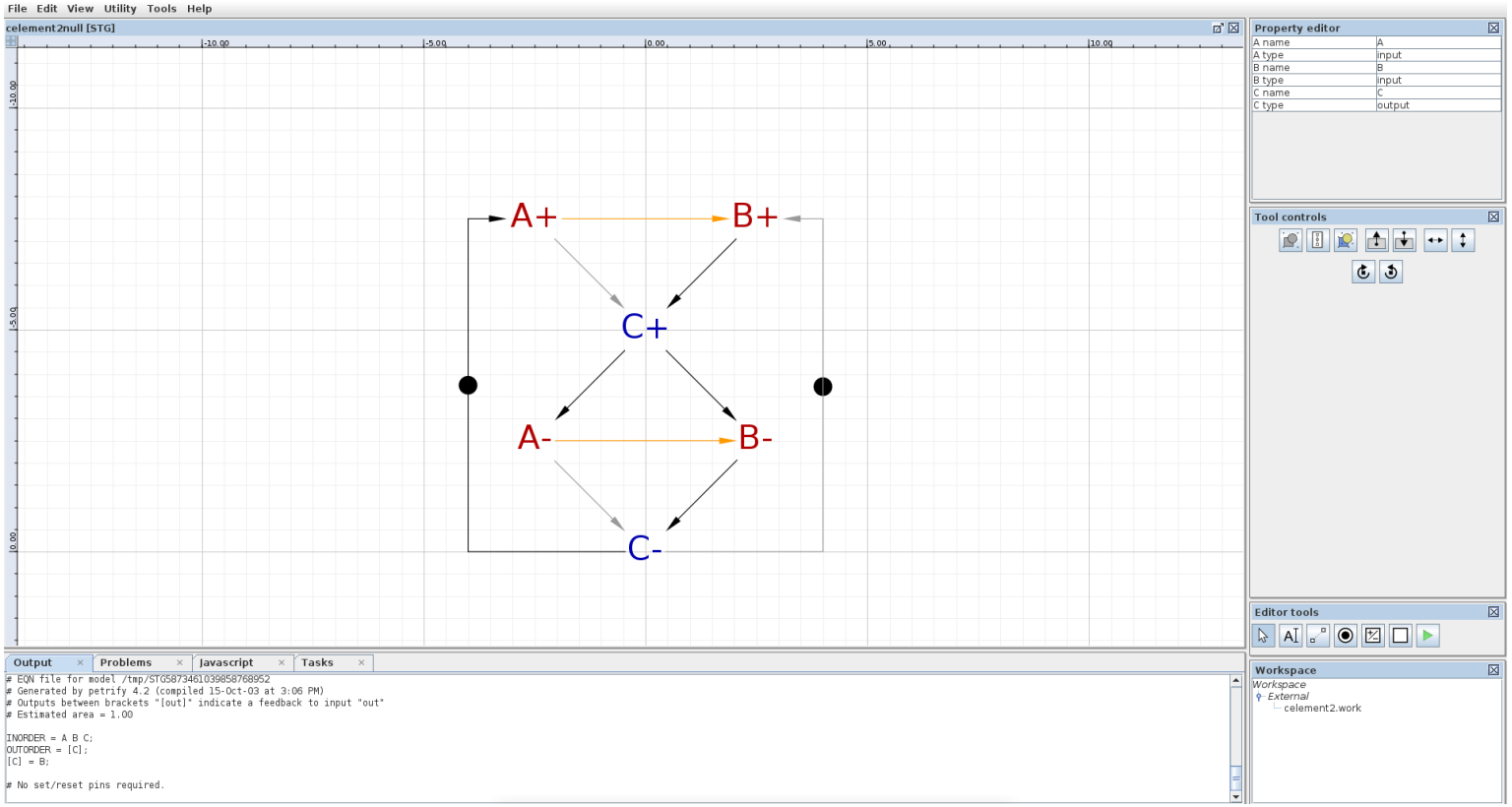
- Insensitive to delays
- Robust to process-voltage-temperature
- Average case performance
- Low power consumption and EMI

Why Asynchronous Logic?

- Insensitive to delays
- Robust to process-voltage-temperature
- Average case performance
- Low power consumption and EMI
- Incompatible with commercial EDA tools

Workcraft

- Modeling with signal transition graphs (STG)
- Formal verification of STG models
- Logic synthesis of asynchronous circuits



Available at http://workcraft.org/

Why Formal Verification?

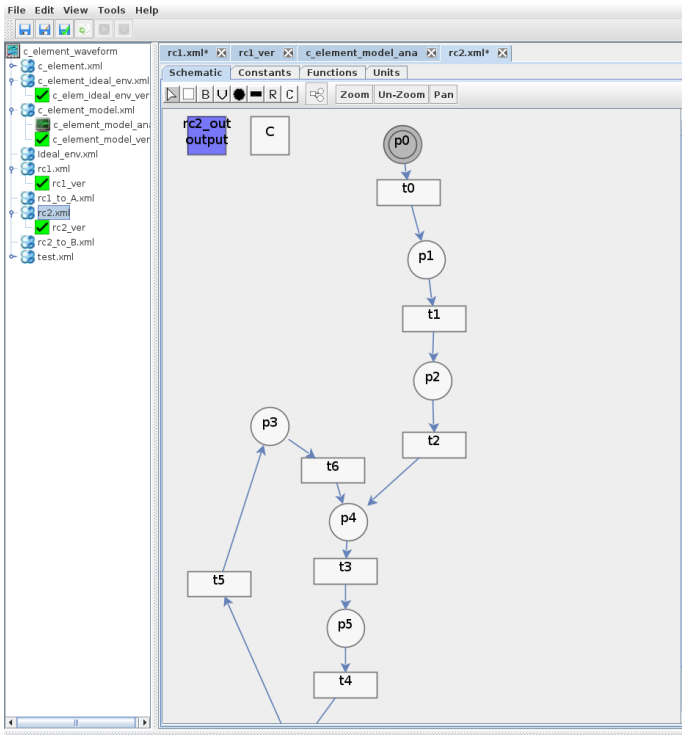
- Increased robustness of the system
- Abstract modeling
- Reduced need for conventional simulation

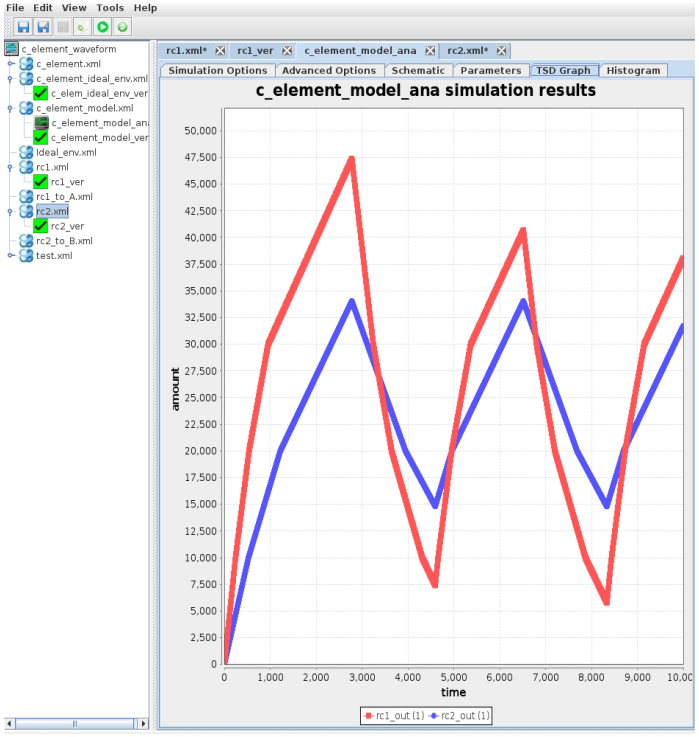
Why Formal Verification?

- Increased robustness of the system
- Abstract modeling
- Reduced need for conventional simulation
- Limited tool support

LEMA

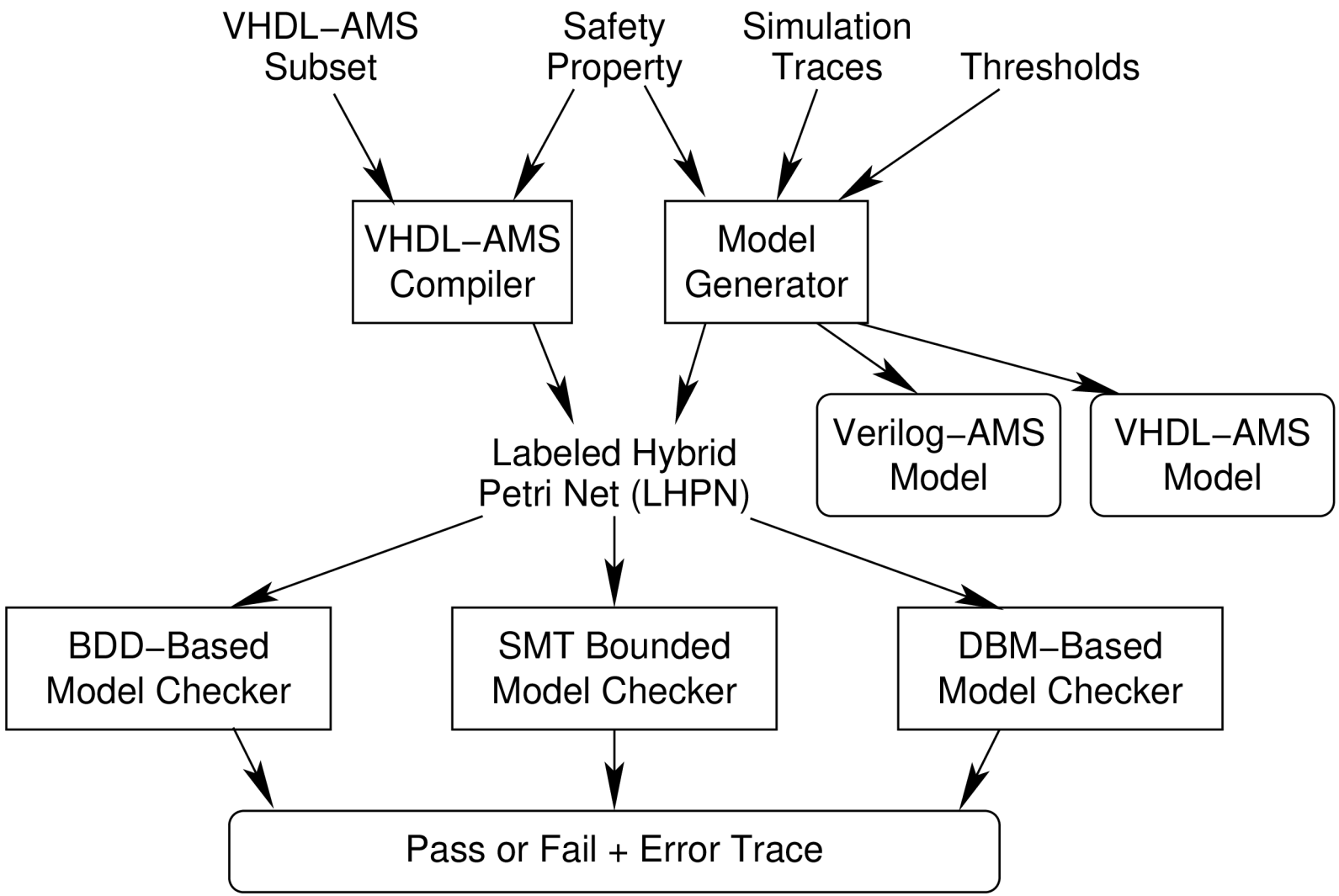
- Modeling with labeled Petri nets(LPN)
- Automatic model generation
- Property expression and checking
- Model extraction as SystemVerilog netlist





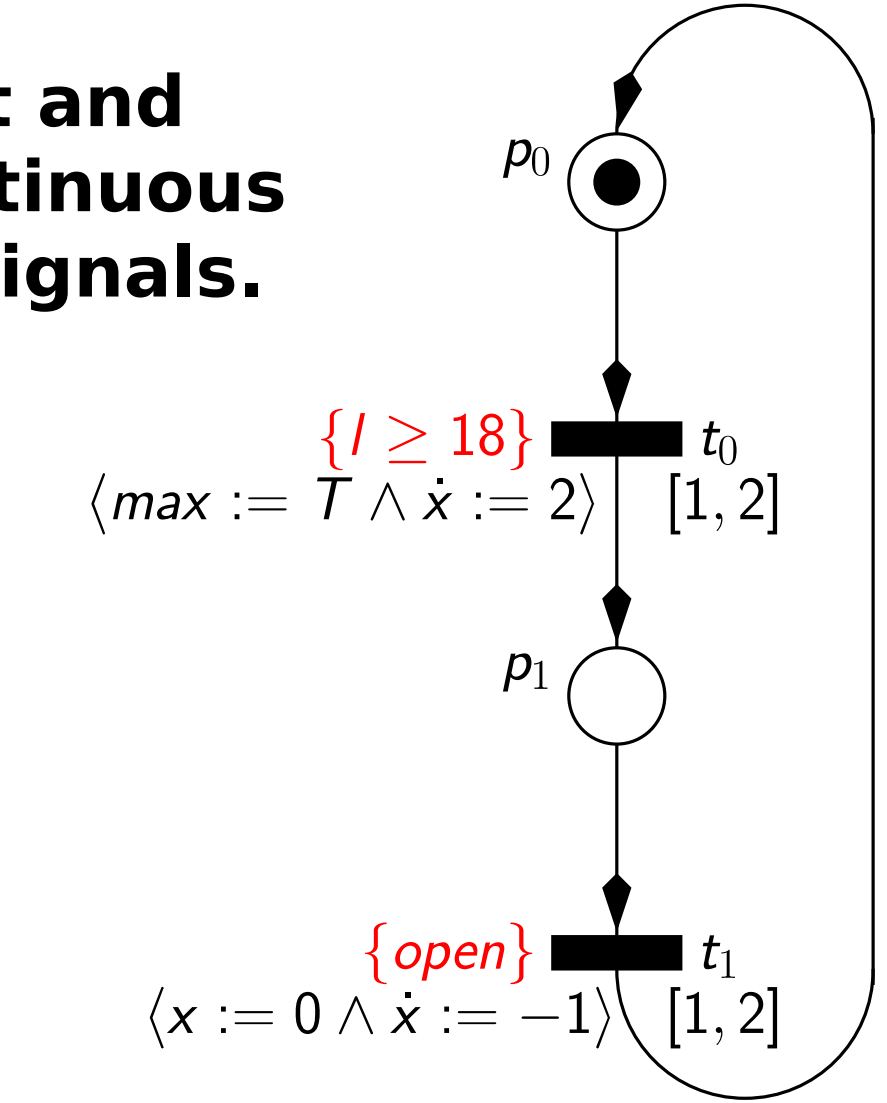
Available at http://async.ece.utah.edu/LEMA/

LEMA Tool Flow



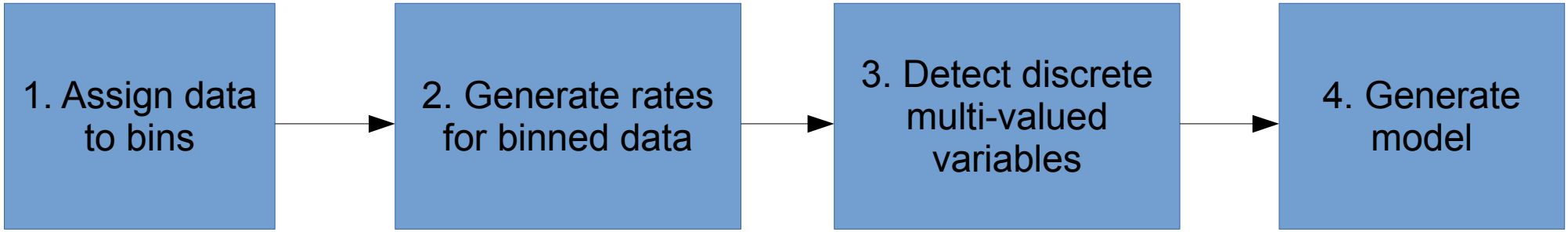
Labeled Petri Nets (LPNs)

- **Composed of a Petri net and labels operating on continuous variables and Boolean signals.**
- **Label types are:**
  - **Enablings**
  - **Delay bounds**
  - **Boolean assignments**
  - **Value assignments**
  - **Rate assignments**

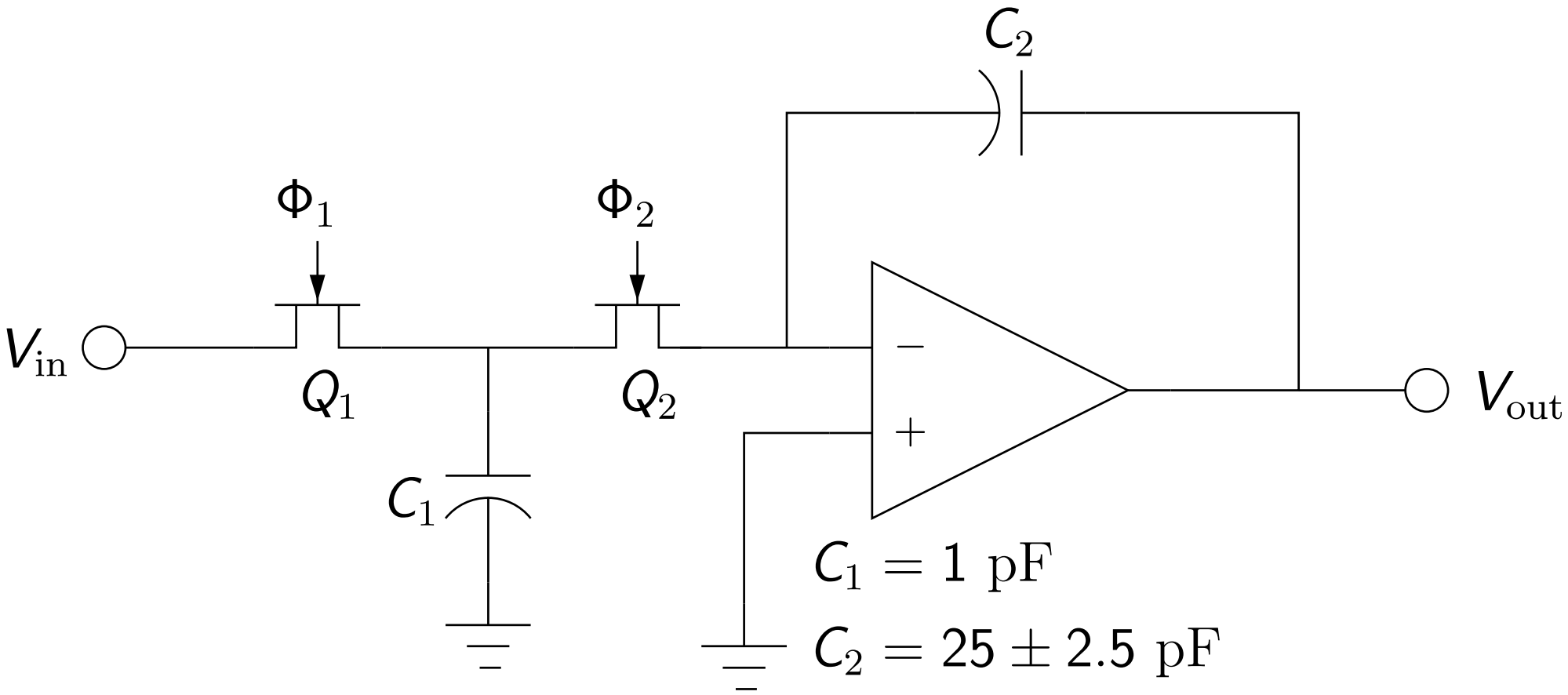


LPN Model Generation

- **Build abstract models of the circuit using:**
  - **Simulation traces.**
  - **Thresholds on the design variables.**
  - **A property to verify.**



Switched Capacitor



$C_1 = 1$ pF

$C_2 = 25 \pm 2.5$ pF

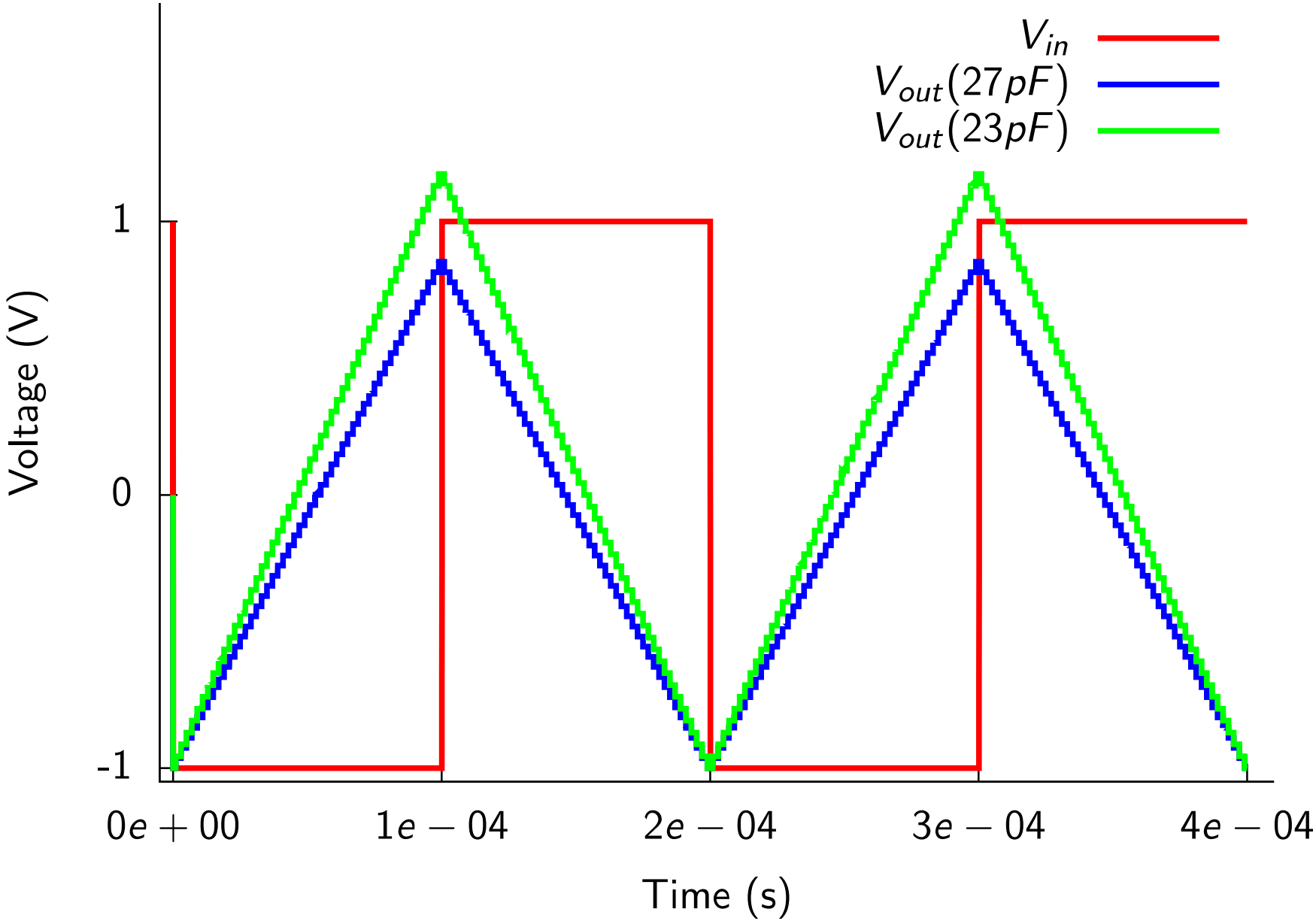
$freq(\Phi_1) = freq(\Phi_2) = 500$ kHz

$V_{\text{in}} = \pm 1000$ mV

$freq(V_{\text{in}}) = 5$ kHz

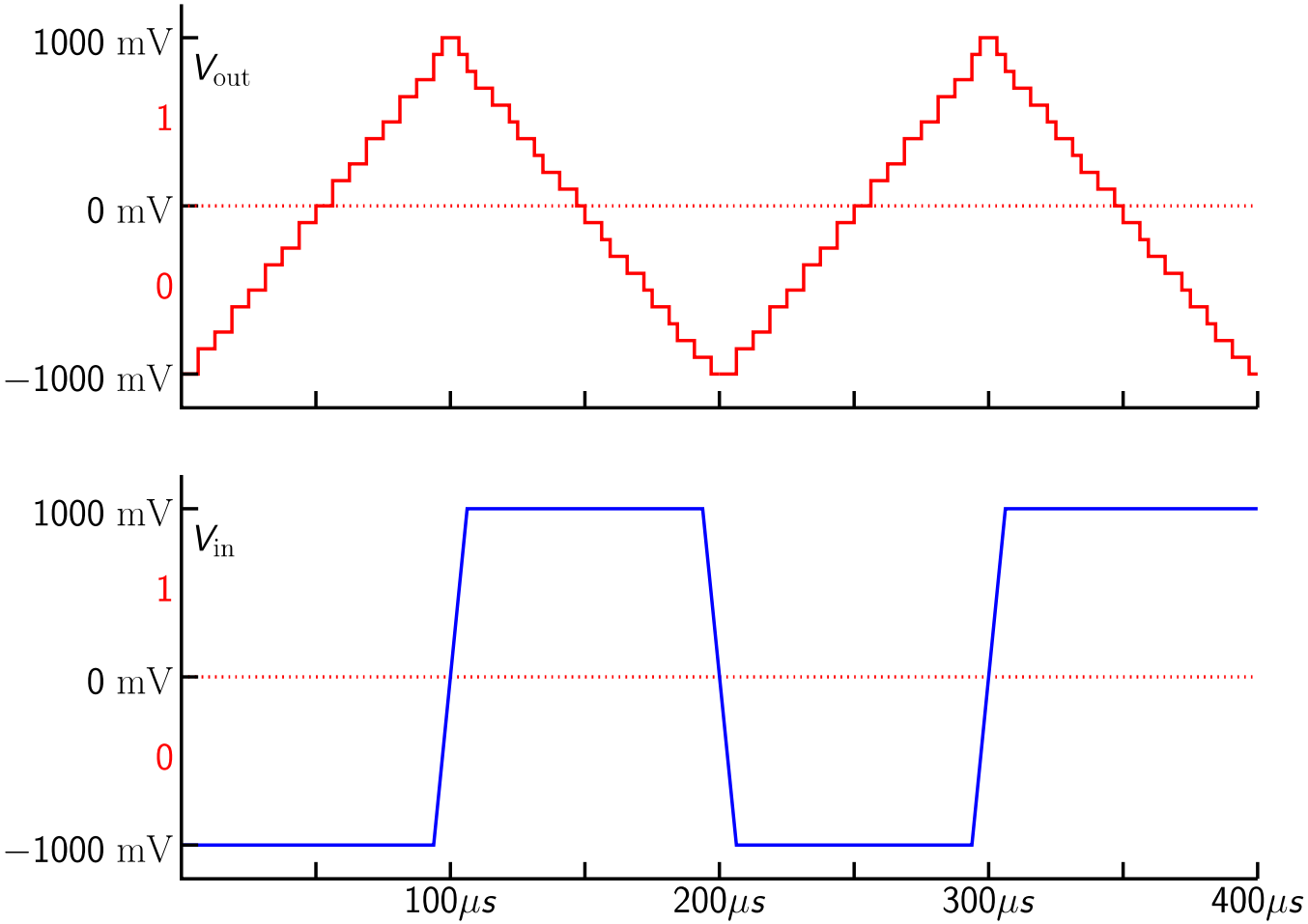
$dV_{\text{out}}/dt = \pm(18 \text{ to } 22)$ mV/μs

Simulation Trace

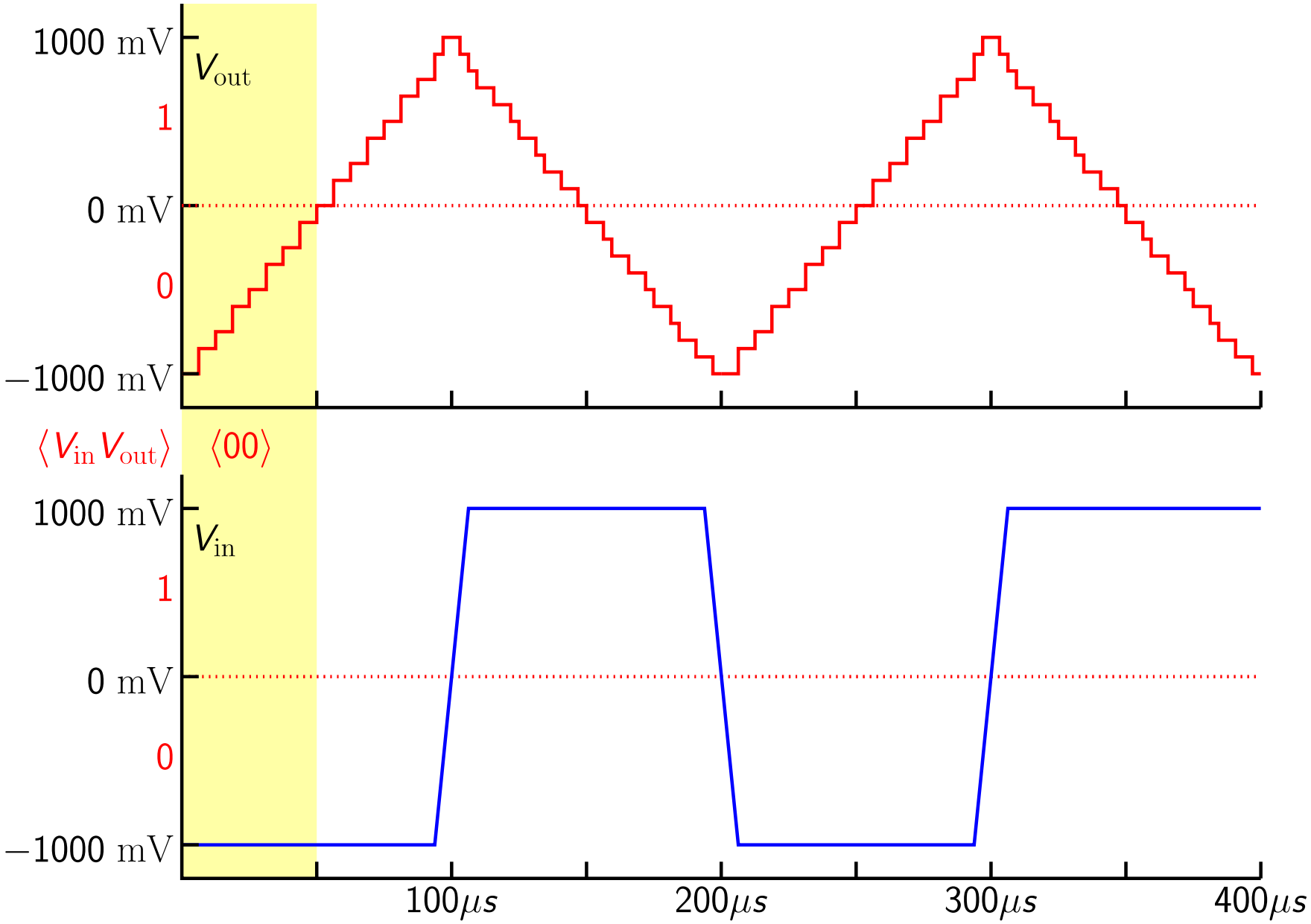


Data Binning

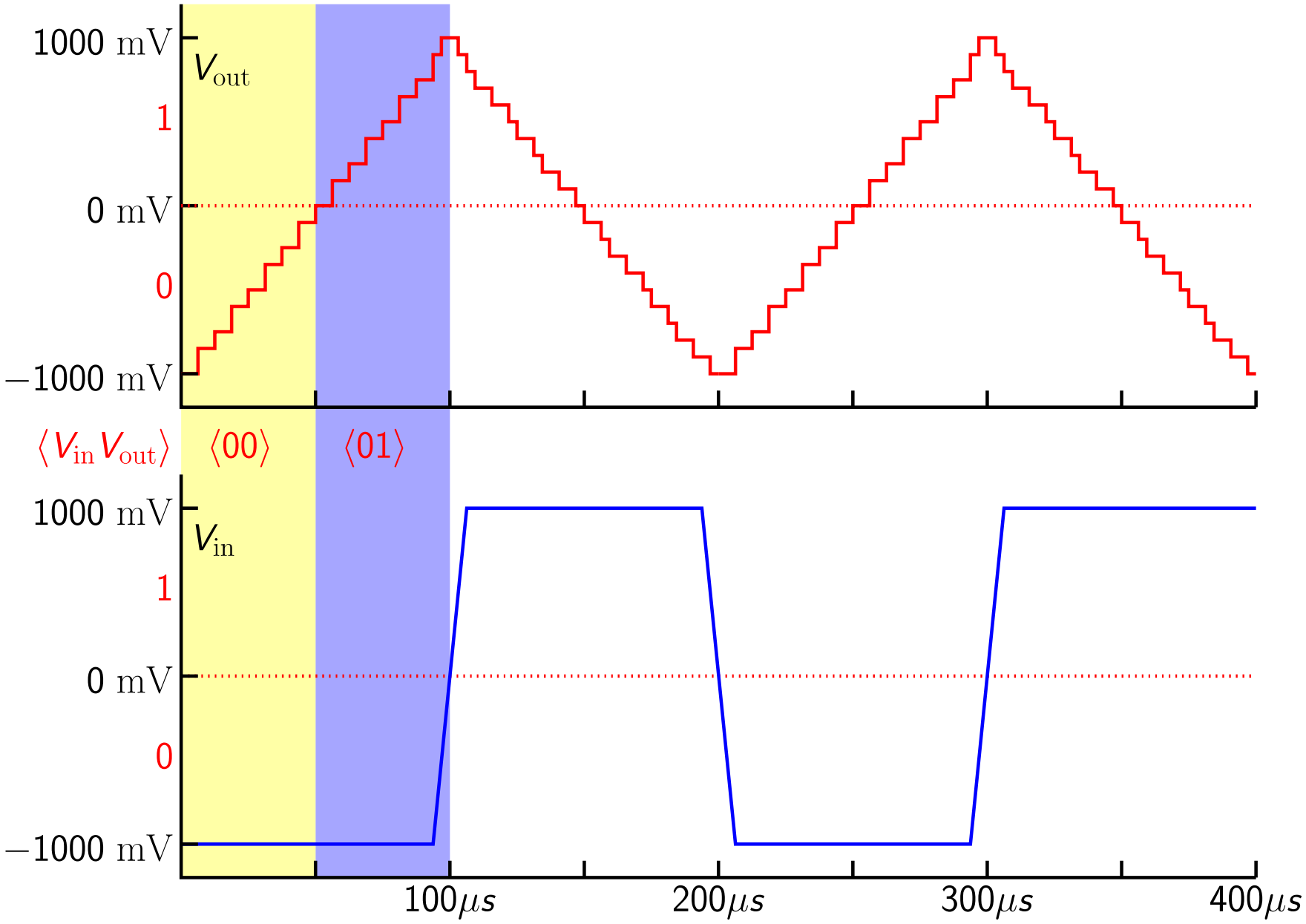
- **Each data point is assigned a bin based upon thresholds.**
- **Each bin represents an operating region of the system.**



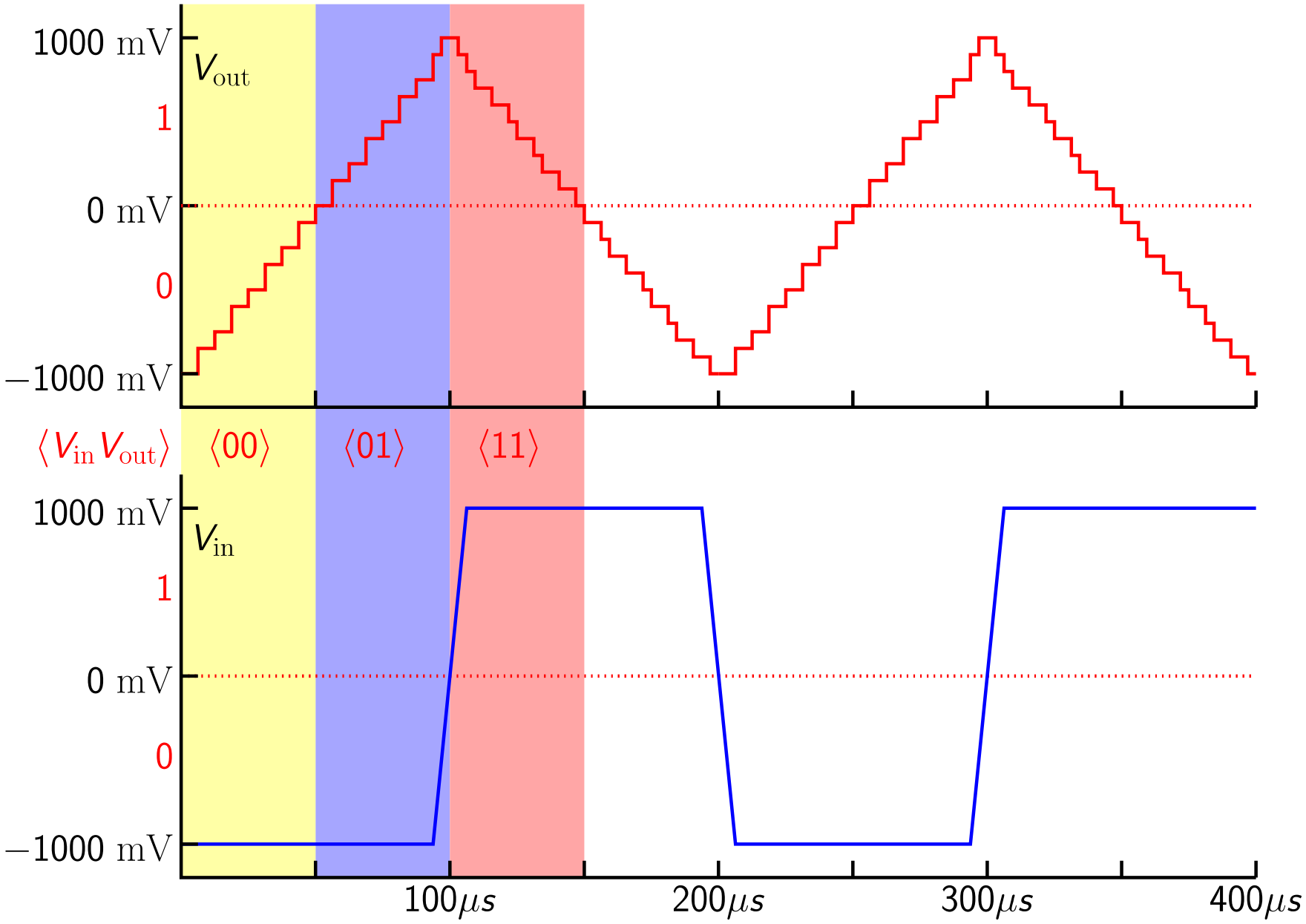
Data Binning



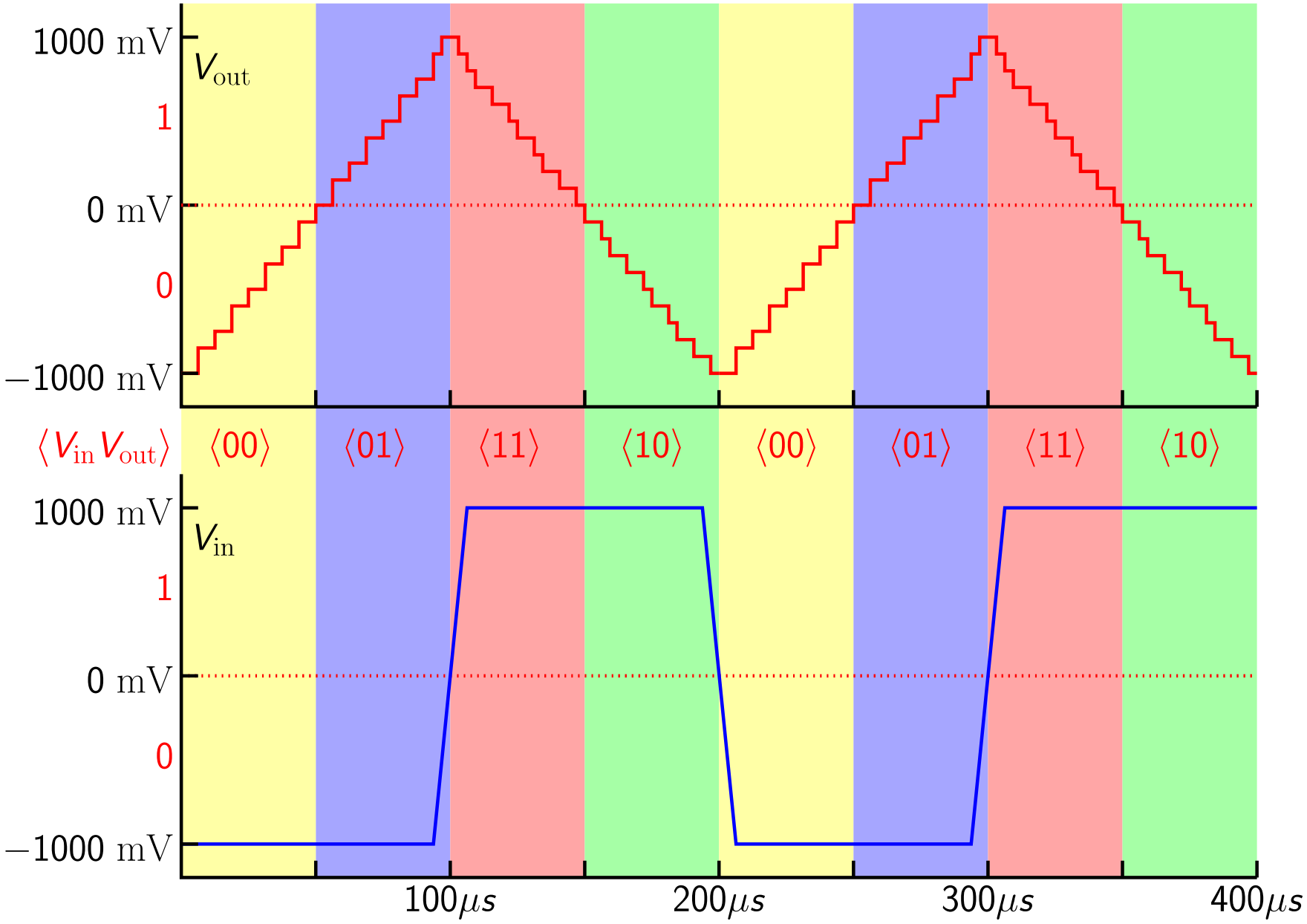
Data Binning



Data Binning

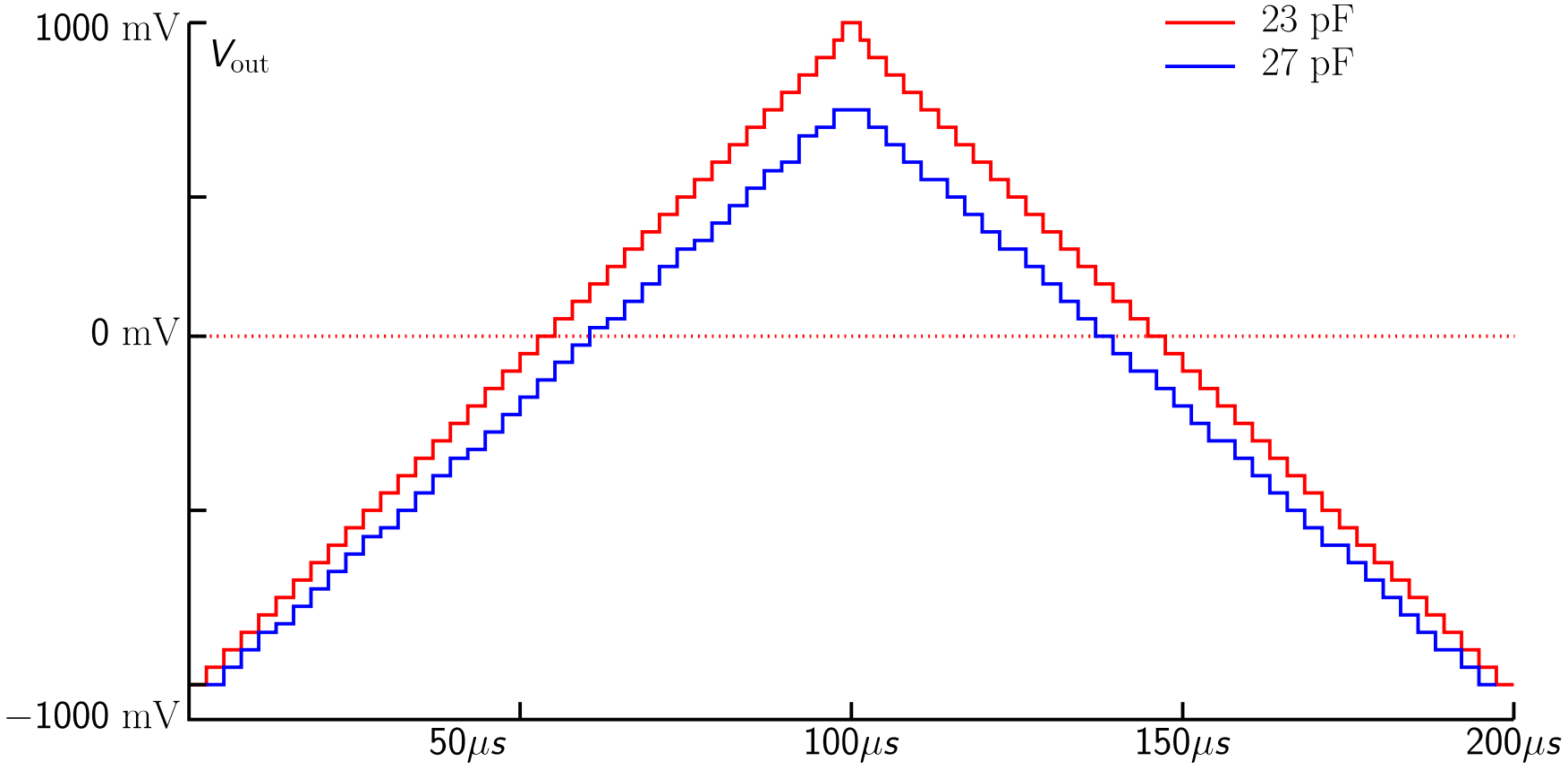


Data Binning

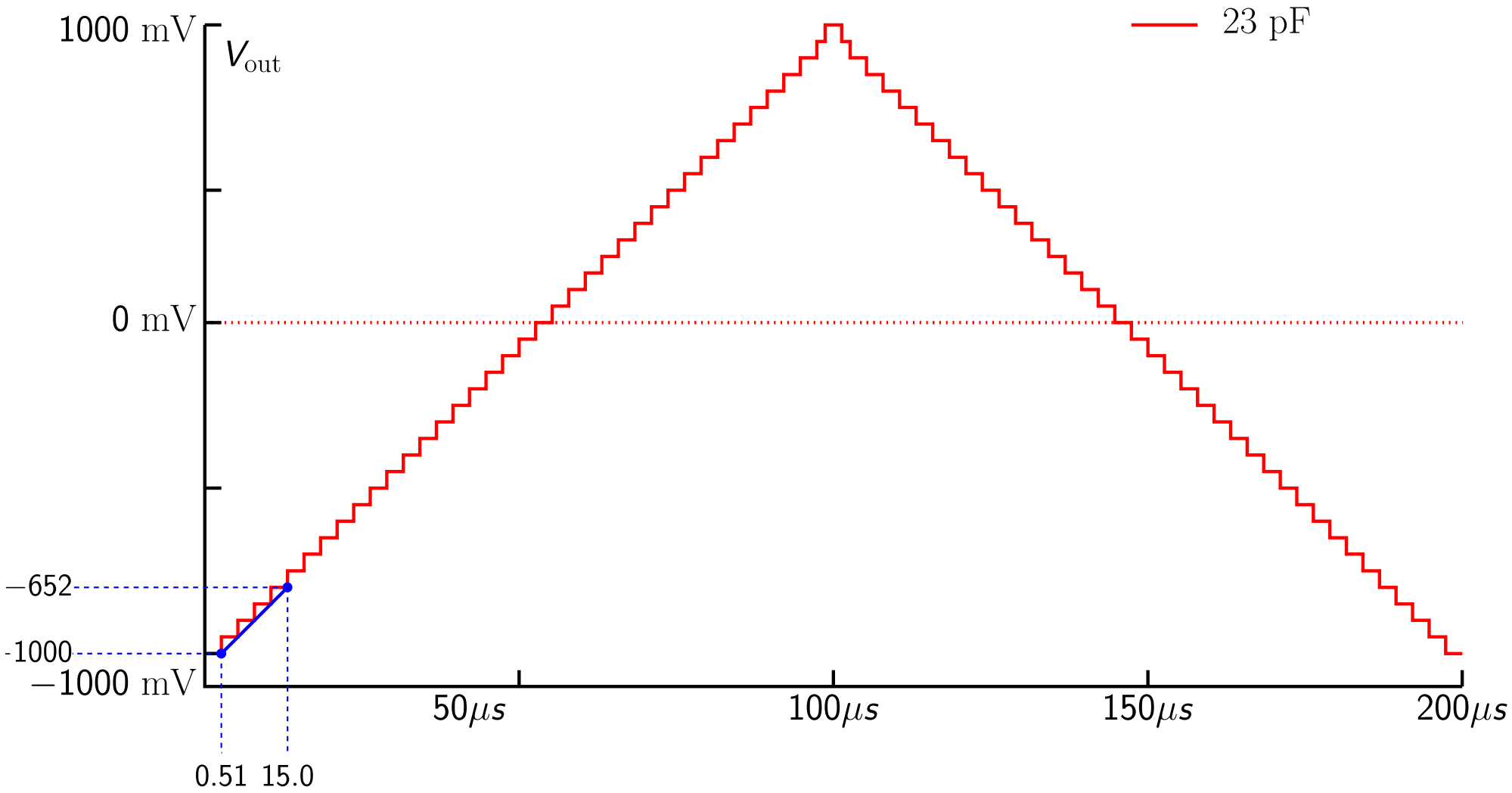


Rate Calculation

- **Rates are calculated for each eligible data point in each bin.**
- **Low pass filtering smooths edge effects and transitory pulses.**
- **Minimum and maximum rates are tabulated for each bin.**



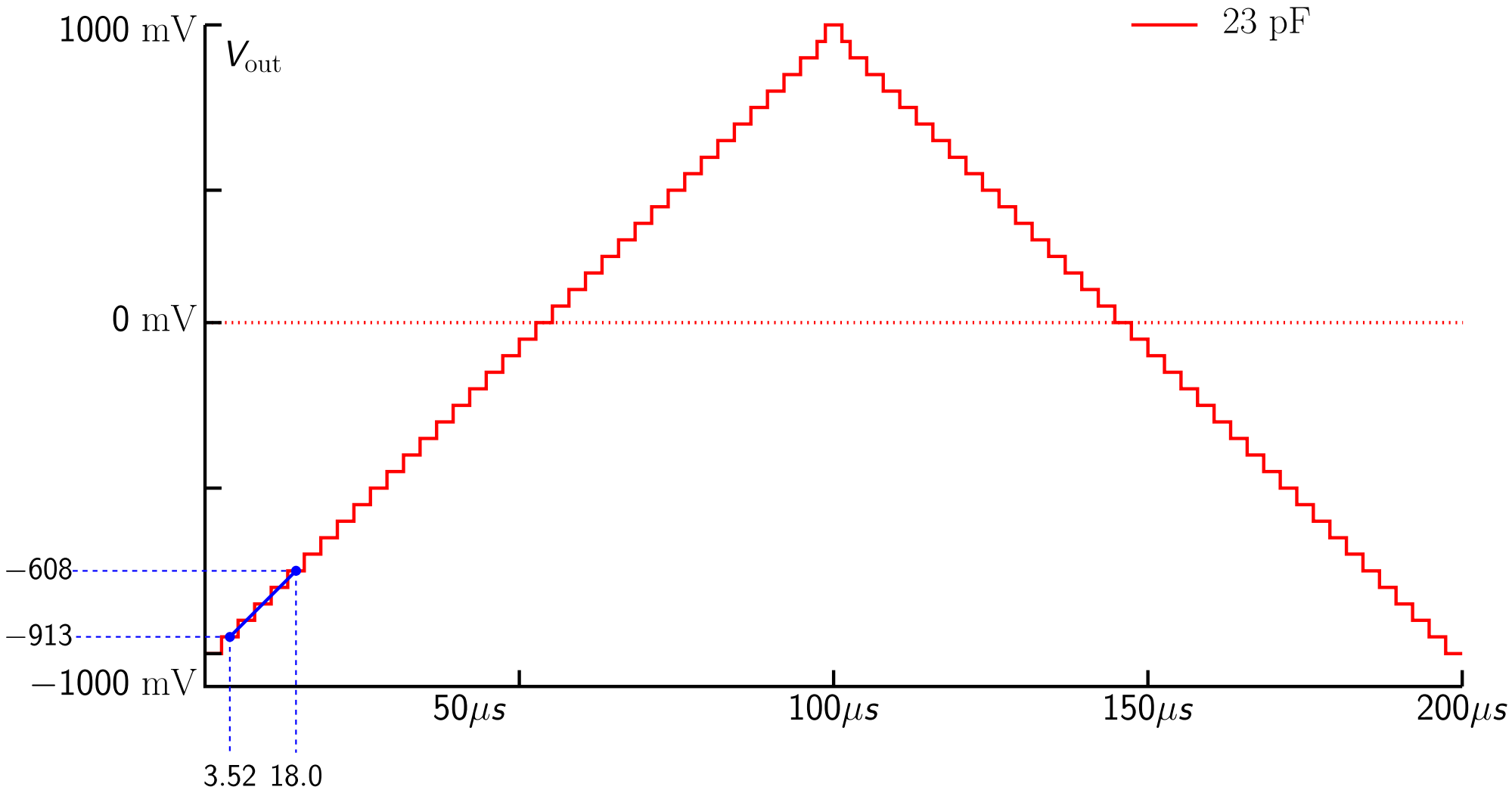
Rate Calculation



$(-652--1000)/(15.0-0.51)=23.93\,mV/\mu s$

$V'_{out\,00}=[23,24]$

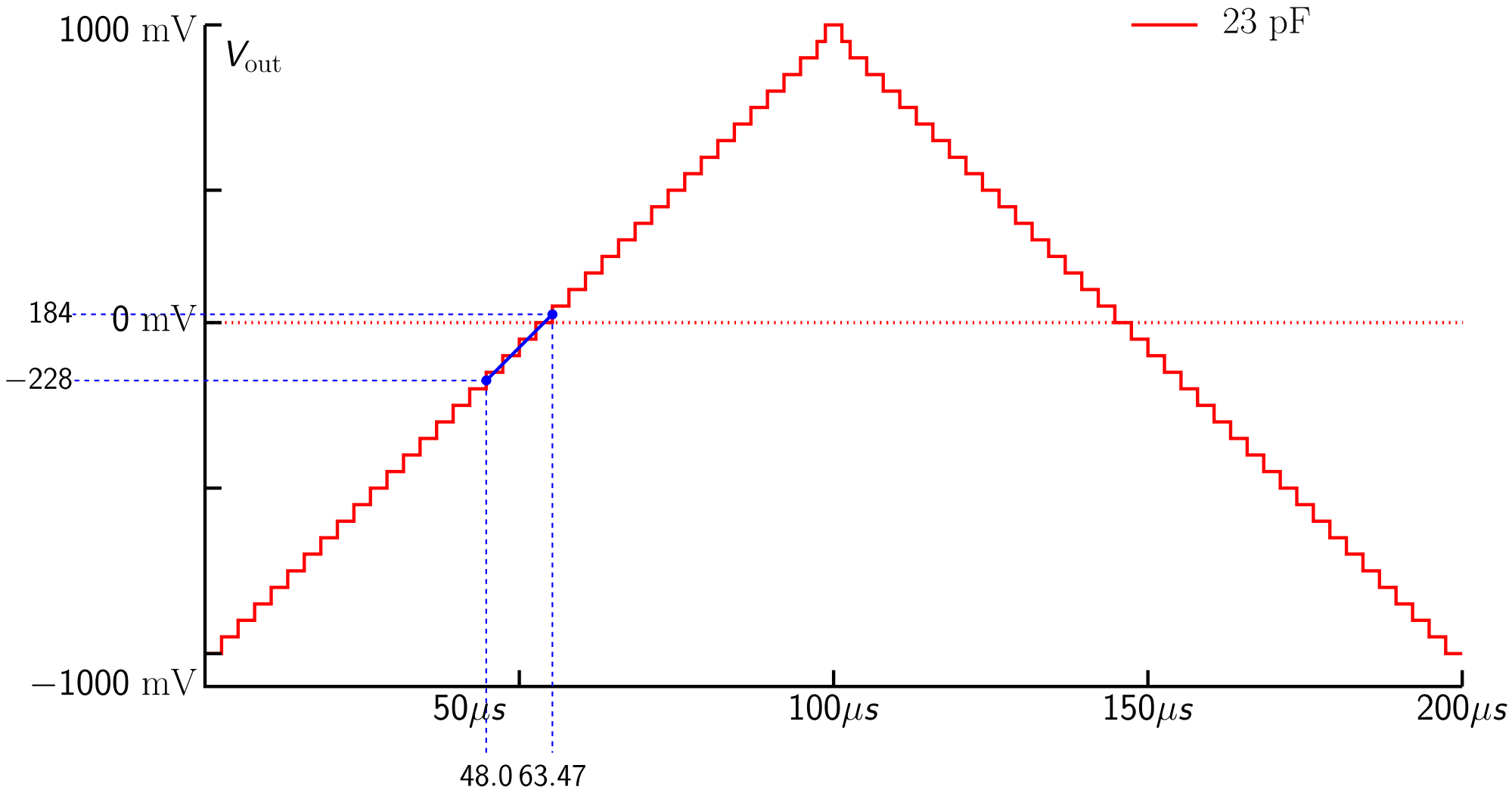
Rate Calculation



$(-608--913)/(18.0-3.52)=21.06 mV/\mu s$

$V'_{out\,00}=[21,24]$

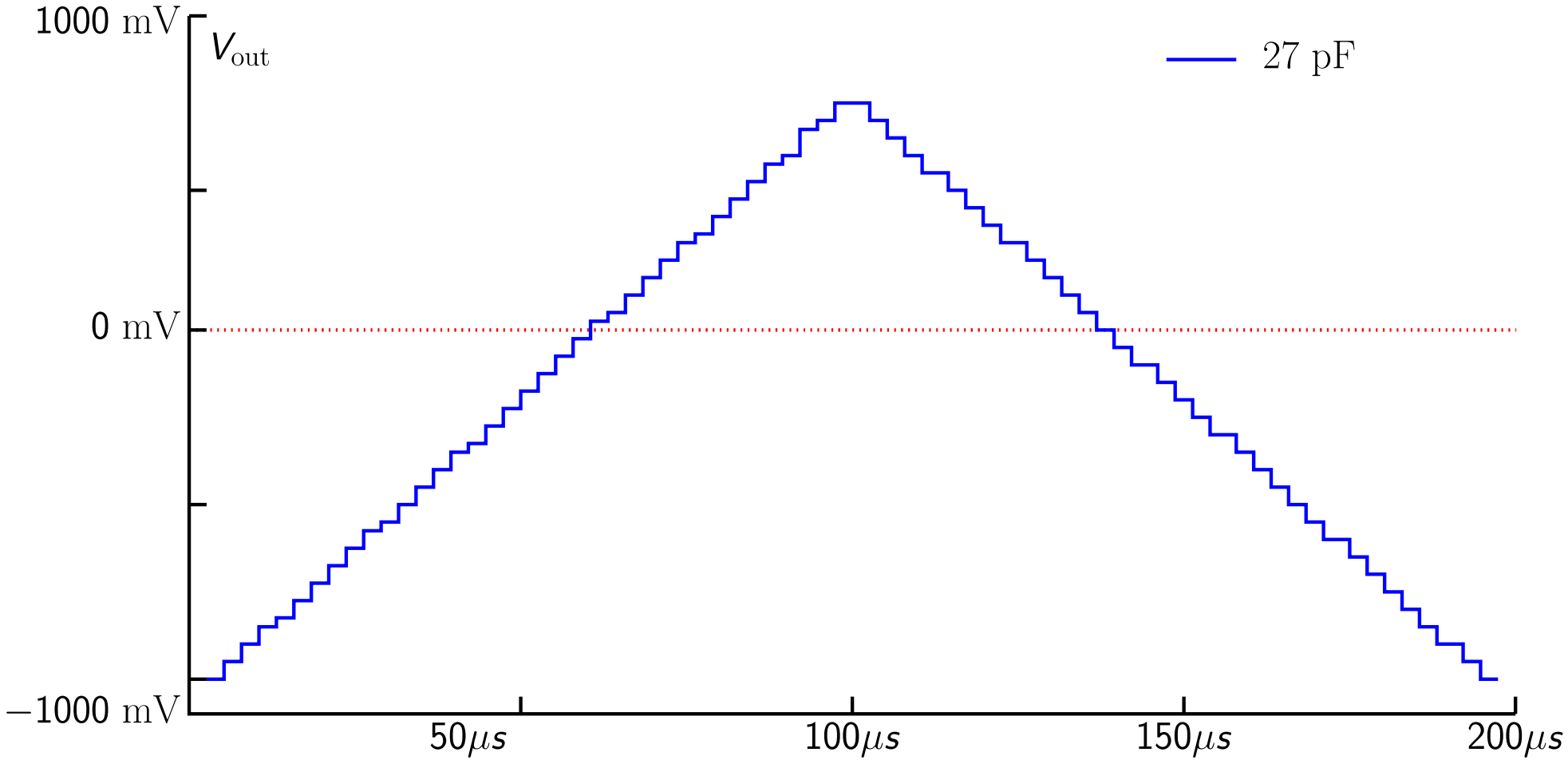
Rate Calculation



No rate calculated.

$V'_{out\,00}=[19,24]$

Rate Calculation



Final rate calculations after $C_2 = 27\,pF$.

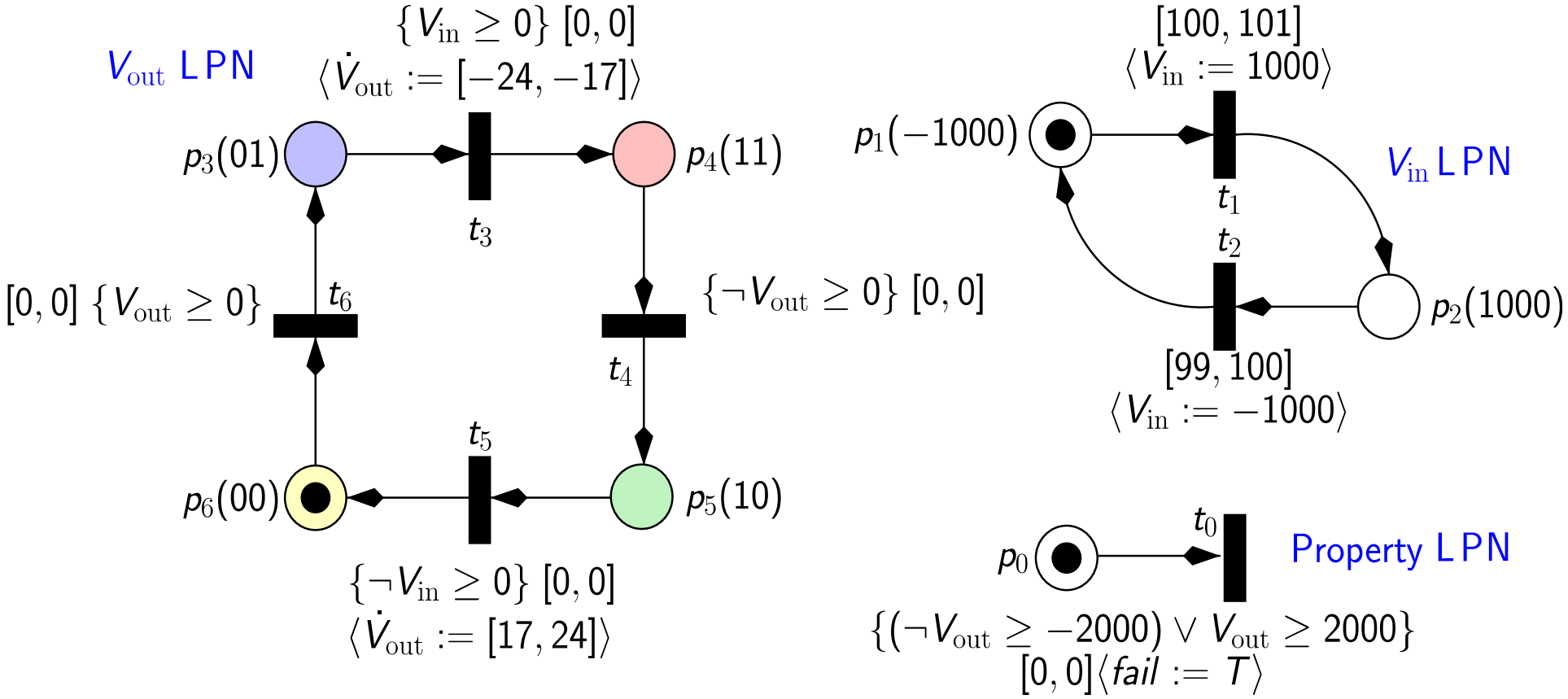
$V'_{out\,00} = [17, 24]$

DMV Variables

- **Stable signals are handled differently to aid efficiency.**
- **Stability is determined by:**
  - **Remaining constant within an epsilon value for a specified time.**
  - **Total percent of the entire signal marked stable.**
- **Delay is calculated for each constant value.**
- **Min/max delay and constant values are extracted.**

Generating an LPN

Initial values=$\{V_{out}=-1000\,mV, V_{\text{in}}=-1000\,mV, fail=F\}$; *Initial rates*=$\{V'_{\text{in}}=0, V'_{out\,00}=[17,24]\}$

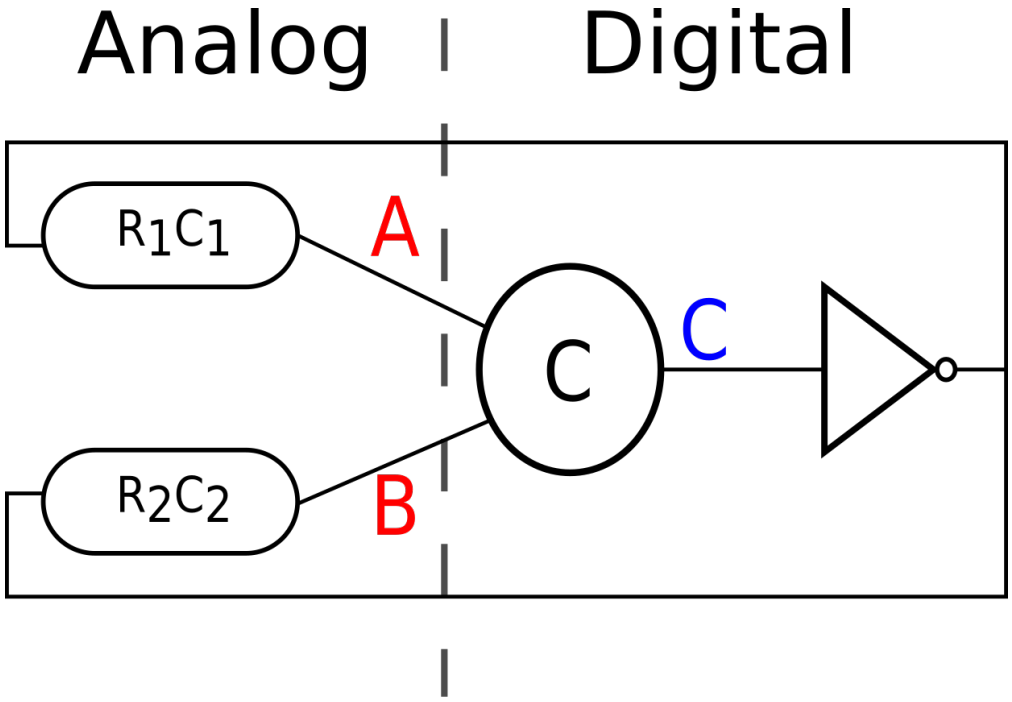


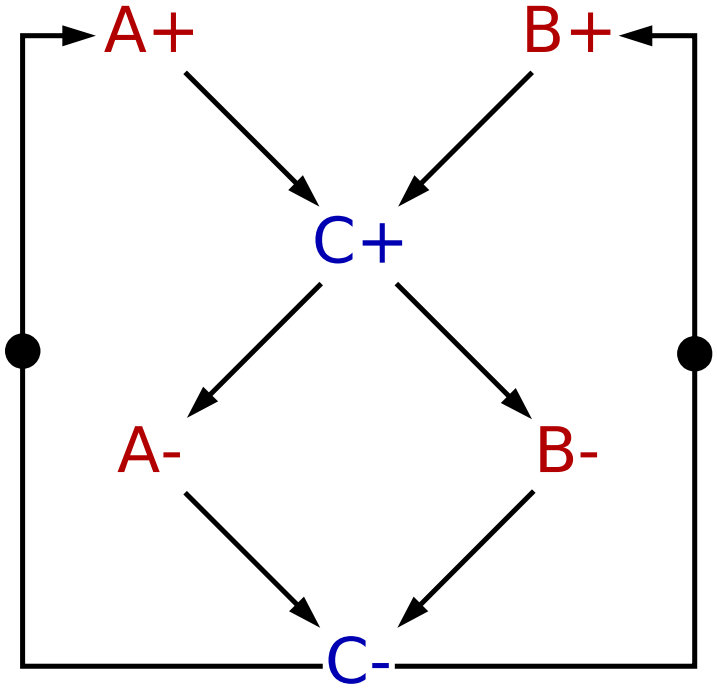
Property Language

- **delay(d) - wait for d time units.**
- **wait(b) - wait until boolean expression, b, becomes true.**
- **waitPosedge(b) - wait for a positive edge on b.**
- **wait(b, d) - wait at most d time units for b to become true.**
- **assert(b, d) - ensure that b remains true for d time units.**
- **assertUntil(b1, b2) - ensure that b1 remains true until b2 is true.**
- **if-else - statement for selections.**
- **always(conditionsList){statements} - continue to execute statements until one of the signals in the list of variables condistionsList changes, then break out.**

LEMA DEMO

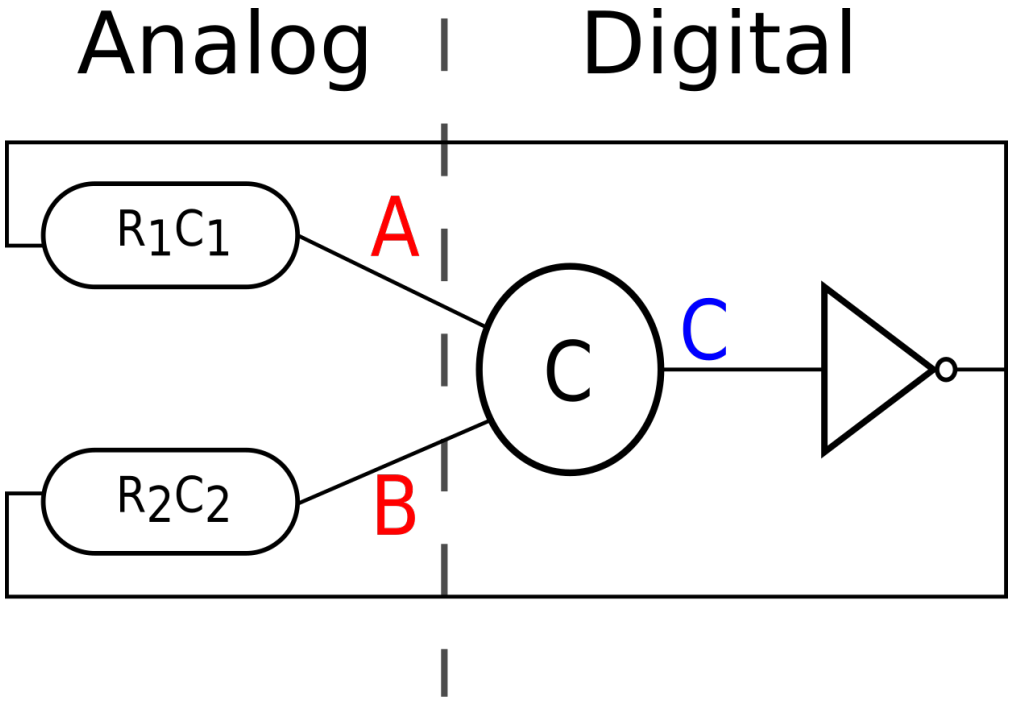
C-element Example

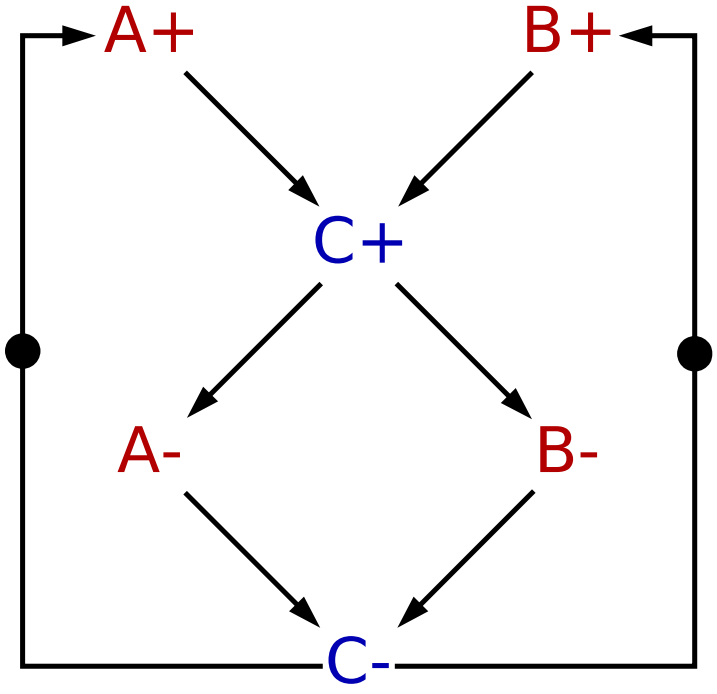




$$R_1 C_1 ? R_2 C_2$$

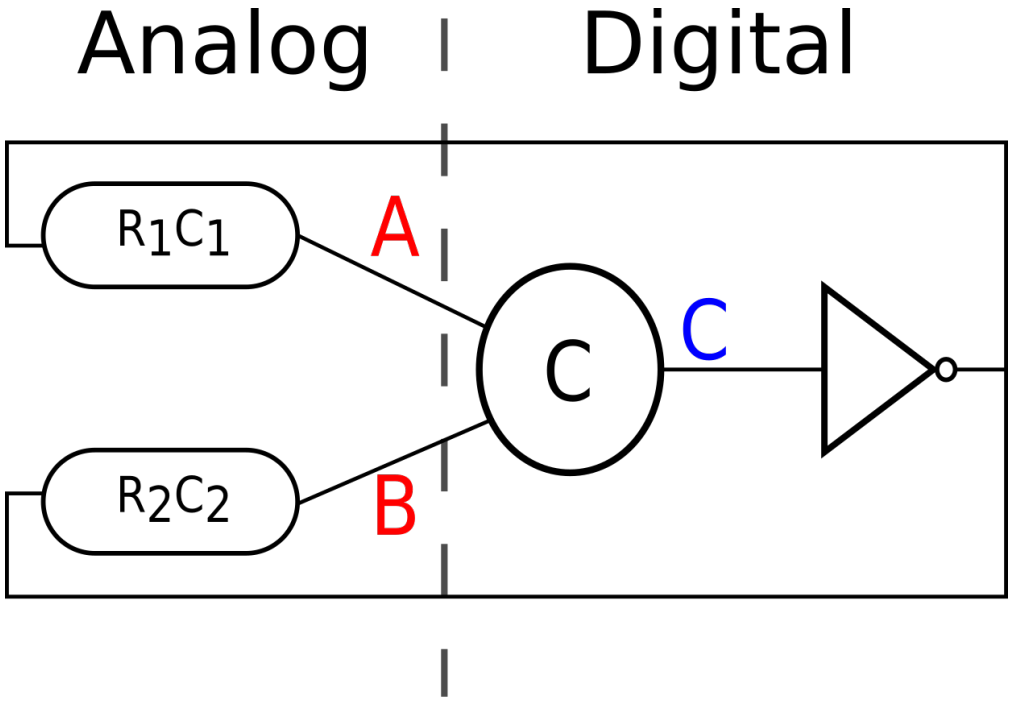
C-element Example

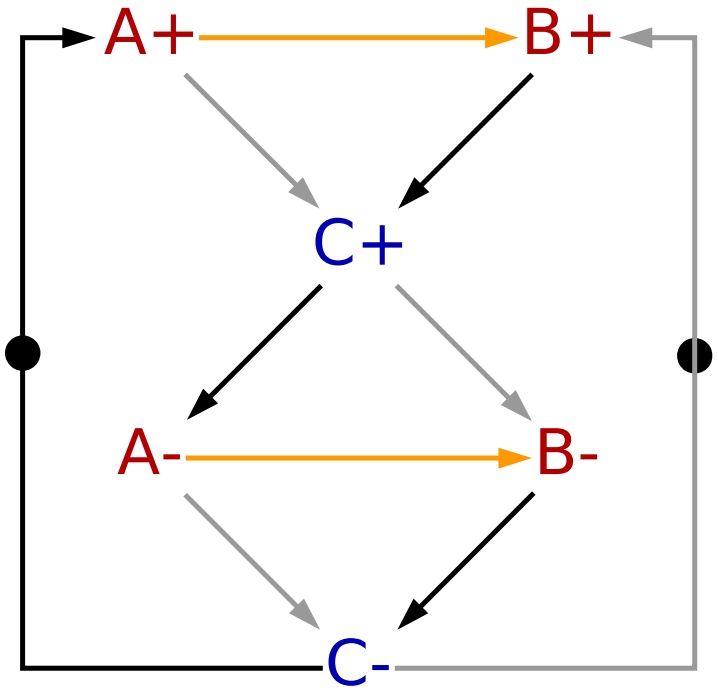




$$R_1 C_1 < R_2 C_2$$

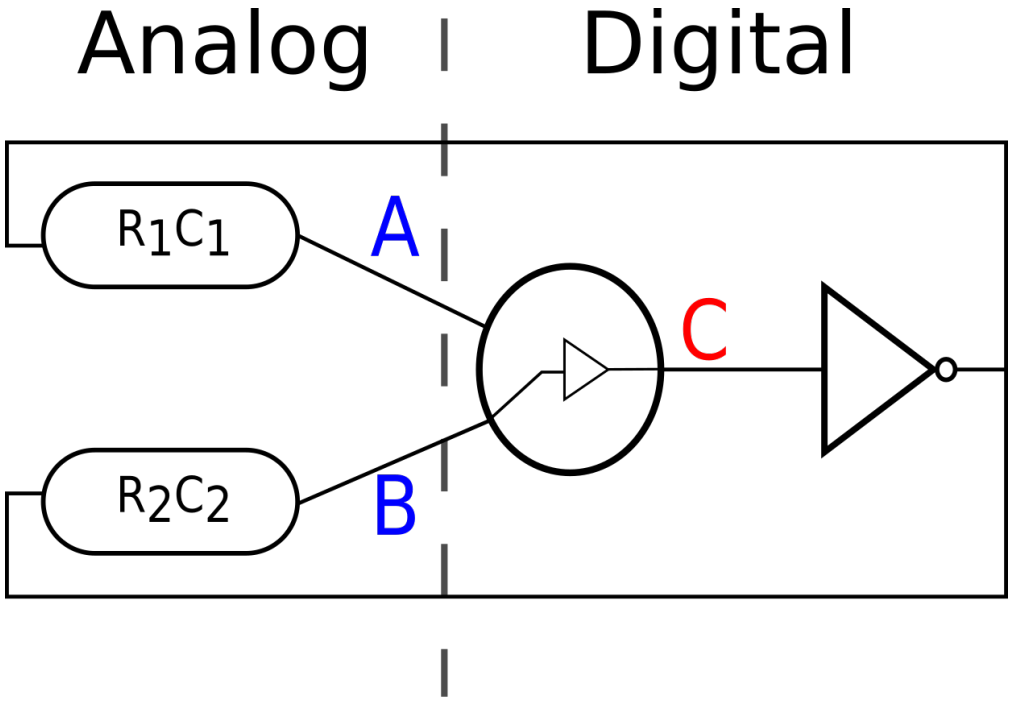
C-element Example

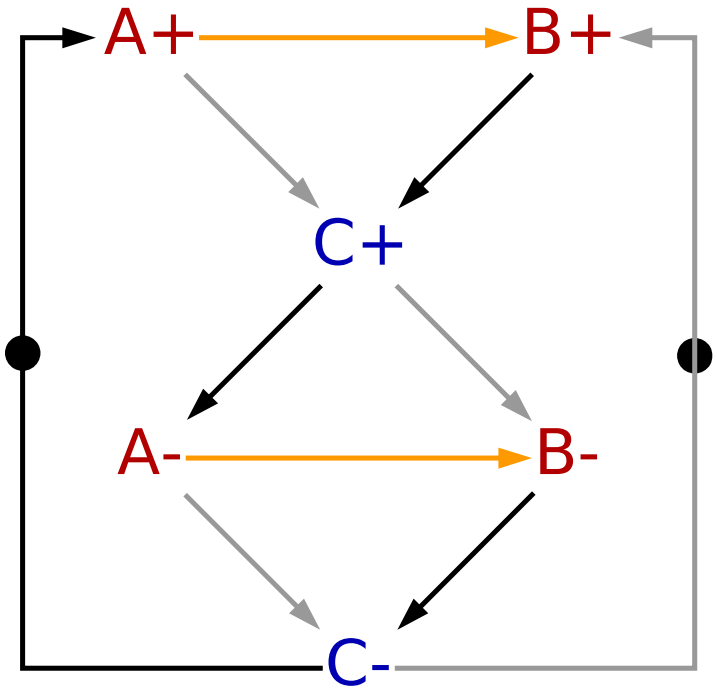




$$R_1 C_1 < R_2 C_2$$

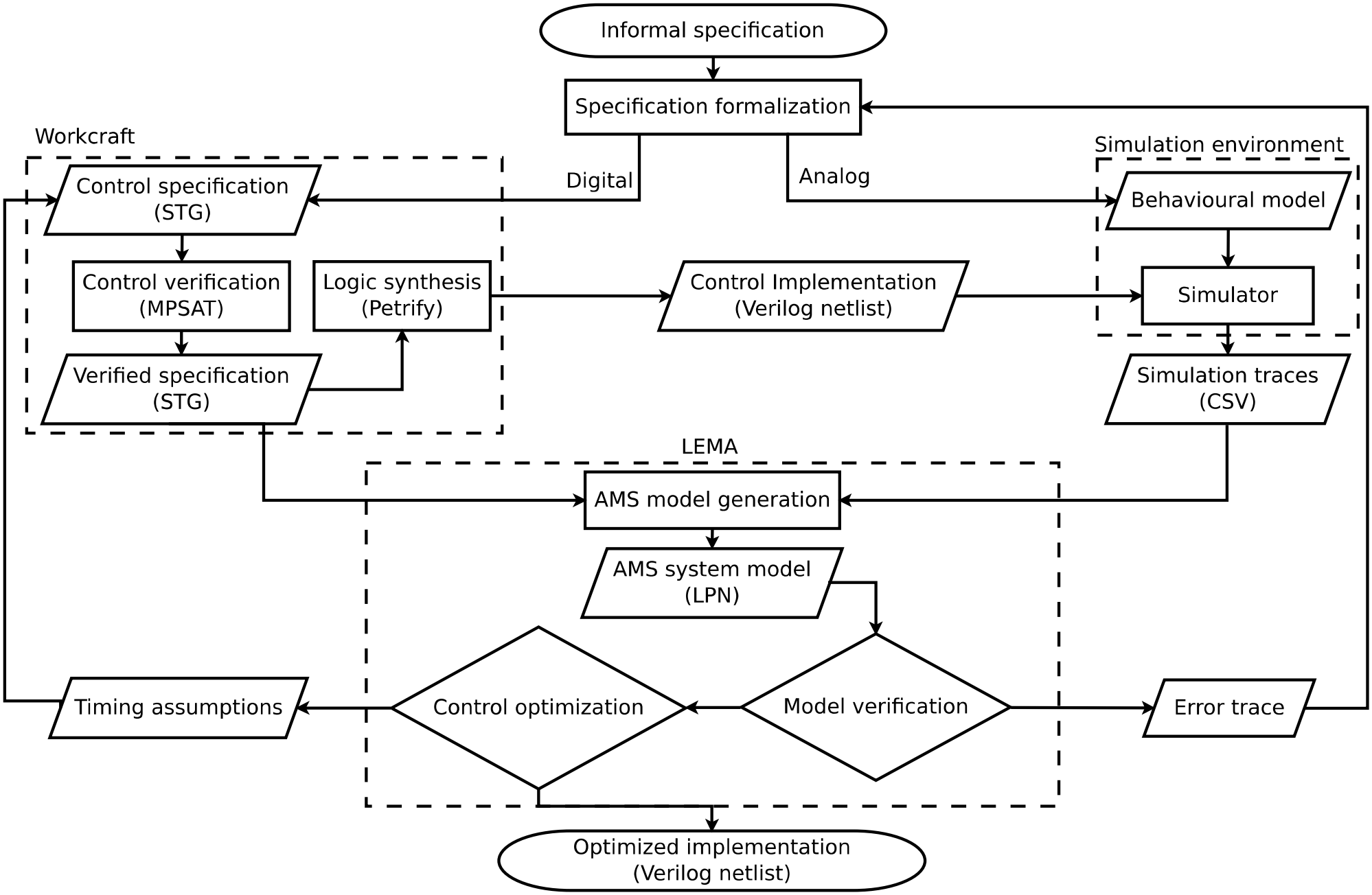
C-element Example



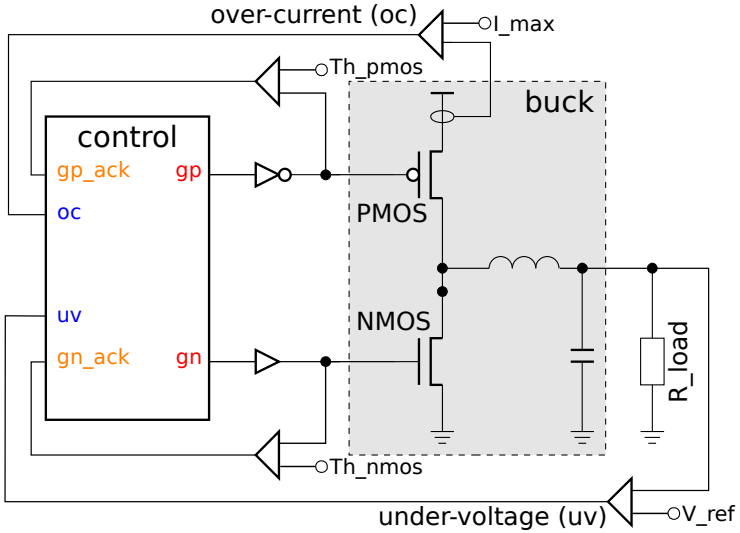


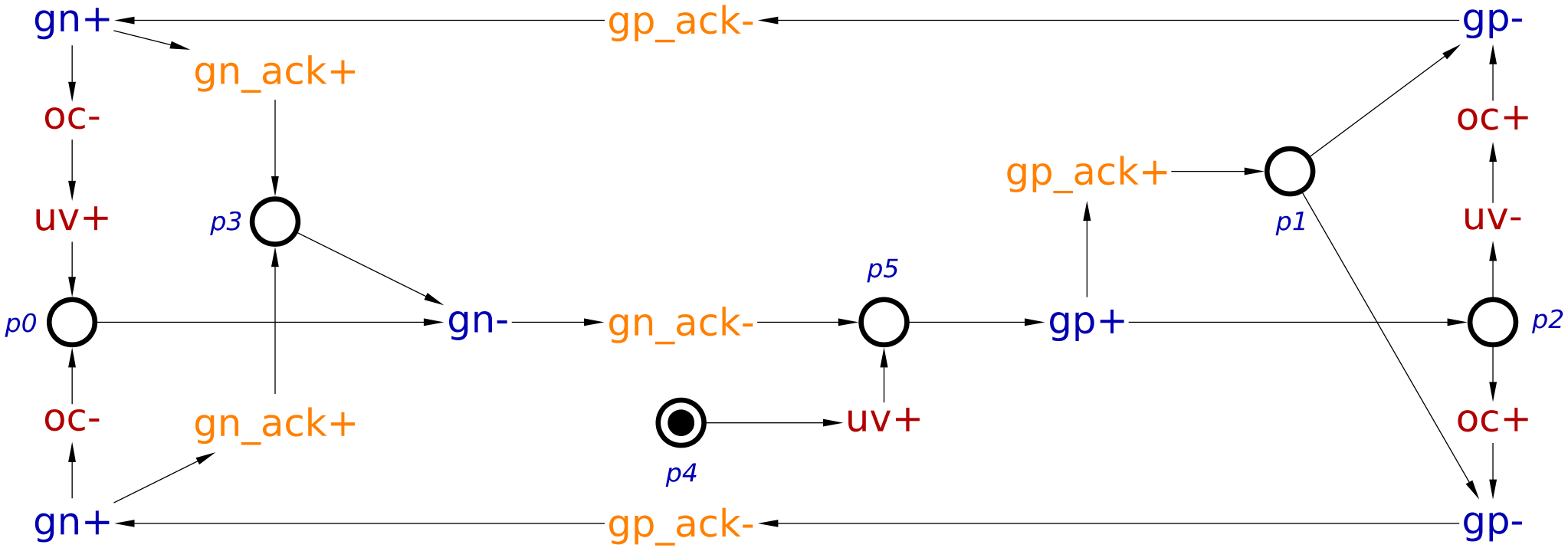
$$R_1 C_1 < R_2 C_2$$

AMS verification workflow

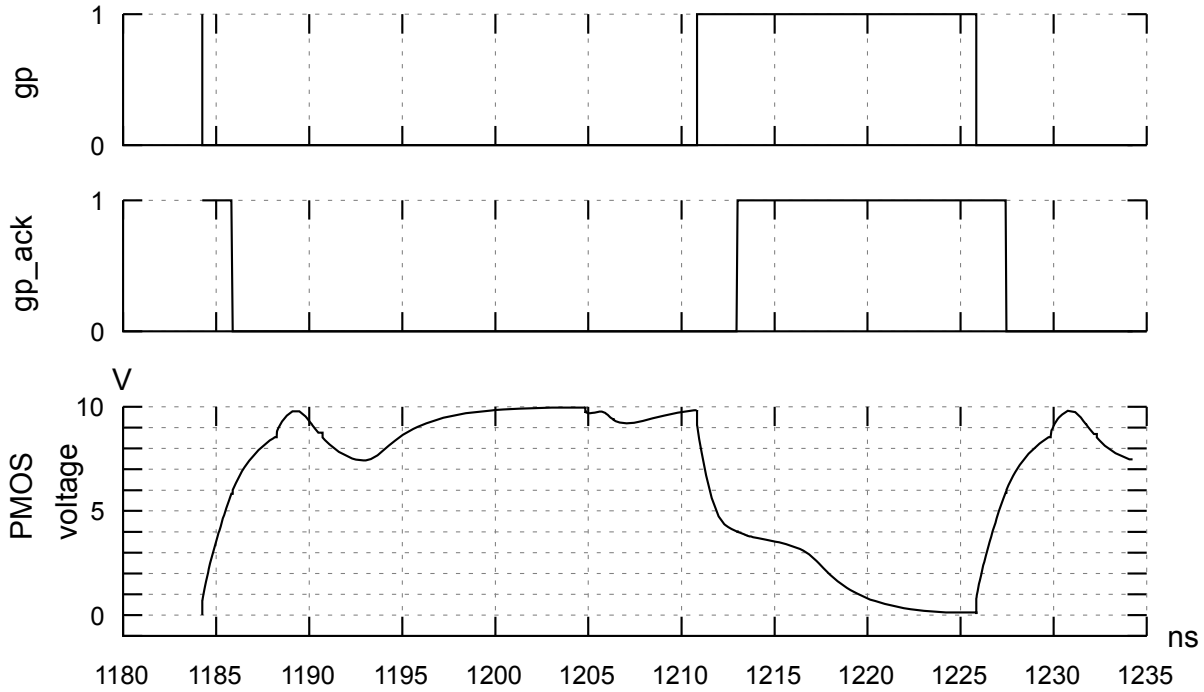


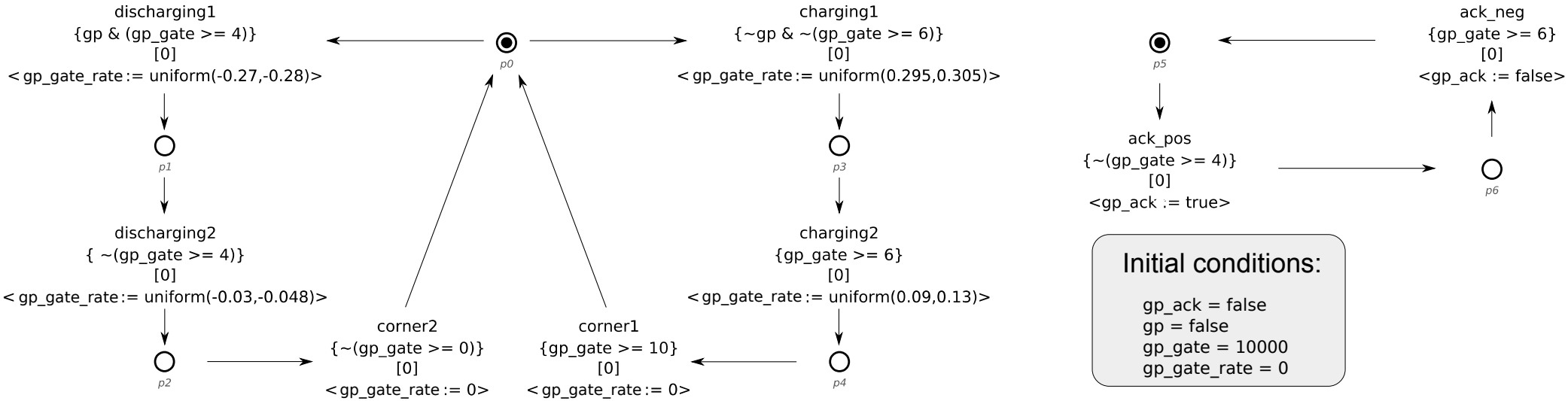
Buck converter





Model generation example



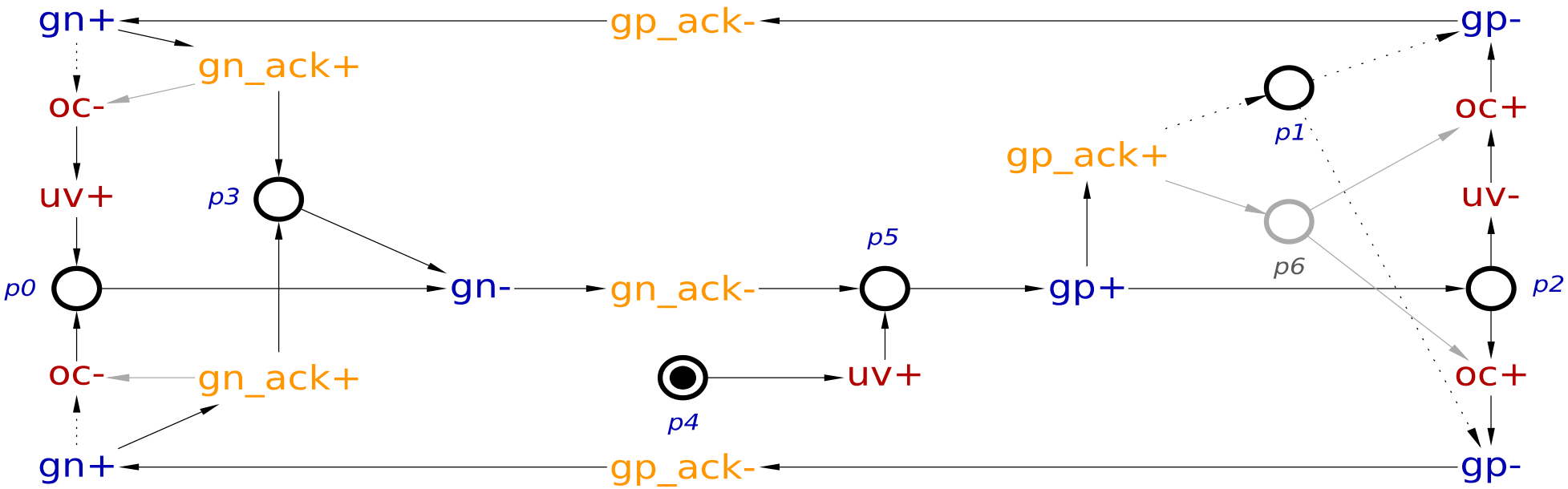


Initial conditions:

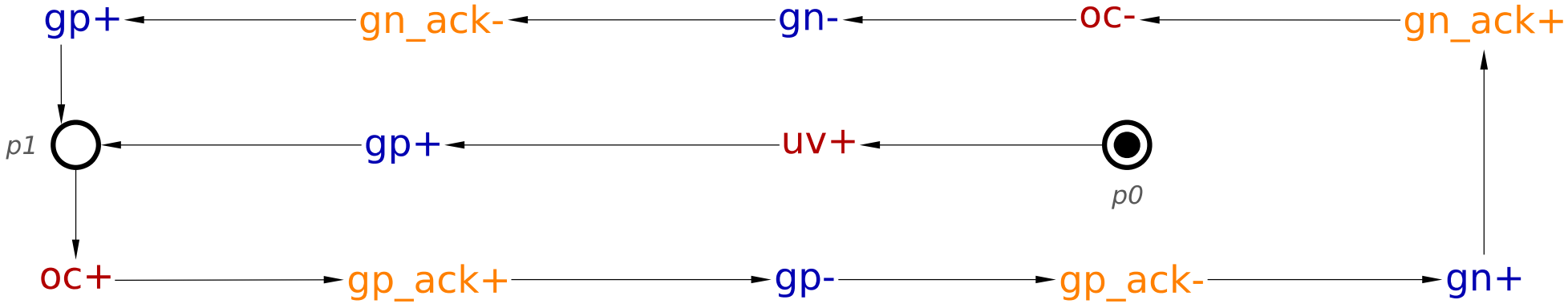
gp_ack = false
gp = false
gp_gate = 10000
gp_gate_rate = 0

Optimized specification

- **Concurrency reduction**



- **Scenario elimination**



Verification challenges

- **Modules partitioning** – trade-off between model’s accuracy and verification speed
- **False positives** – dealing with verification false fail states due to overapporximation
- **Properties expression** – models properties expressed via non-standard language