Verifying Global Convergence for a Digital Phase-Locked Loop

Jijie Wei & Yan Peng & Mark Greenstreet & Grace Yu

University of British Columbia
Vancouver, Canada

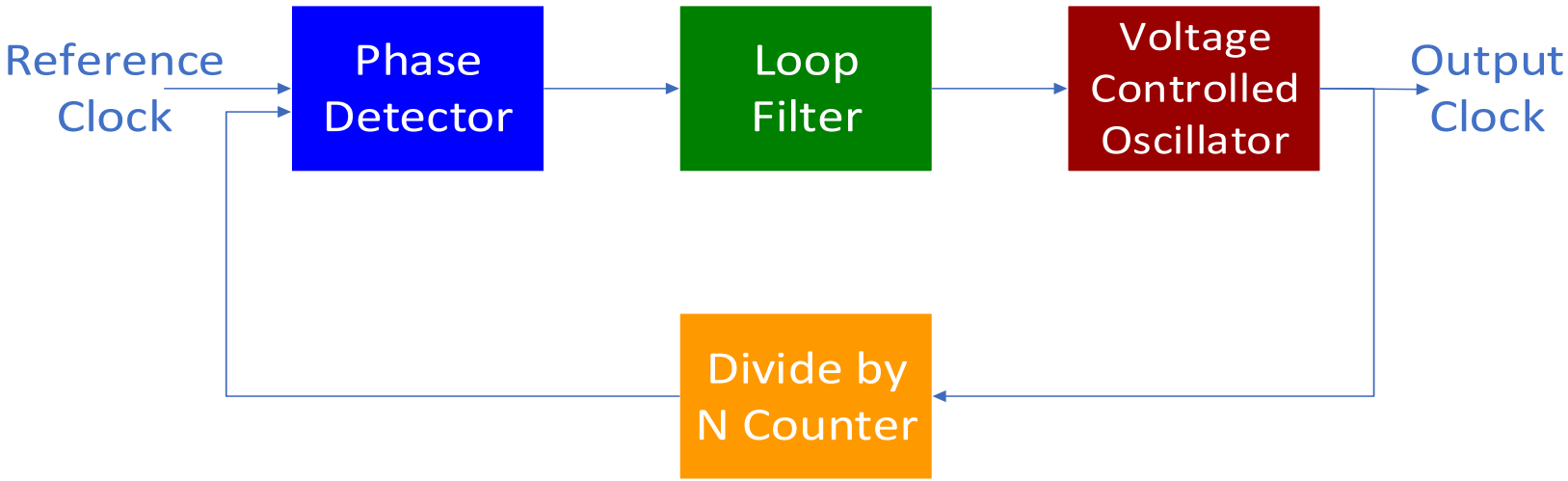
October 22, 2013

Outline

Verifying global convergence of a state-of-the-art digital PLL.

- **Phase-Locked Loop (PLL) introduction**
- Verification as a hybrid automata reachability problem
- Verification as a SMT problem
- Conclusion

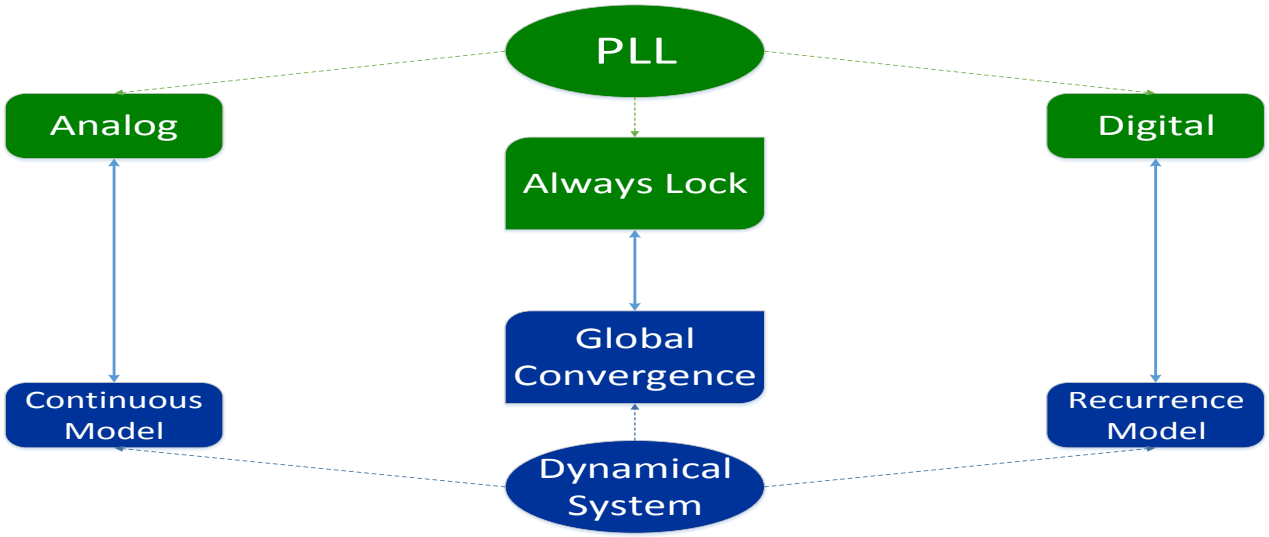
PLL: Block Diagram



A PLL is a feedback control system that, given an input reference clock, it outputs a clock at a frequency that's *N* times of the input clock frequency and aligned with the reference in phase.

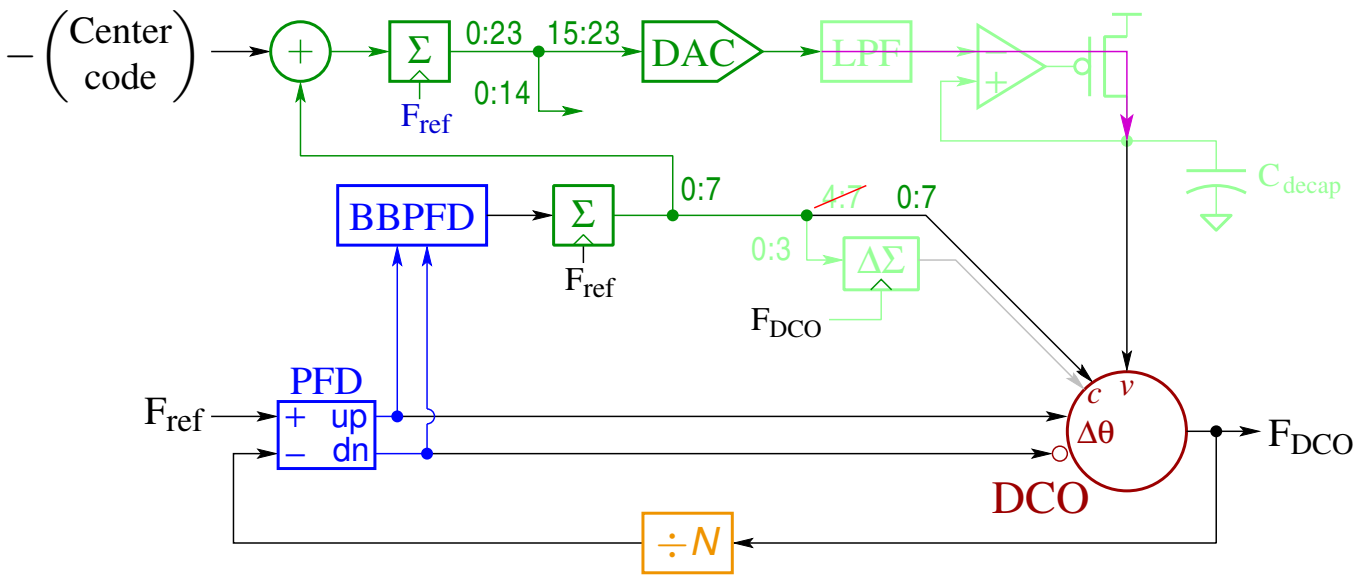
PLLs are ubiquitous in analog and mixed-signal designs. E.g. frequency multiplication for clock-acquisition in high-speed links, modulators and demodulators for wireless communication.

PLL: Global Convergence



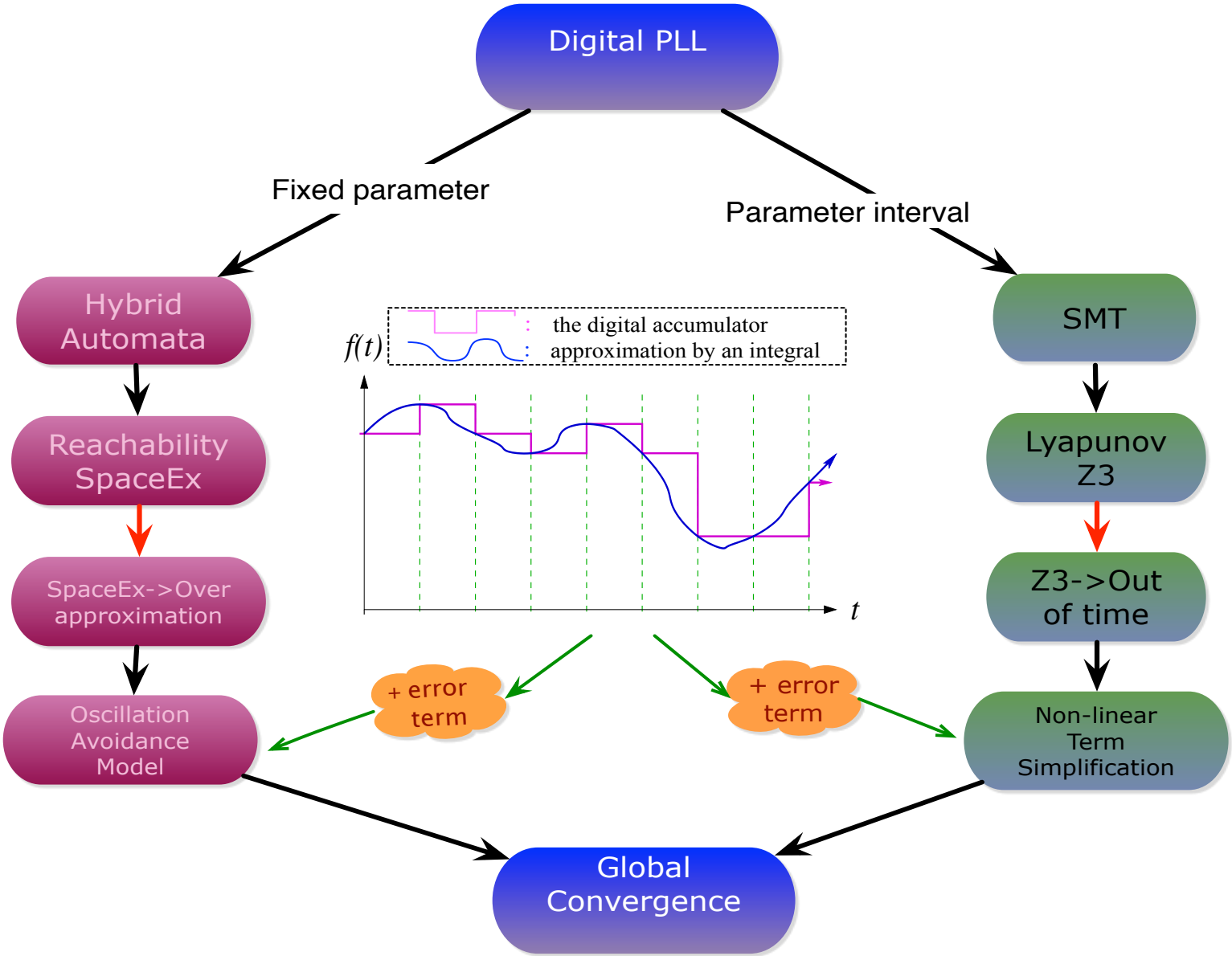
- The most essential function of a PLL is to lock at the right phase and frequency. The more quickly a PLL locks the better.
- Showing convergence from a single initial state requires very long simulation runs – this is intractable for many designs.
- **Global convergence** requires that the PLL locks from all initial states. This needs a formal approach.

A State-of-the-Art Digital PLL (from CICC 2010)[CNA10]



- DCO has three control inputs:
  capacitance setting (digital), supply voltage (linear), phase correction (time-difference of digital transitions).
- Uses linear and bang-bang PFD.
- Integrators are digital.
- LPF and decap to improve power-supply rejection.
- It is impractical to verify global convergence using simulation.

Verification Outline

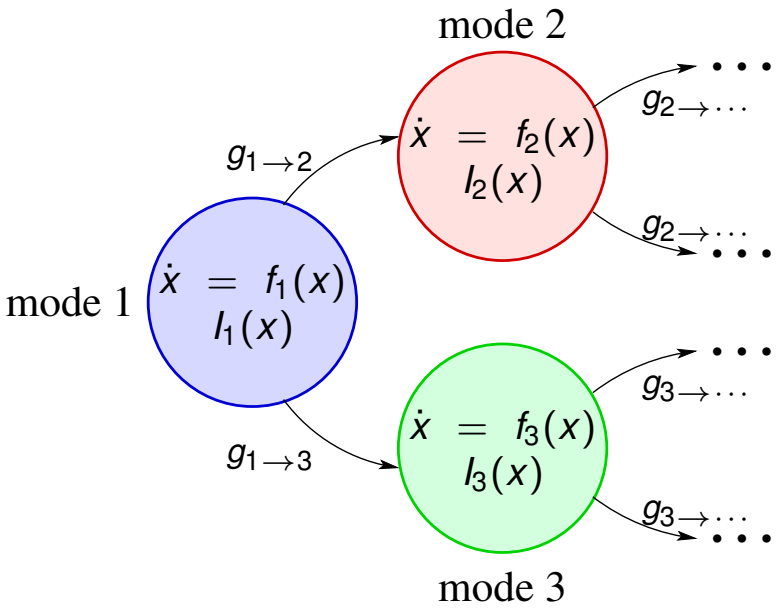


Outline

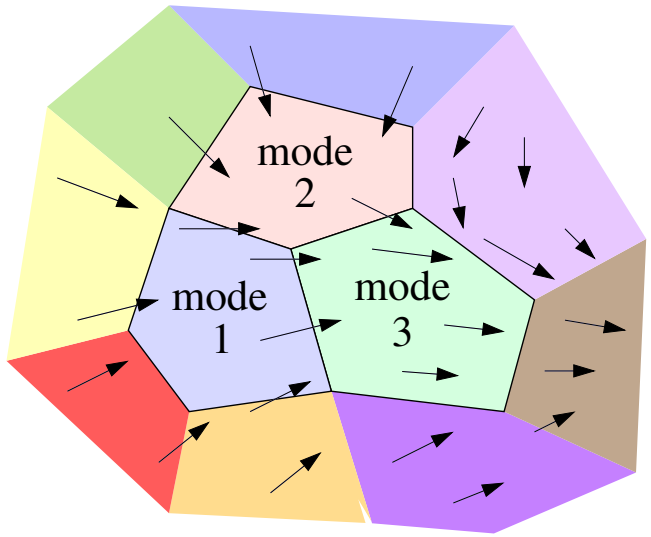
Verifying global convergence of a state-of-the-art digital PLL.

- Phase-Locked Loop(PLL) introduction
- **Verification as a hybrid automata reachability problem**
  - **Digital PLL overview**
  - **Modelling as a product hybrid automaton**
  - **Verifying the digital PLL in three phases**
  - **Modelling and Verifying the reset delay**
- Verification as a SMT problem
- Conclusion

Hybrid Automata Approach



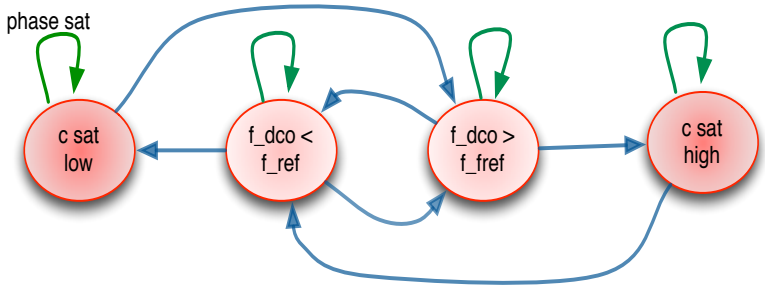
A Hybrid Automaton



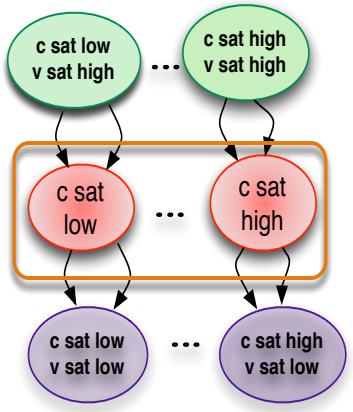
Corresponding Piecewise Model

- Each mode has its own continuous dynamics: $\dot{x} = f_m(x)$.
- Each mode has an **invariant**, I_m. The automaton must transition to another mode before I_m is violated.
- Transitions are guarded by conditions on the continuous state.

Hybrid Automata product



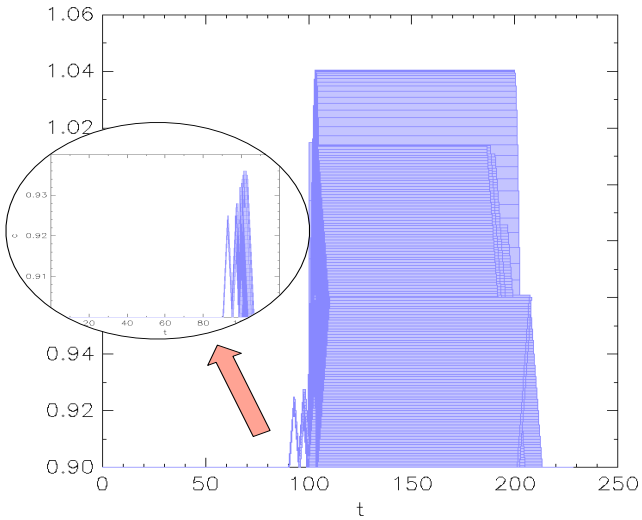
Hybrid Automata product of BBPFD and PFD



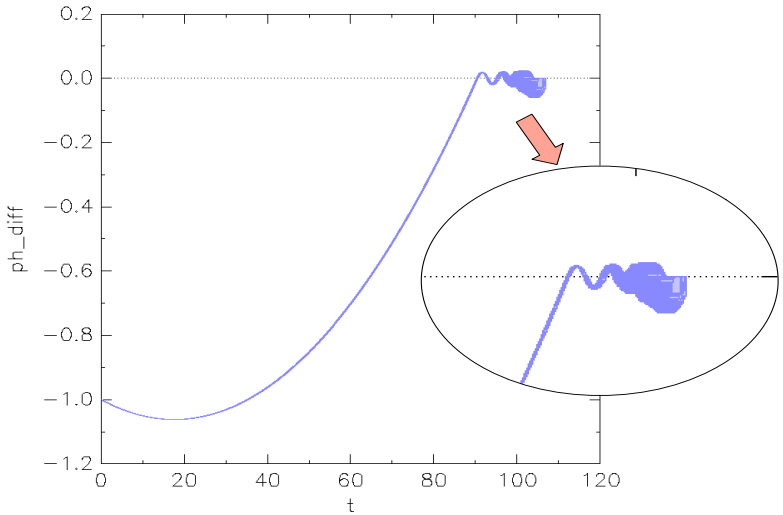
Hybrid Automata product of DCO

- DCO is linearized and v_{ctrl} is divided into 7 overlapping intervals.
- v_{ctrl} has saturation modes, so 3 modes for each v_{ctrl}.
- Because of the BBPFD, C has 4 modes: up, down, saturated low and saturated high.
- PFD: 1 mode, with self-loop
- Product automaton has 84 modes.

Difficulty



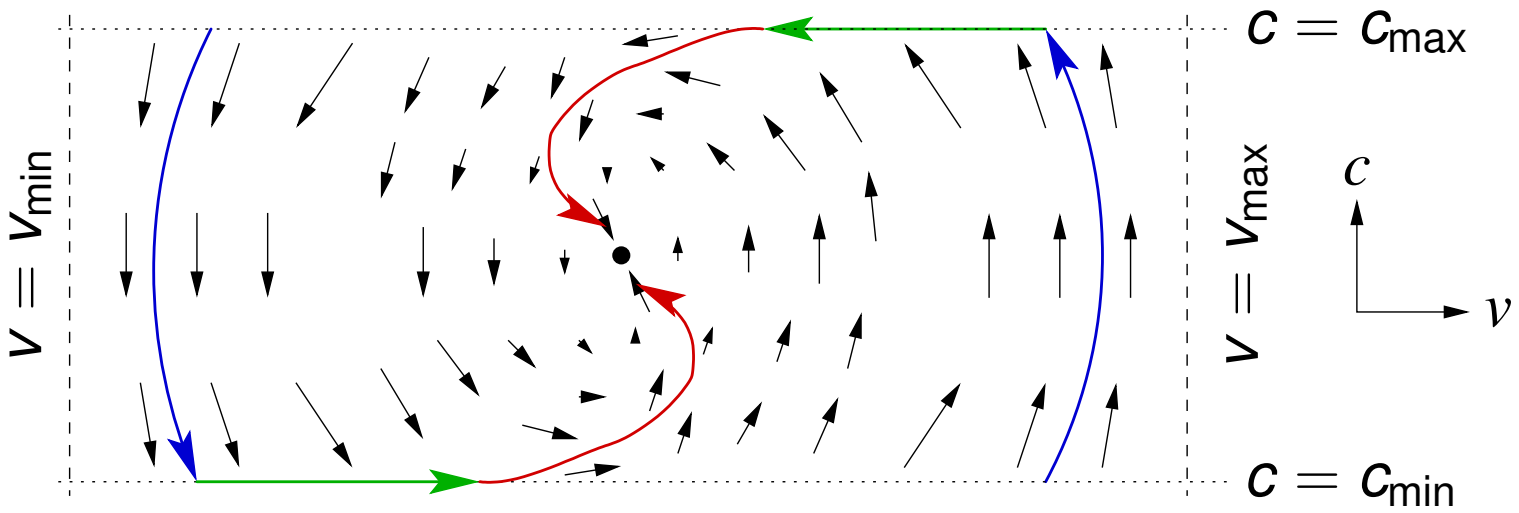
C VS time



Ph_diff VS time

- BBPFD causes limit-cycle oscillations which create thousands of transitions on path to convergence.
- SpaceEx[FLGD+11] uses a support function representation that introduces large over-approximations on mode changes $\Rightarrow$ **false-negatives**.

Verification Strategy



Identify the phases the digital PLL goes through when locking.

Phase 1: Large frequency difference – C heads to limit and saturates.

Phase 2: Large frequency difference, C-saturated, the linear phase path acquires frequency lock.

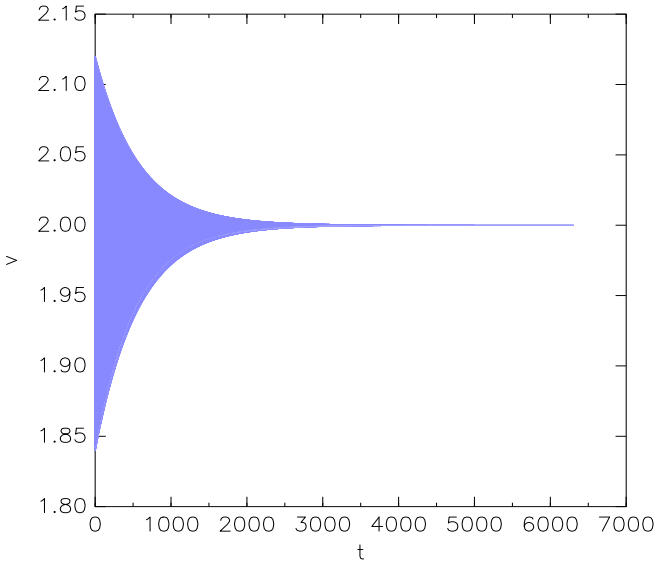
Phase 3: $\Delta\theta$ changes sign – C and v_{ctrl} converge to phase and frequency lock.

Phase 3: avoiding mode transitions

- C leaves its saturation mode
- make a change of variables, let

$$w = f_{dco} - f_{fref}$$

- the convergence of w is strong enough that SpaceEx shows convergence, even with its over-approximations for mode transitions.
- SpaceEx shows that the reachable region of w shrinks
- Given the invariant that w will shrink asymptotically, convergence of v_{ctrl} can be obtained

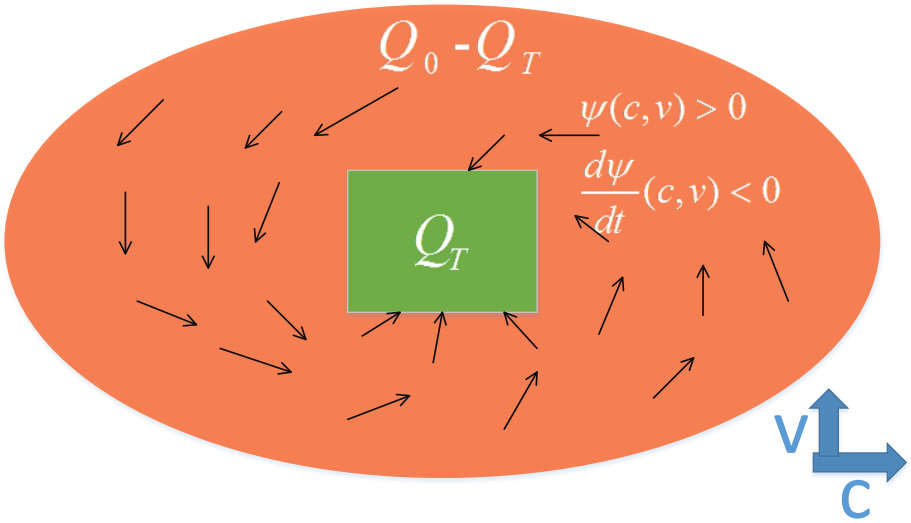


Outline

Verifying global convergence of a state-of-the-art digital PLL.

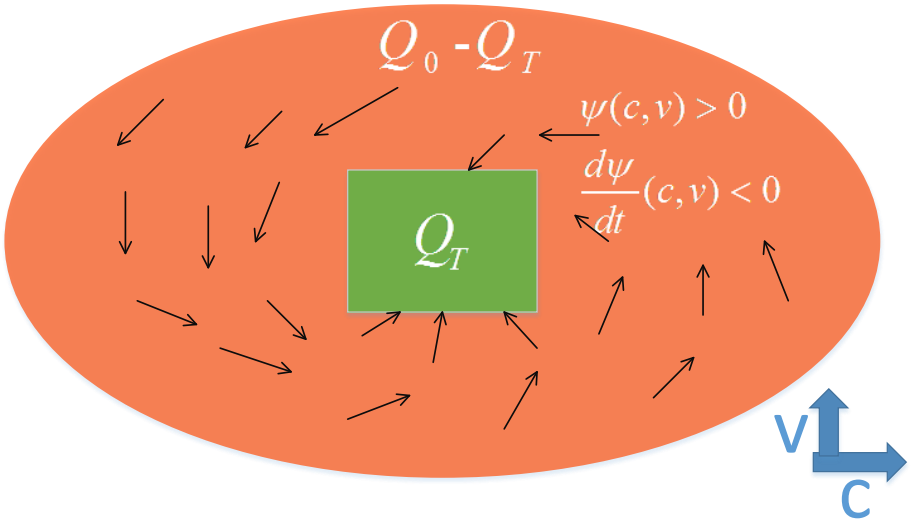
- Phase-Locked Loop(PLL) introduction
- Verification as a hybrid automata reachability problem
- **Verification as a SMT problem**
  - **Verifying a simplified model with Lyapunov functions**
  - **Improved model with quantization error and parameter ranges**
- Conclusion

Lyapunov Function



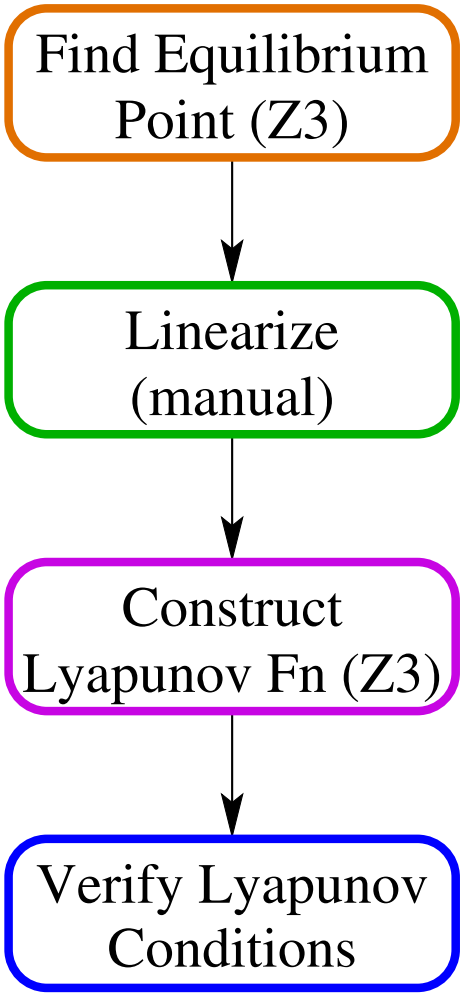
- A Lyapunov function defines a **potential** for the system.
  - If this potential is always positive and decreasing outside of the target region, then the system will eventually reach the target.
  - A Lyapunov function is the continuous counterpart of a **ranking function** for discrete progress arguments.
- A simple function for linear ODEs: $\dot{x} = Ax$:

Lyapunov Function



- A Lyapunov function is the continuous counterpart of a ranking function.
- A simple function for linear ODEs: $\dot{x} = Ax$:
  - Let P be the solution of $A^T P + PA = -I$. A possible Lyapunov function becomes $\Psi(x) = X^T PX$.
  - By construction, P is symmetric.
  - If P is positive definite, then the system $\dot{x} = Ax$ globally converges to $x = 0$

Verifying a Simplified Nonlinear Model using Z3[DMB08]



Strong candidate for automation using symbolic/programatic differentiation

This step guarantees soundness of the entire process.

- Z3 easily proves global convergence.
- Fixed procedure: promising for automation.

Improving the Model

- Z3 easily proves the simple system (just shown)
- We added details to the model to make it more realistic
  - Parameters with ranges
  - Quantization error: we approximate discrete sums in the real digital PLL with integrals
  - Both ranges and quantization
- In all of these, Z3 would run longer than we had patience
- *Solution*: manually simplify terms in the proposed Lyapunov function
- The approach remains sound because Z3 checks the Lyapunov conditions, it doesn't matter how we came up with the Lyapunov function.

Adjust the Proof in Z3

Parameters with ranges:

Inequality ranges for parameters $\alpha \in 1 \pm 0.2$, $\beta \in 1 \pm 0.2$, $v_0 \in 1 \pm 0.2$ and $c_0 \in 1 \pm 0.2$.

Strategy: Check where the parameters are used, try simplify non-linear part. e.g. we replace some of the parameters in the Jacobian matrix with its nominal value.

Example:

Simplifying an element of the Jacobian matrix:

$$\frac{g_1 \alpha}{c_0 + \beta c_{code}} \rightarrow \frac{g_1}{c_0 + \beta c_{code}}$$

The approximation is based on the observation that $\alpha \in 1 \pm 0.2$.

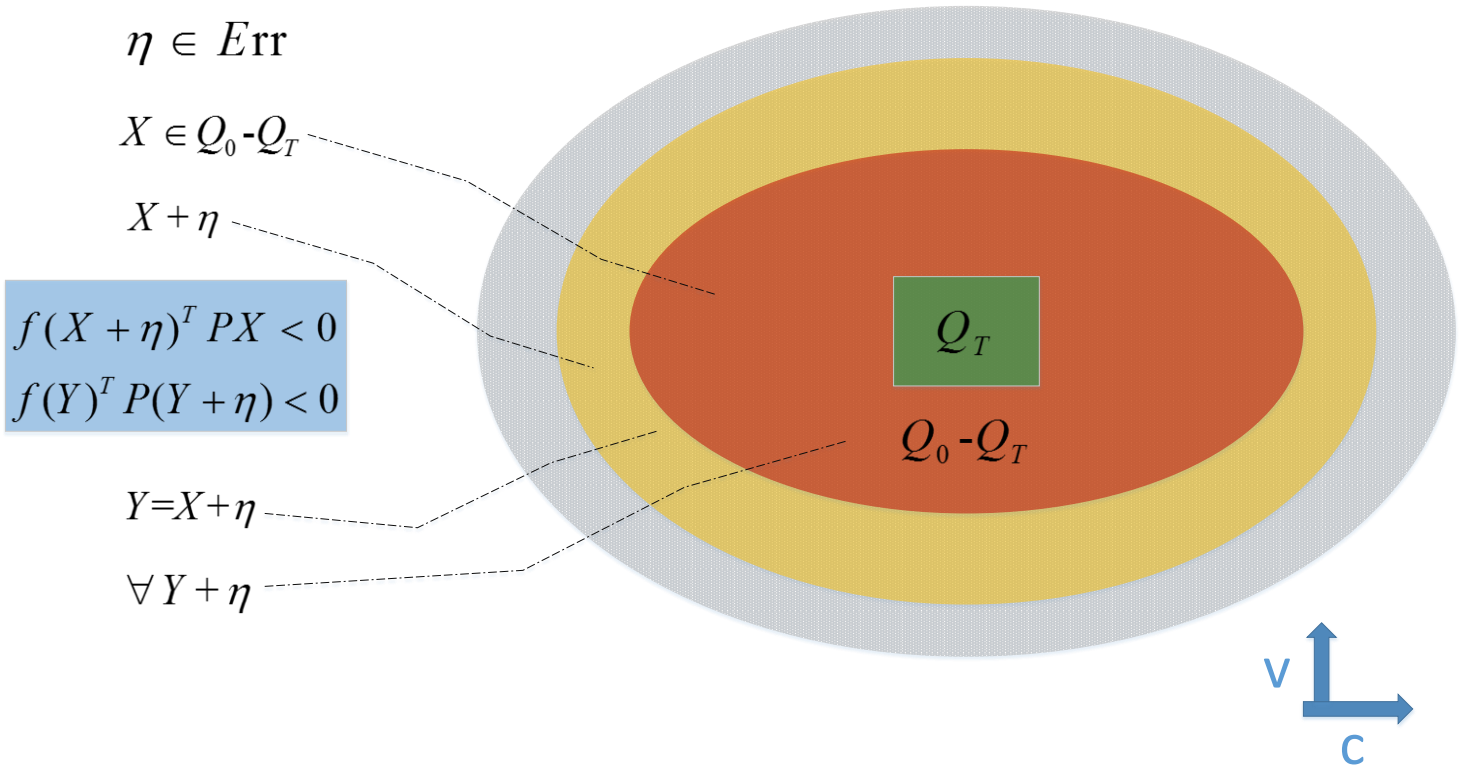
Z3 happily proved the conditions, again.

Adjust the Proof in Z3

Adding quantization error:

Need to show: $\forall x \in Q_T - Q_0$ and $\forall \eta \in Err$, $f(x+\eta)^T Px < 0$, where f is nonlinear. This creates nonlinear terms for the components of η.

Strategy: Choose any $y = x + \eta$ and any $\eta \in Err$. If $y + \eta \in Q_0 - Q_T$, then need to show $f(y)^T P(y+\eta) < 0$



Conclusion and Future Work

- We verified **global convergence** for a *state-of-the-art*, digital phase locked loop (PLL) using piecewise linear differential inclusions with **SpaceEx**.
- Using a simplified model, we showed convergence where specifications for components are **interval bounds** using **Z3**.
- Future Work:
  - Provide bounds on lock time.
  - Complete models for low-pass filter and Delta-Sigma modulator.
  - Examine other digital PLL architectures to assess the reusability and automatability of this verification.
  - *Component validation*: formalize the connection between the models used and those used in other phases of the analog and mixed-signal design process.

Conclusion and Future Work

- We verified **global convergence** for a *state-of-the-art*, digital phase locked loop (PLL) using piecewise linear differential inclusions with **SpaceEx**.
- Using a simplified model, we showed convergence where specifications for components are **interval bounds** using **Z3**.
- Future Work:
  - Provide bounds on lock time.
  - Complete models for low-pass filter and Delta-Sigma modulator.
  - Examine other digital PLL architectures to assess the reusability and automatability of this verification.
  - *Component validation*: formalize the connection between the models used and those used in other phases of the analog and mixed-signal design process.

Thank You!

References

- J. Crossley, E. Naviasky, and E. Alon, *An energy-efficient ring-oscillator digital pll*, Custom Integrated Circuits Conference (CICC), 2010 IEEE, 2010, pp. 1–4.
- Leonardo De Moura and Nikolaj Bjørner, *Z3: an efficient smt solver*, Proceedings of the Theory and practice of software, 14th international conference on Tools and algorithms for the construction and analysis of systems (Berlin, Heidelberg), TACAS'08/ETAPS'08, Springer-Verlag, 2008, pp. 337–340.
- Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler, *Spaceex: scalable verification of hybrid systems*, Proceedings of the 23rd international conference on Computer aided verification (Berlin, Heidelberg), CAV'11, Springer-Verlag, 2011, pp. 379–395.

Appendix: the Simplified Continuous Nonlinear Model

The ODE model:

$$\begin{aligned} \dot{c} &= g_1 \cdot (f_{ref} - \frac{1}{N}\frac{v_0+\alpha v}{c_0+\beta c}) \quad \textit{where} \quad c \in (c_{min}, c_{max}) \\ \dot{v} &= g_2 \cdot (c - c_{code}) \end{aligned}$$

Note: $\dot{c} = 0$ when $c = c_{\text{min}}$ or $c = c_{\text{max}}$

Quantization error is included in later work.

Linear phase path is simplified away for convenience.

Appendix: the Proposition

Formal definition of the proposition we introduced:

Proposition:

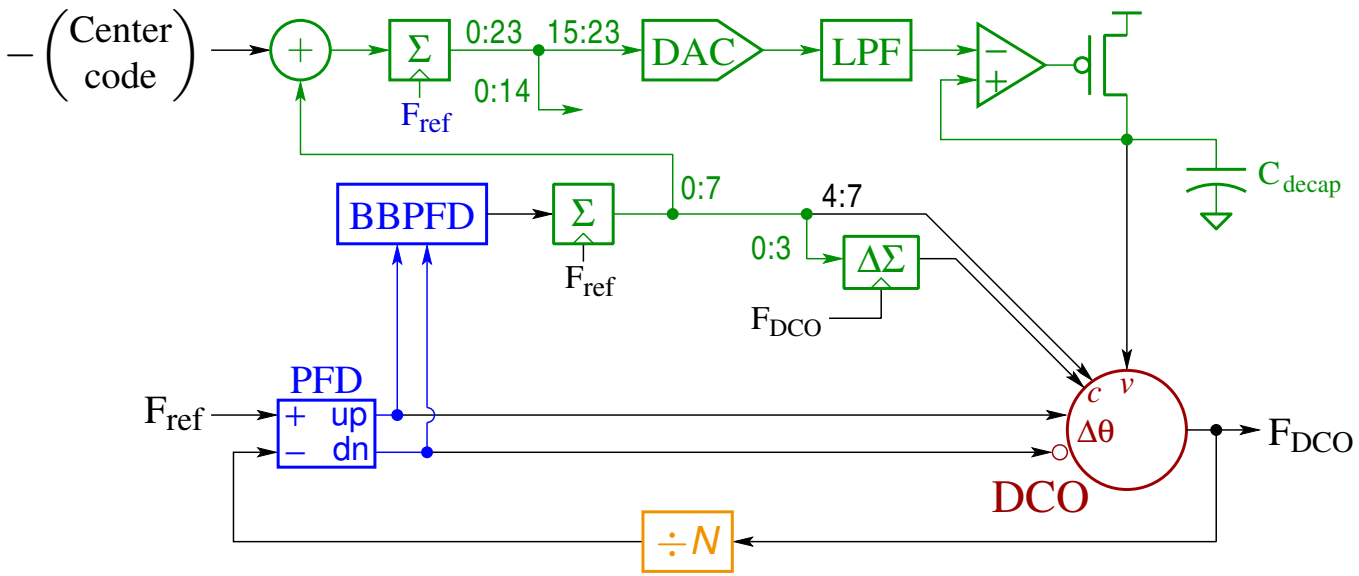
Let *Err* denote quantization error and assume *Err* is symmetric around 0: if $\eta \in Err$ then $-\eta \in Err$ as well. We can easily show:

$$\forall x \in Q_0 - Q_T . \forall \eta \in Err . h(x+\eta)^T P x < 0 \Leftrightarrow$$
$$\forall y \in (Q_0 - Q_T) \oplus Err . \forall \eta \in Err . (y + \eta \in Q_0 - Q_T) \Rightarrow (h(y)^T P (y+\eta) < 0)$$

The Minkowski sum of two sets, $A \oplus B$, is the set of elements that can be obtained as the sum of an element from A and an element from B:

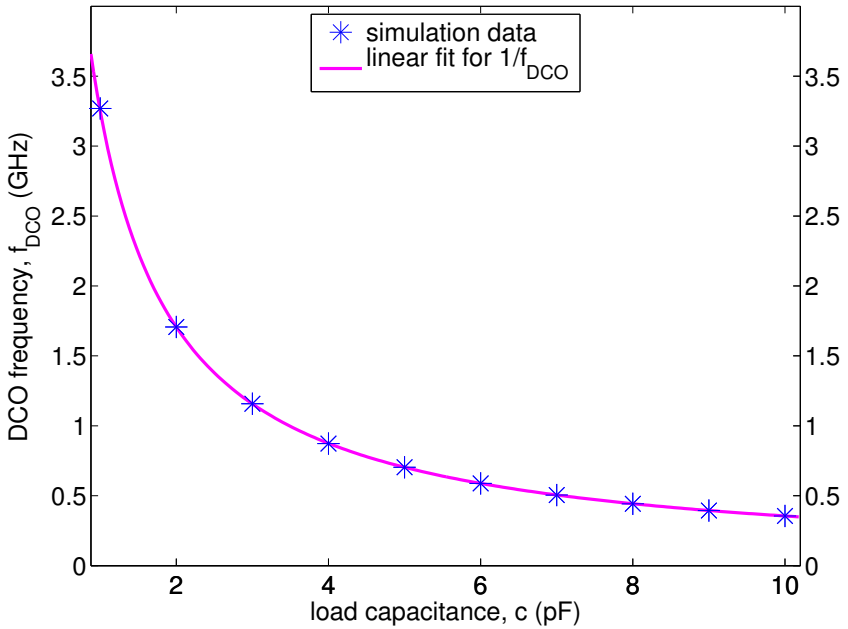
$$A \oplus B \quad = \quad \{z \mid \exists a \in A.\ \exists b \in B.\ z = a + b\}$$

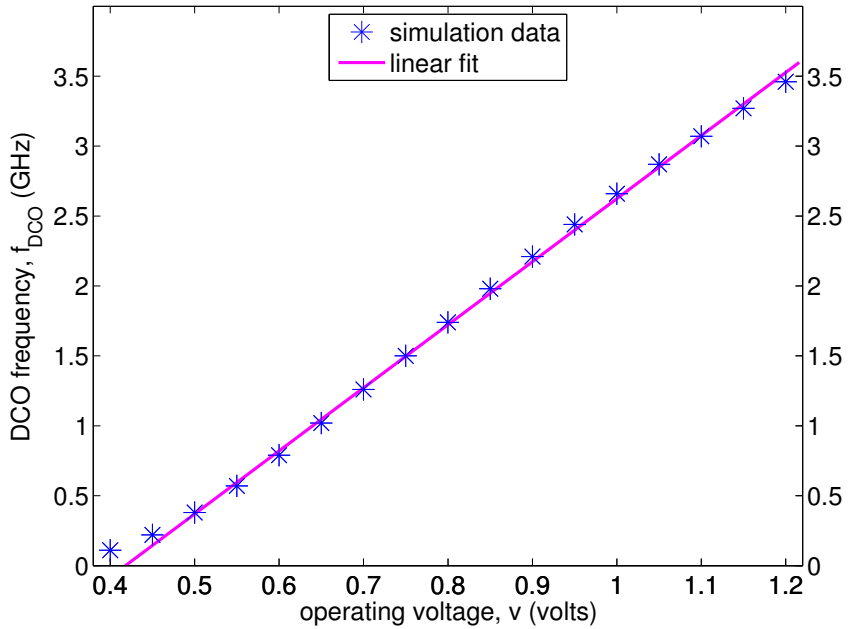
PLL: Verified digital PLL



- We omitted the Delta-Sigma modulator, low-pass filter, and linear regulator for simplicity.
- We believe all of these could be included using the same methods that we've used for the rest of the digital PLL.

Modeling the DCO

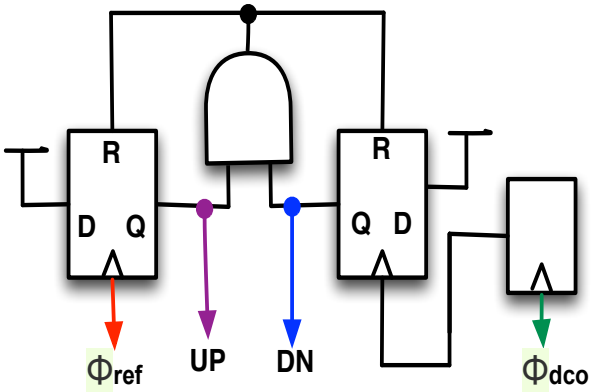


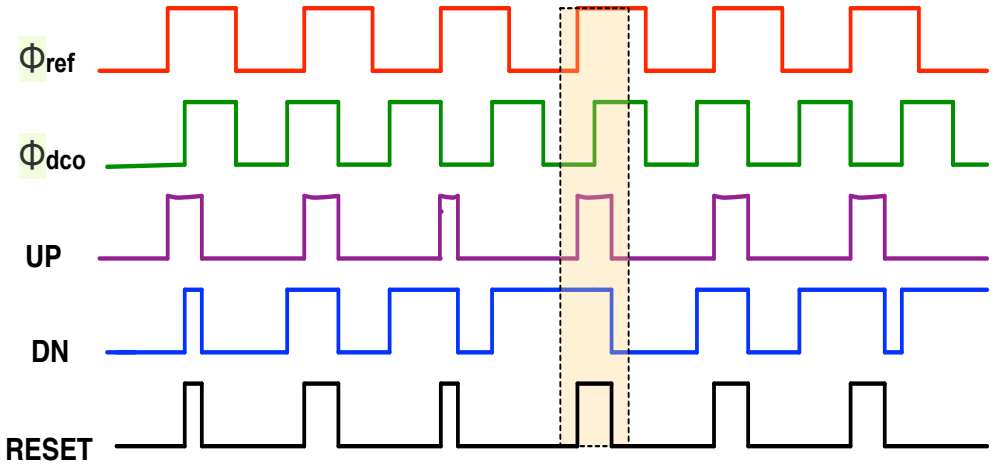


- Simulation results (Spectre, 65nm, simple ring-oscillator) show DCO period is nearly linear in C.
- DCO frequency is linear in v_{ctrl}.
- **Our model:**

$$f_{DCO} \in \frac{1 + \alpha v_{ctrl}}{1 + \beta C} f_0 \oplus err_{DCO}$$

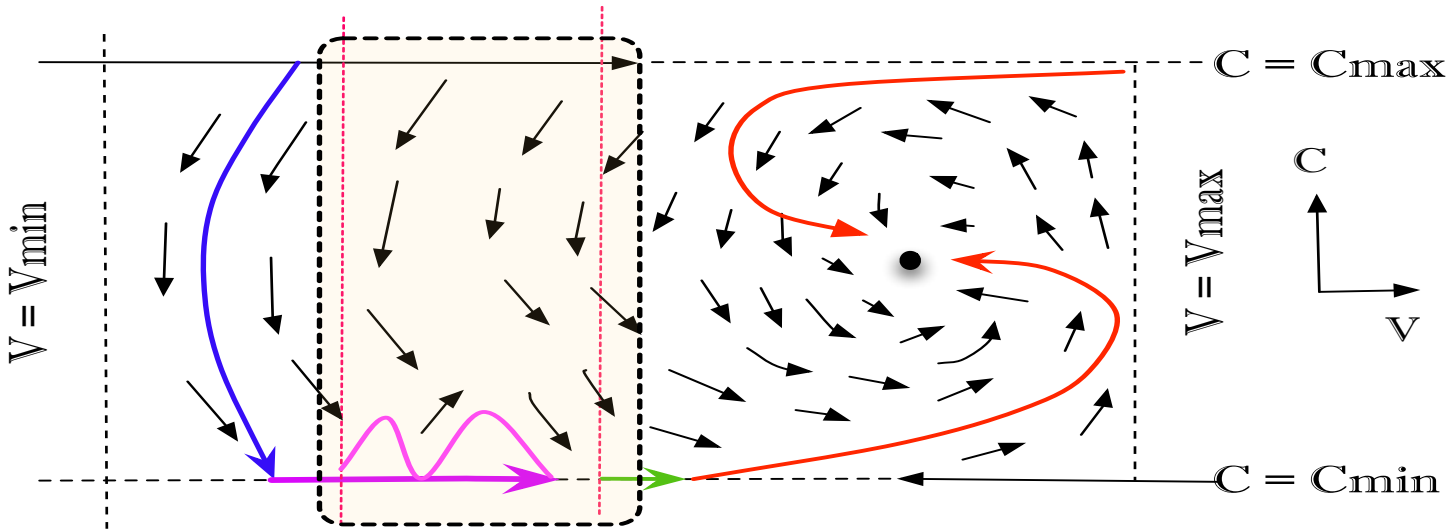
Reset Delay





- Because of reset delays, edges of Φ_{ref} or Φ_{dco} maybe missed.
  - This can cause occasional flips in the sign of the phase-detector output.
- Analytically, we derived a v_{ctrl} bound, $[V_{lo}, V_{hi}]$ that precludes such sign flips.
- Our bound shows that such sign flips cannot occur for the design parameters of the CICC'10 PLL.

Reset Delay in Phase two



- In Phase two, the *c* signal could move from its saturation value because of the spurious sign changes at the output of the PFD.
- We simulated the PFD in Spectre to measure the reset delay.
- SpaceEx verified that *V* will still make process even with these spurious sign changes.

Appendix: Comparison

- The hybrid-automaton approach very closely followed the structure of the digital PLL.
  - Seems likely to be more intuitive for real-world designers and verifiers.
  - Verified the digital PLL for a specific choice of model parameters.
  - ”Feels” like model-checking.
  - Manual decomposition into lemmas required.
- The SMT approach is more general:
  - Handles some of the non-linearities directly.
  - Verified the digital PLL with model parameters in interval ranges.
  - ”Feels” like theorem proving.