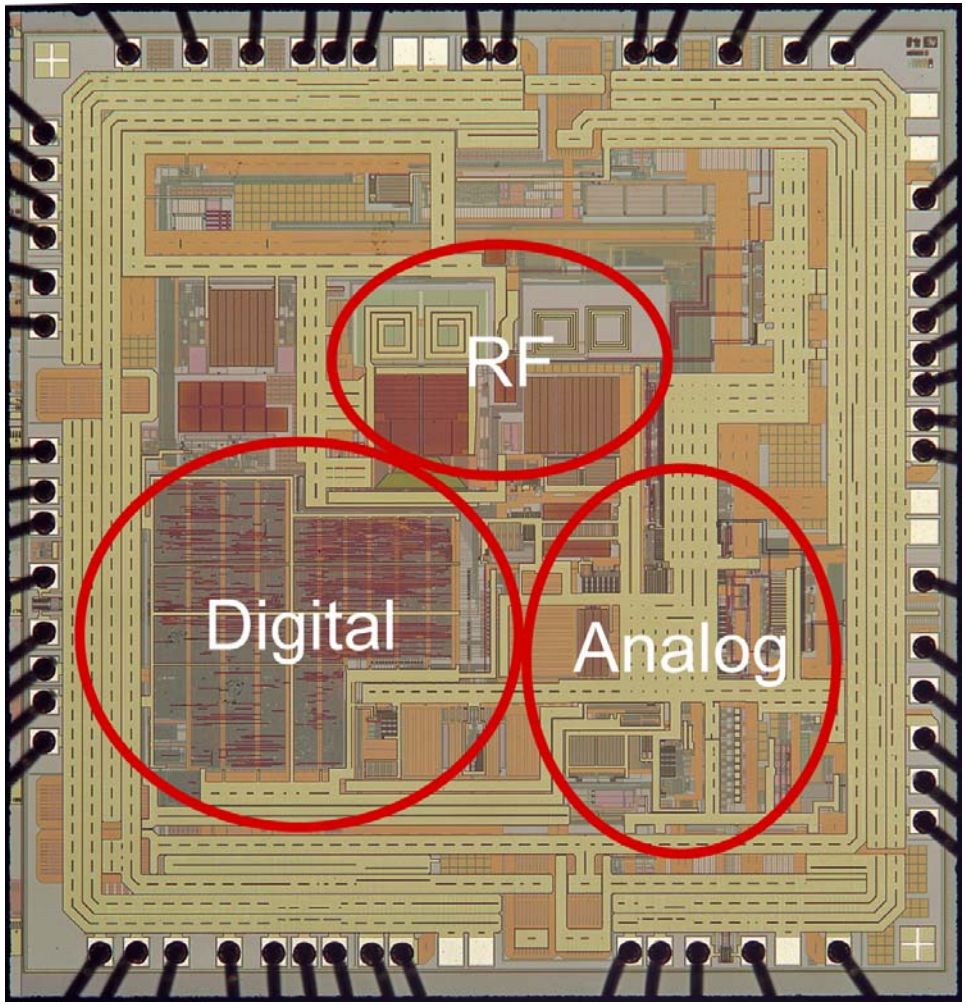
A Top Down Design Methodology for Mixed-Signal Integrated Circuits using the VppSim Simulator

CDNLive! 2006
Session Track: Custom IC Design
Session Number: 3.4

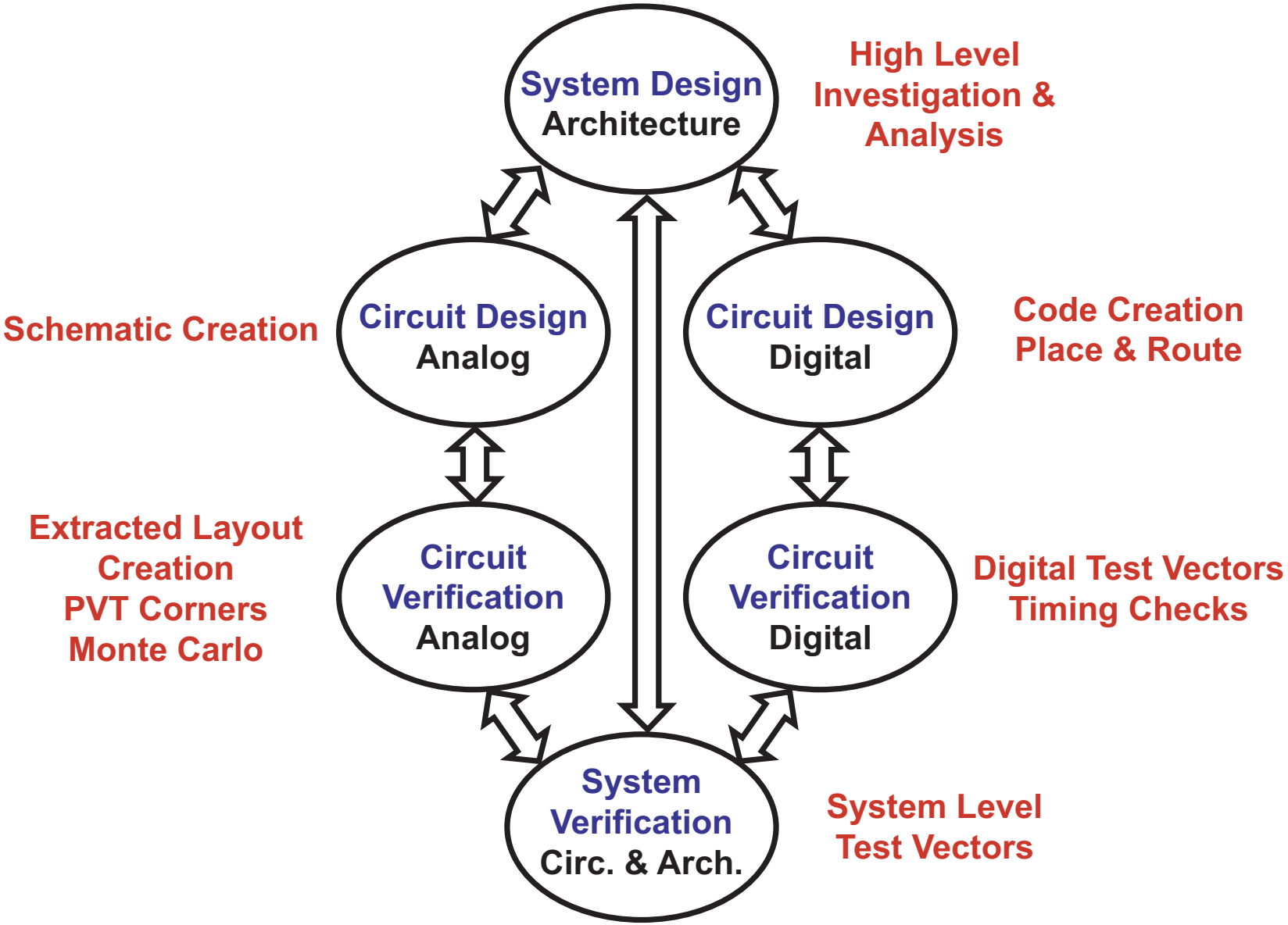
Michael Perrott
Massachusetts Institute of Technology
September 2006

A Modern "Analog" Custom IC

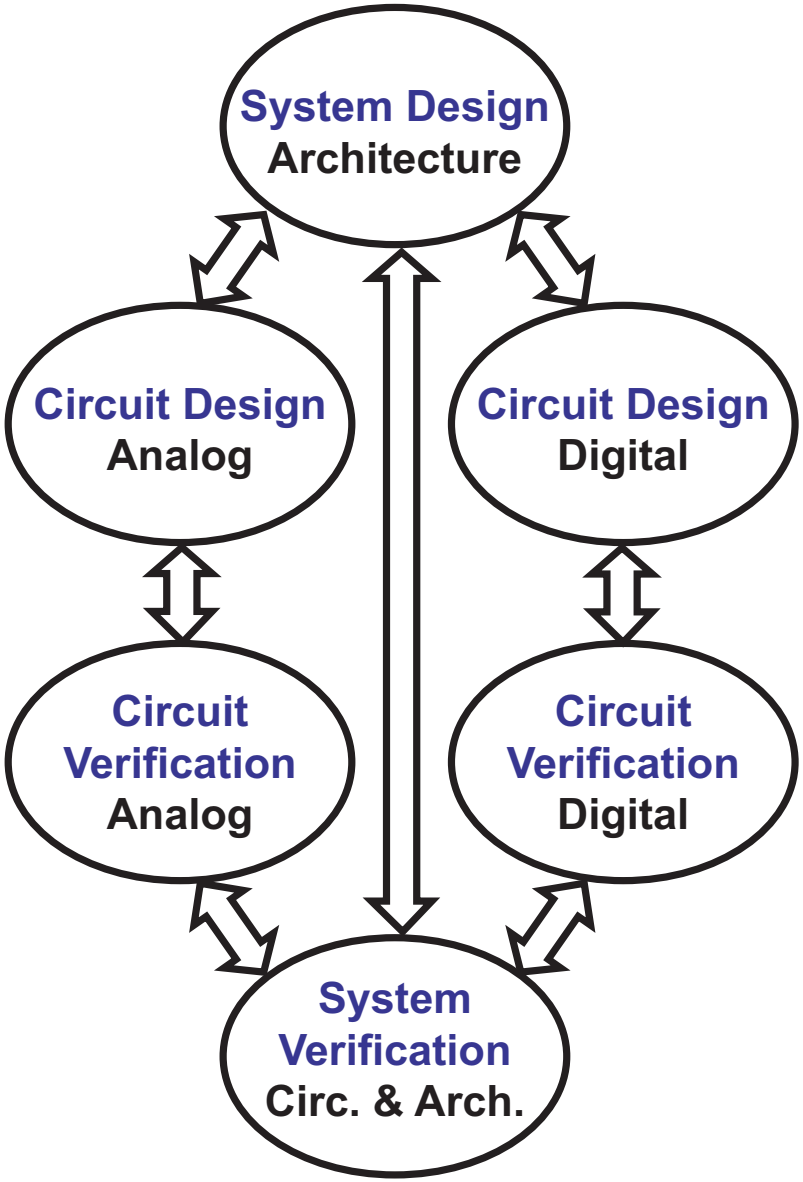


- **A 2.5 Gb/s CDR for high speed links**
  - ***Analog*** **amplification and phase sensing**
  - ***Digital*** **filtering and calibration**
  - ***RF*** **clock generation (2.5 GHz)**
- **How do we design such chips?**

Simplified View of a Top Down, Mixed-Signal Design Flow



The Many Simulation Tools Involved …



Popular Tools

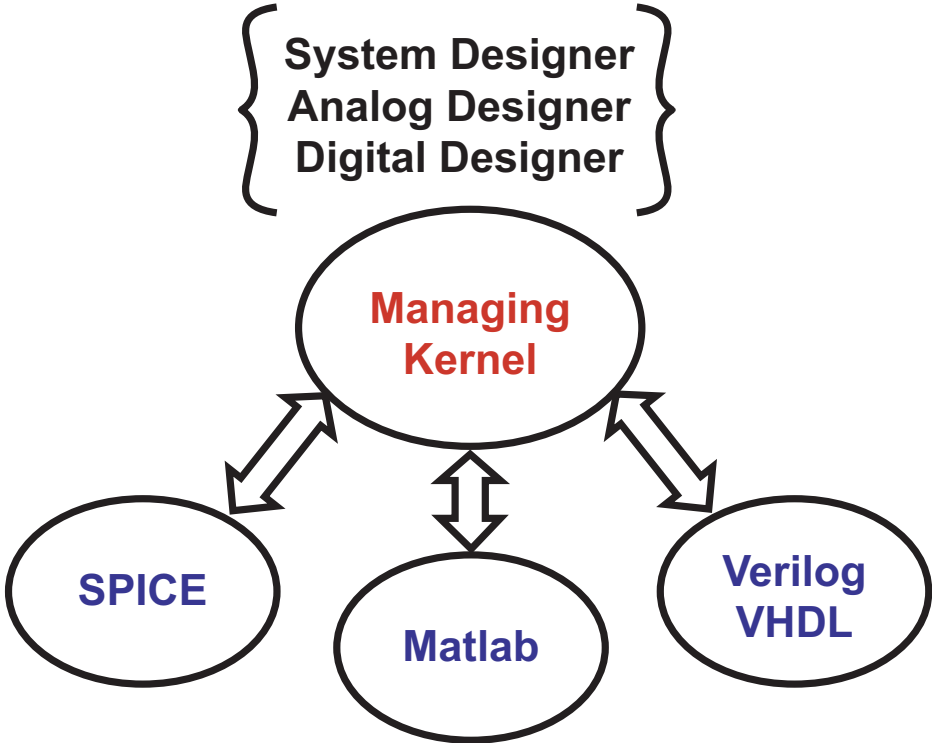
Matlab/Simulink
ADS

SPICE **Verilog/VHDL**

Verilog AMS
AMS Designer

????????

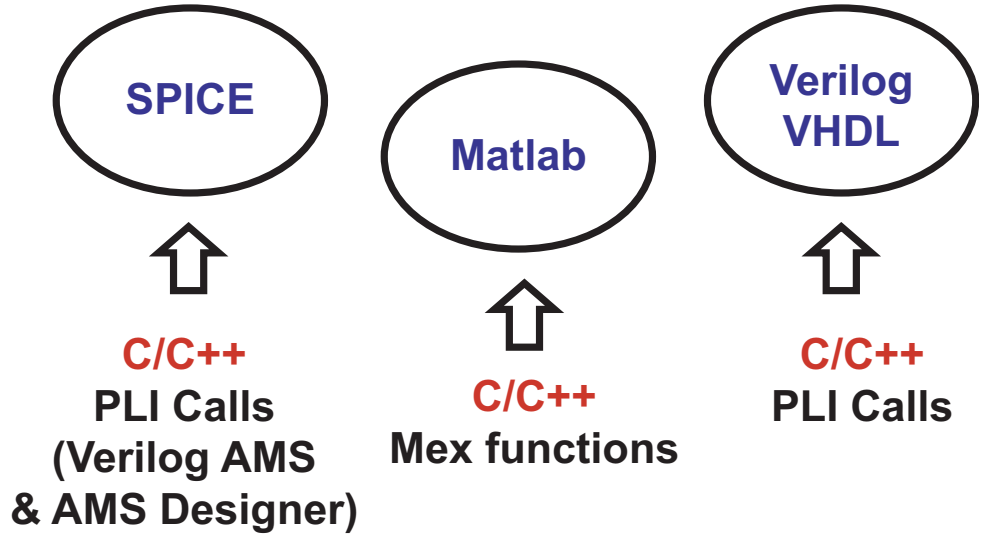
Goal: Create a Universal Simulator



- The challenges of developing the Managing Kernel
  - Difficult to match up simulator signals at their boundaries without overly complicating the life of the user
  - Difficult to maintain fast simulation speed
  - Difficult to retrain designers on a new tool with a new flow

A Different Approach

- Look for commonality among simulators to allow "universal" simulation models to be used
  - C++ provides one such hook
- Allow designers to use their tool of choice while sharing "universal" simulation models



Key idea: bootstrap into existing simulators

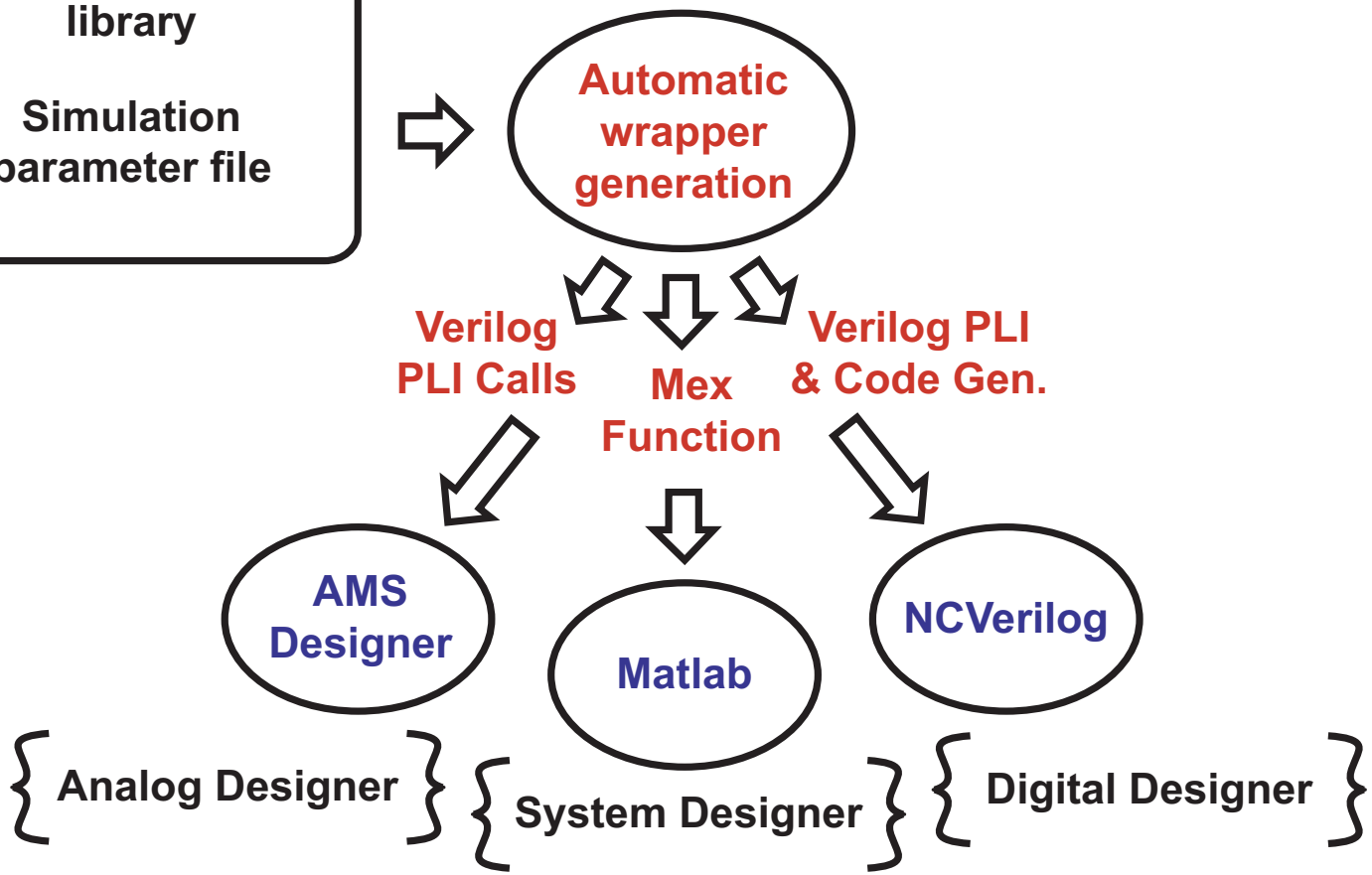
The VppSim Simulator Greatly Simplifies This Process

Cadence Schematic Entry

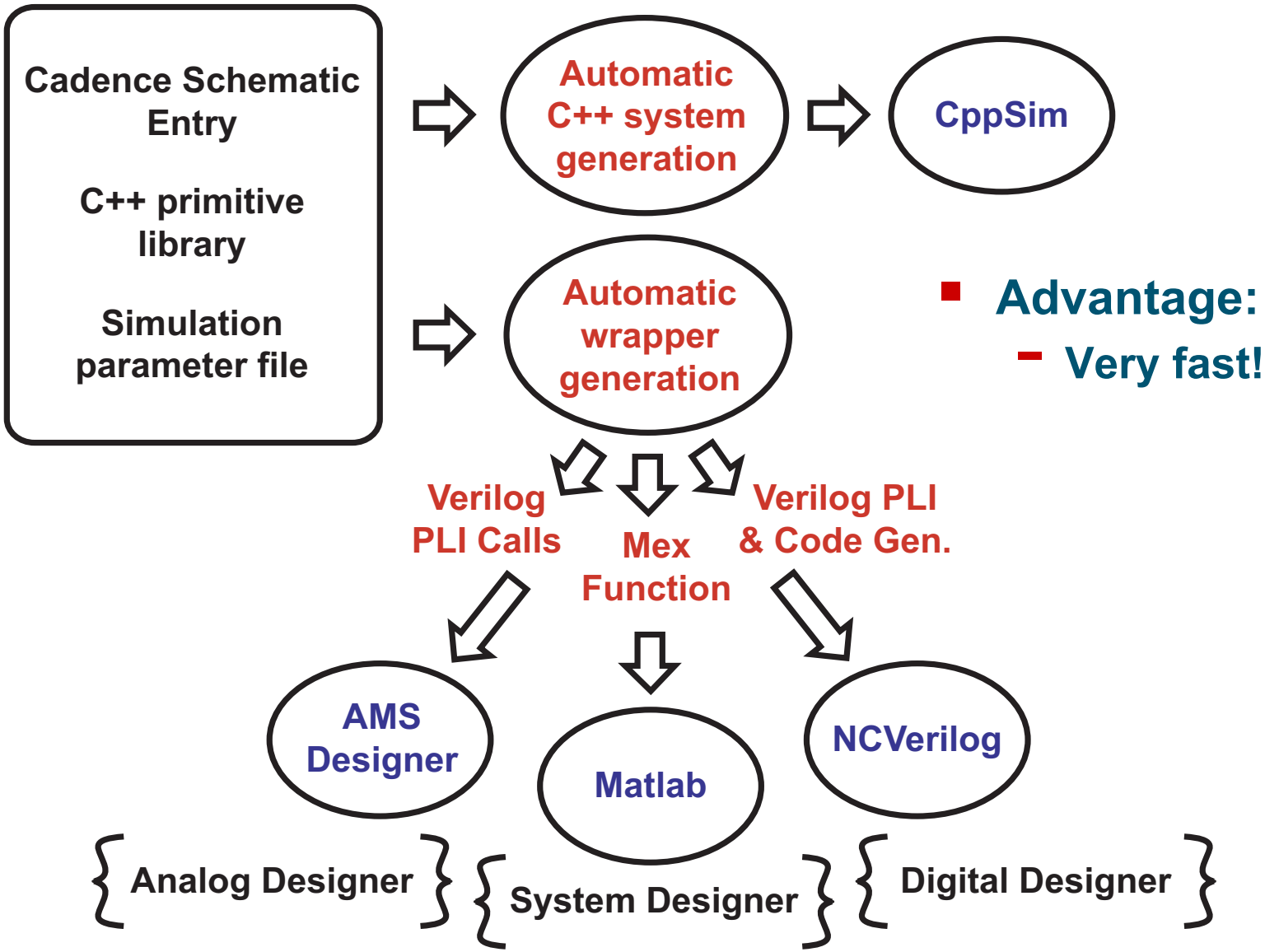
C++ primitive library

Simulation parameter file

- **Graphically driven generation of simulator blocks based on a C++ primitive library**

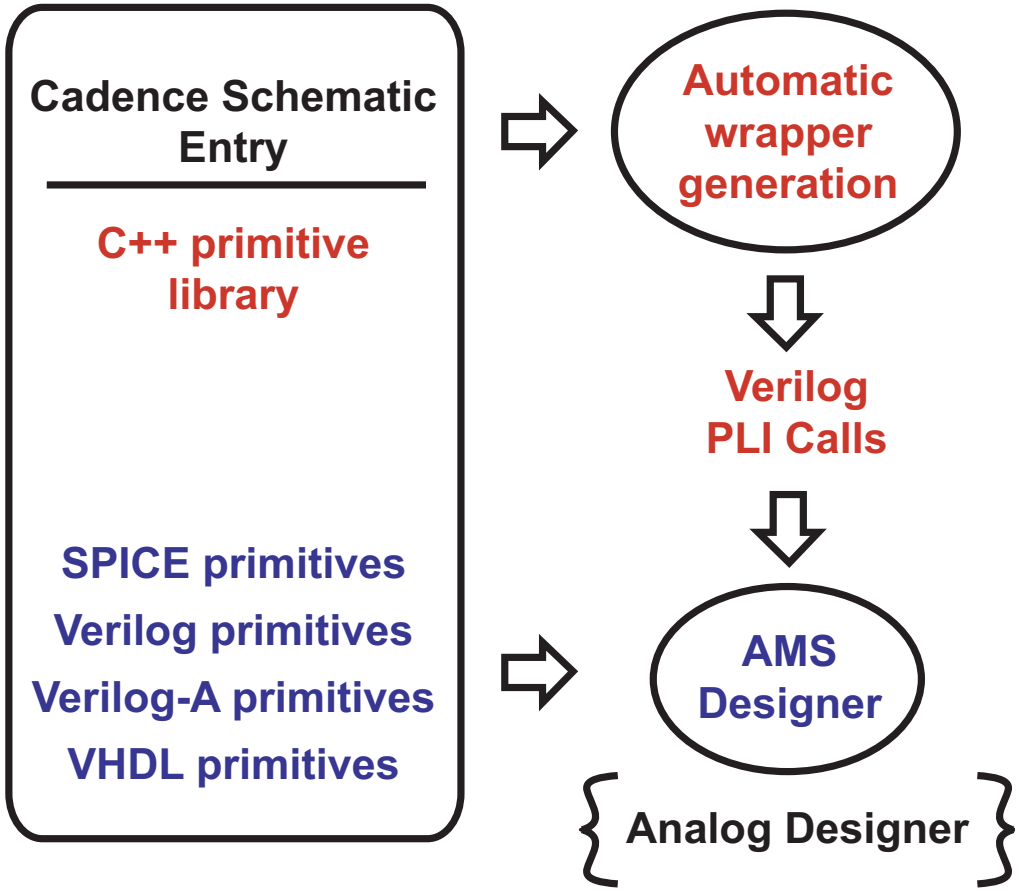


C++ Code Can Also Be Directly Simulated



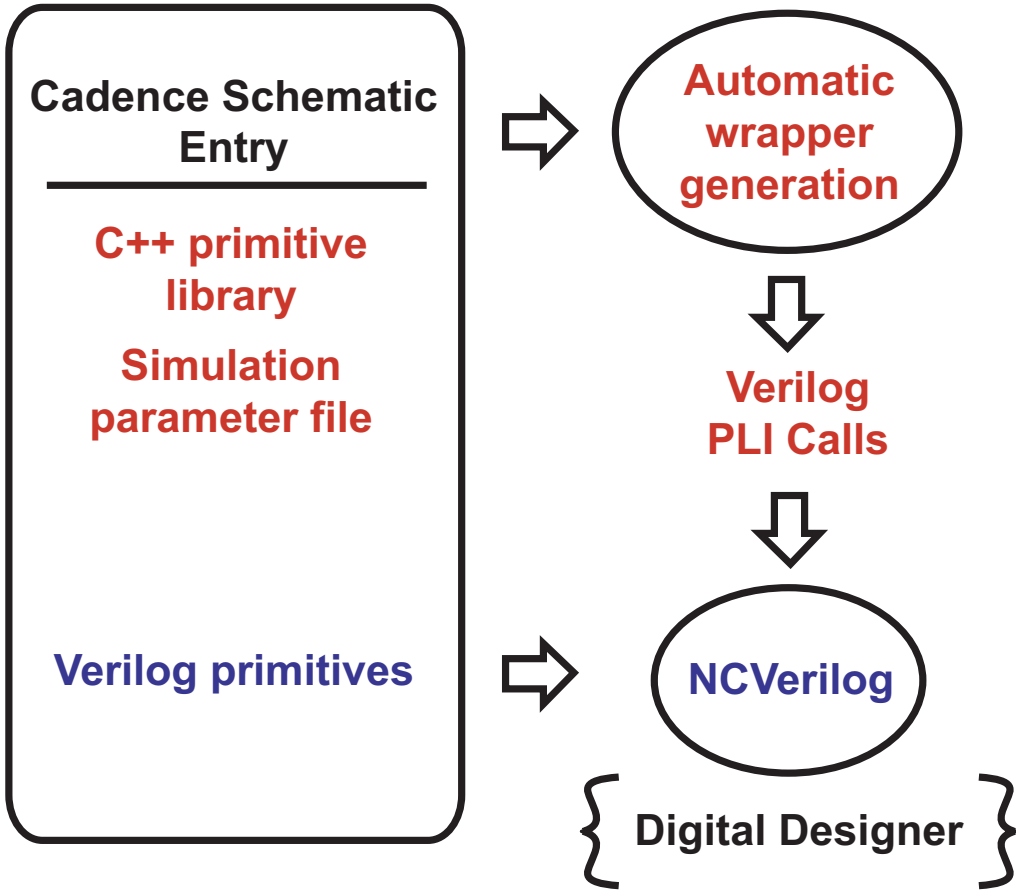
- **Advantage:**
  - **Very fast!**

Using VppSim Within an Analog Design Flow



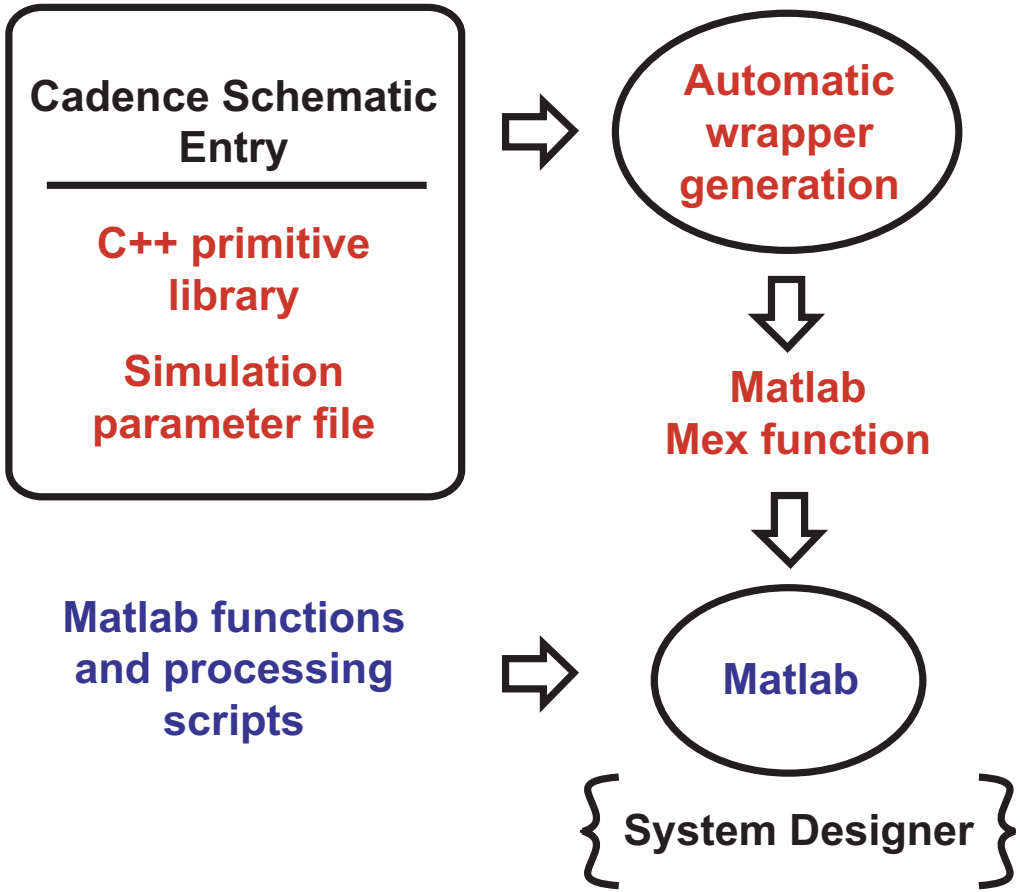
- C++ behavioral models allow accurate representation of:
  - Analog blocks, algorithms, sophisticated signal sources
- Advantages of C++: fast, portable, and easy to template

Using VppSim Within a Digital Design Flow



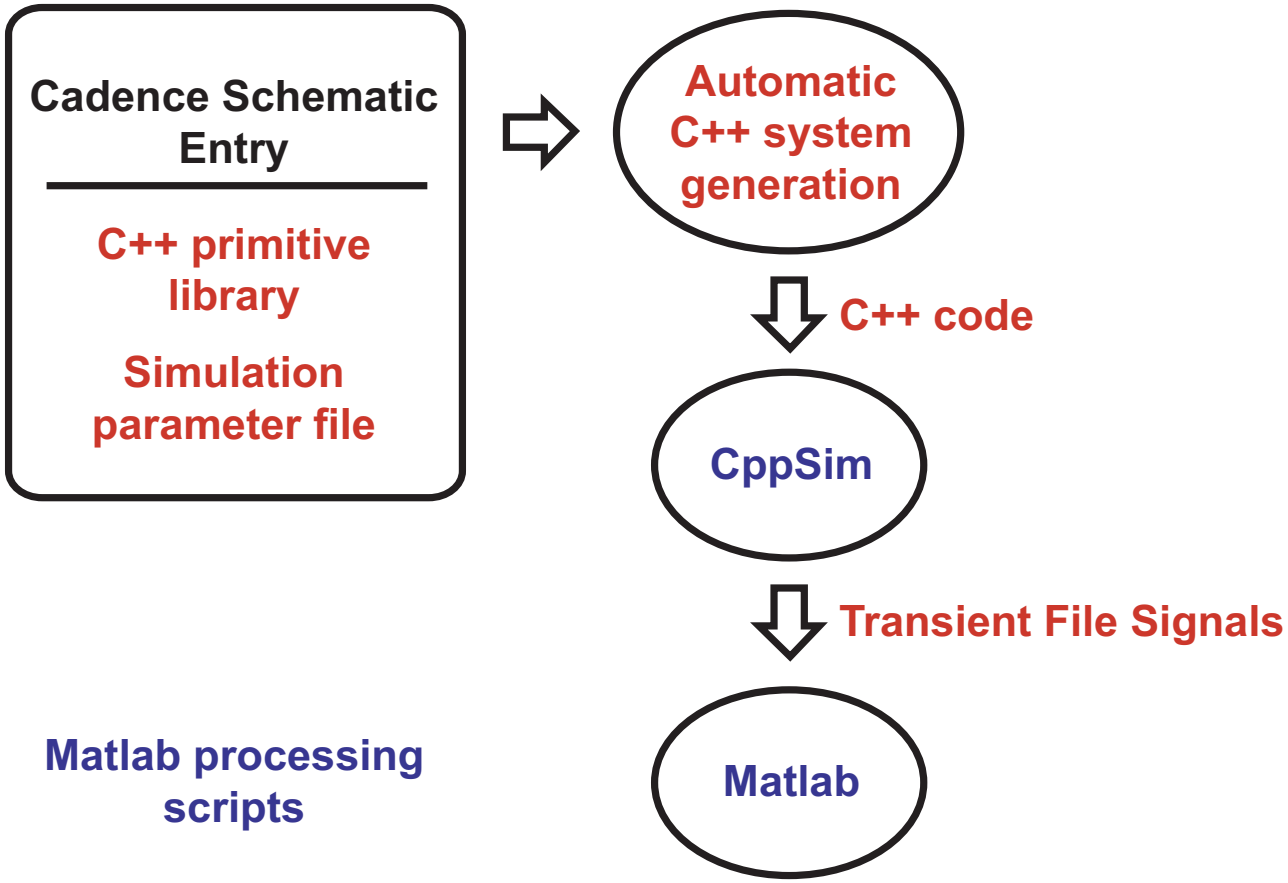
- Allows digital designer to easily leverage models built by analog designer
- Digital design flow remains unchanged otherwise!

Using VppSim within a System Design Flow



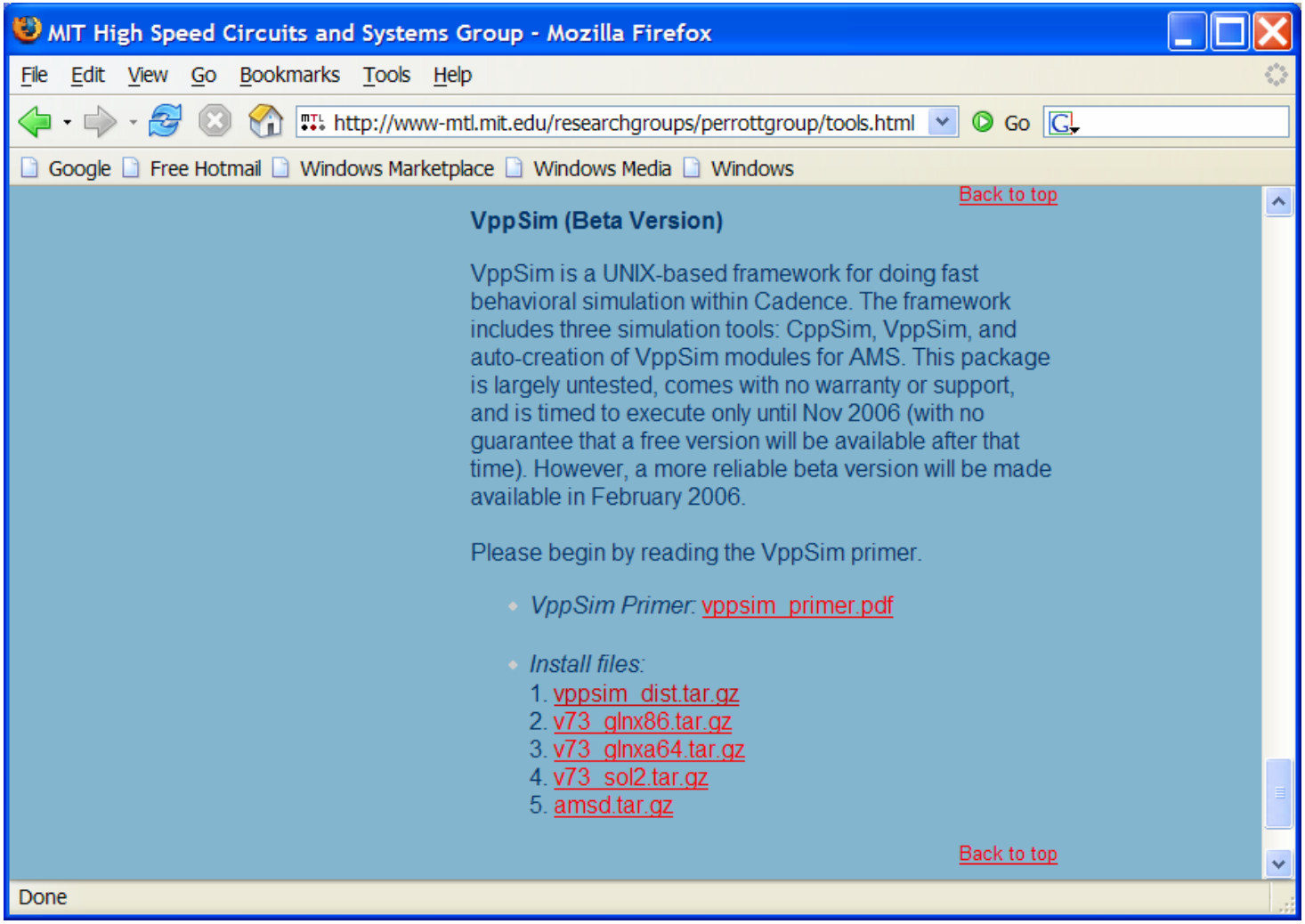
- Allows system designer to directly leverage analog and digital C++ behavioral models developed by the analog and digital designers
- System design flow is otherwise unchanged!

An Alternative System Flow



- CppSim can be used to achieve very fast C++ behavioral simulation more directly within a graphical framework
- Matlab can be used for sophisticated post-processing

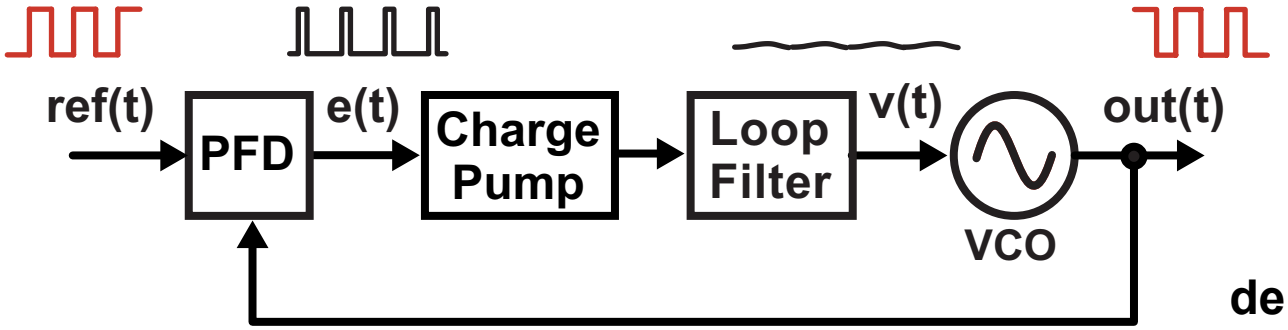
Downloading VppSim …



http://www-mtl.mit.edu/researchgroups/perrottgroup/tools.html

Can You Accurately Model Complicated Mixed-Signal Circuits using C++ Behavioral Models?

Focus Area: Phase-Locked Loops

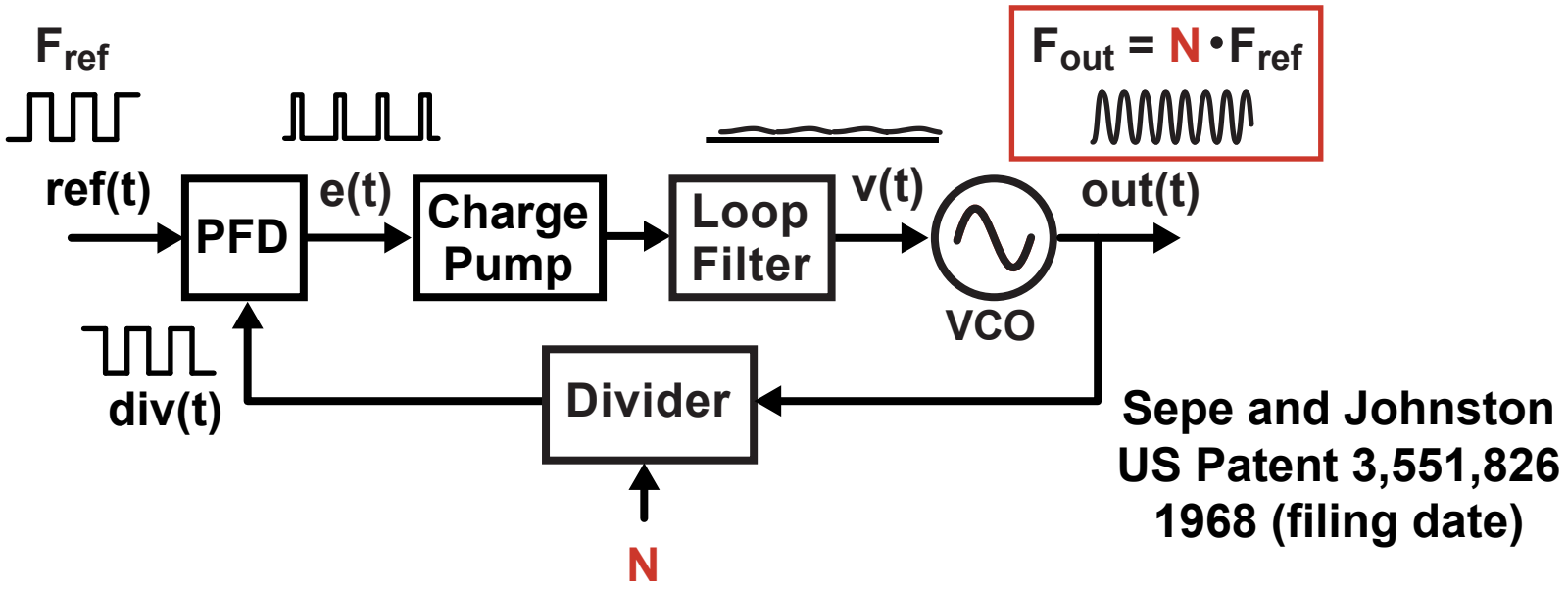


de Bellescize
Onde Electr, Vol. 11
1932

- VCO ⇨ produces variable frequency output
- Reference ⇨ provides input frequency/phase
- PFD ⇨ compares phase of ref and VCO output
- Charge pump ⇨ simplifies loop filter implementation
- Loop filter ⇨ smooths PFD signal

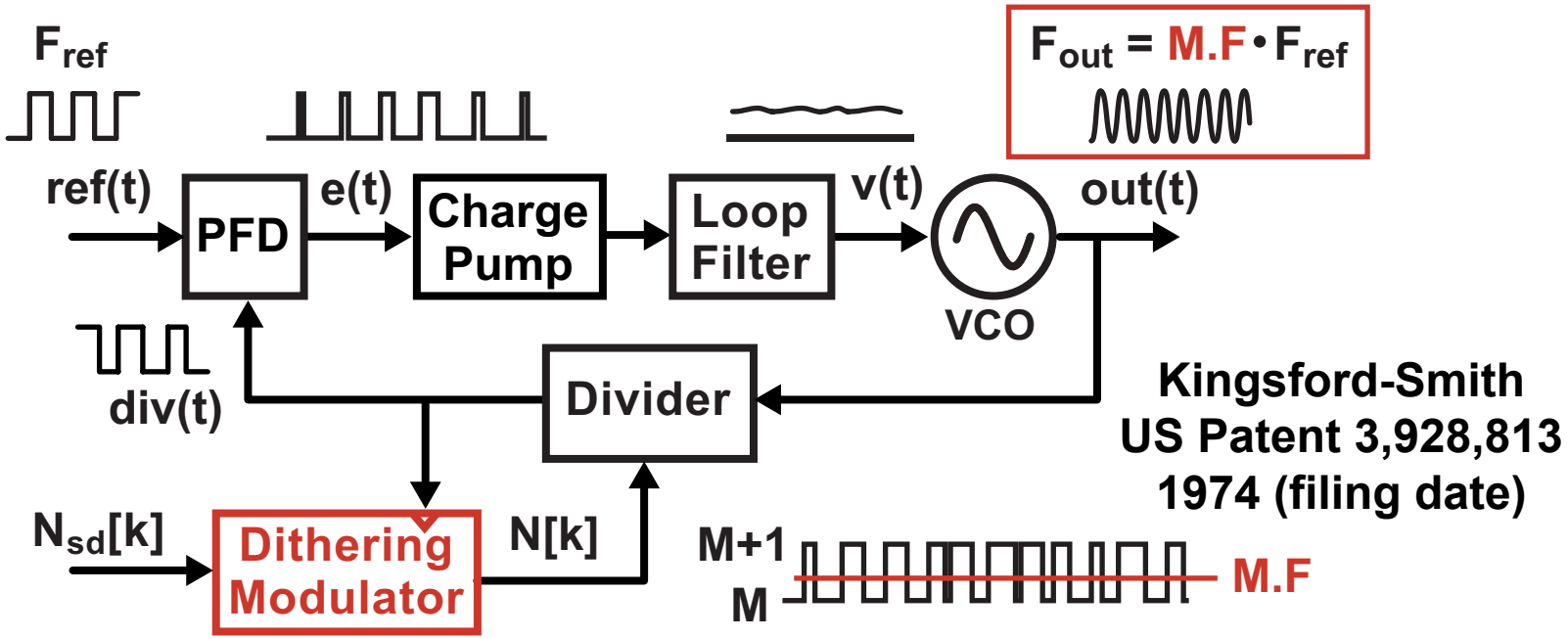
Objective: "Lock" VCO phase to reference phase

Integer-N Frequency Synthesizer



- Leverages frequency divider to create "indirect" frequency multiplication
  - Allows digital adjustment of output frequency in increments of the reference frequency

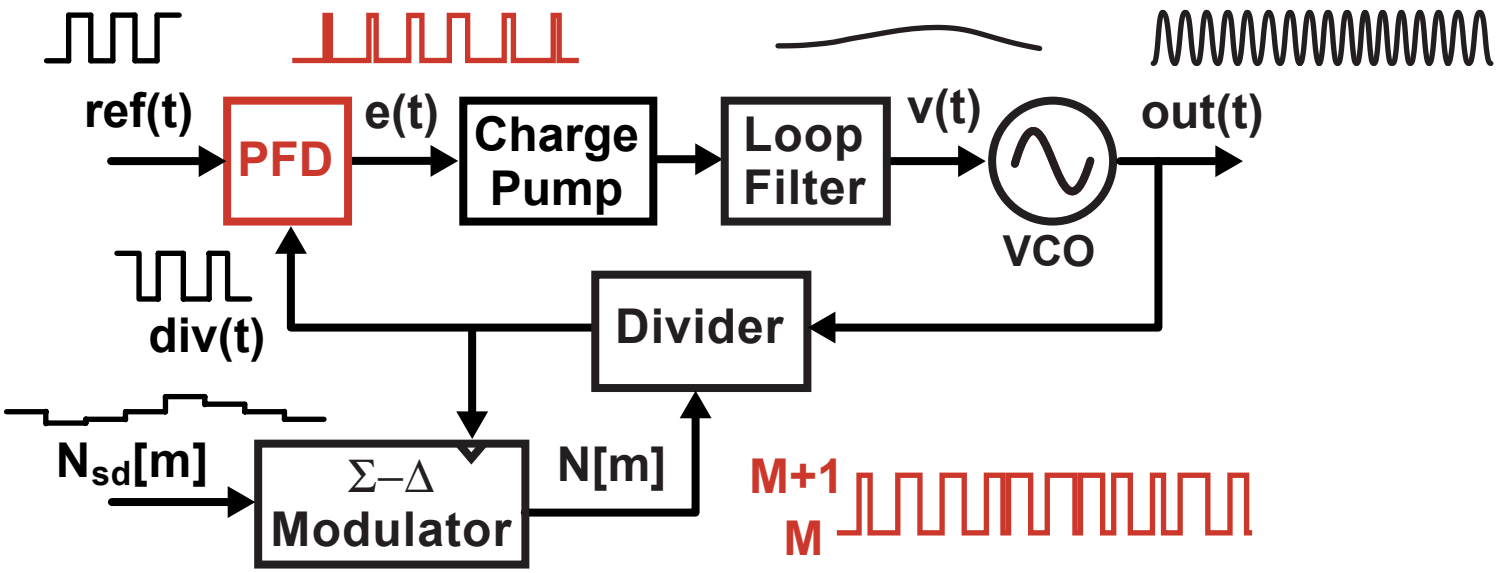
Fractional-N Frequency Synthesis



- Divide value is dithered between integer values
  - Fractional divide values are realized
  - Apply Sigma Delta concepts to dithering

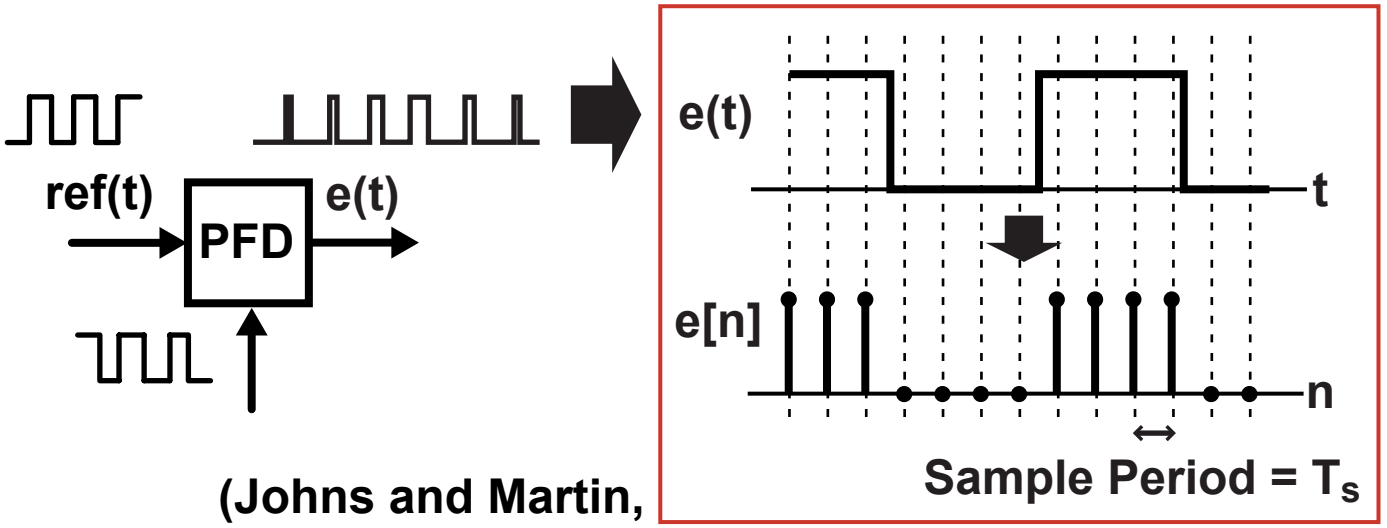
Very high frequency resolution!

PLL Simulation Issues (Behavioral Level)



- Slow simulation time
  - High output frequency ⇨ High sample rate
  - Long time constants ⇨ Long time span for transients
- Inaccurate results
  - PFD output information conveyed in pulses

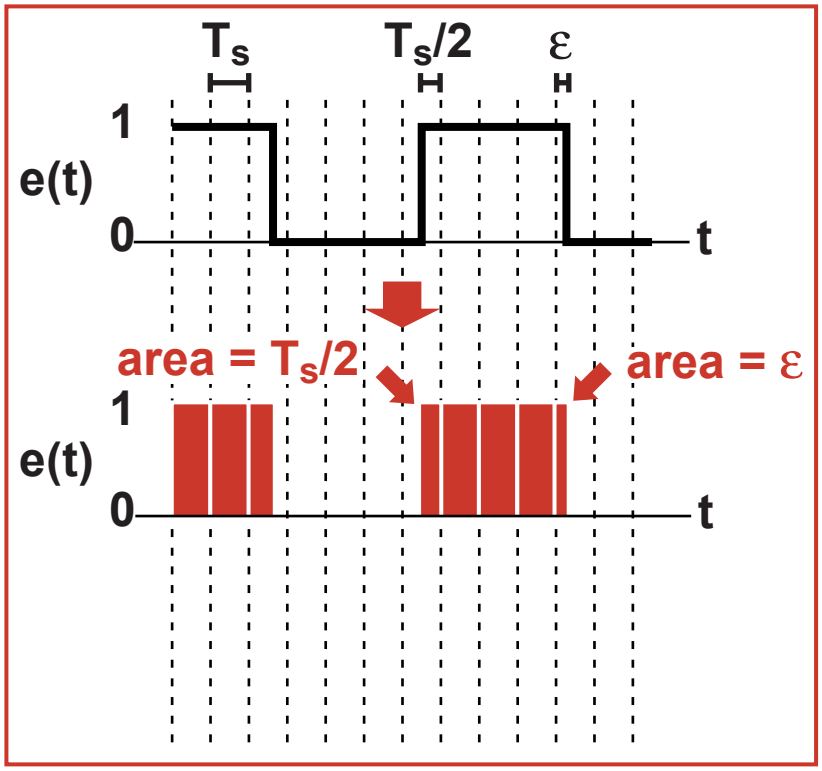
Classical Constant-Time Step Method

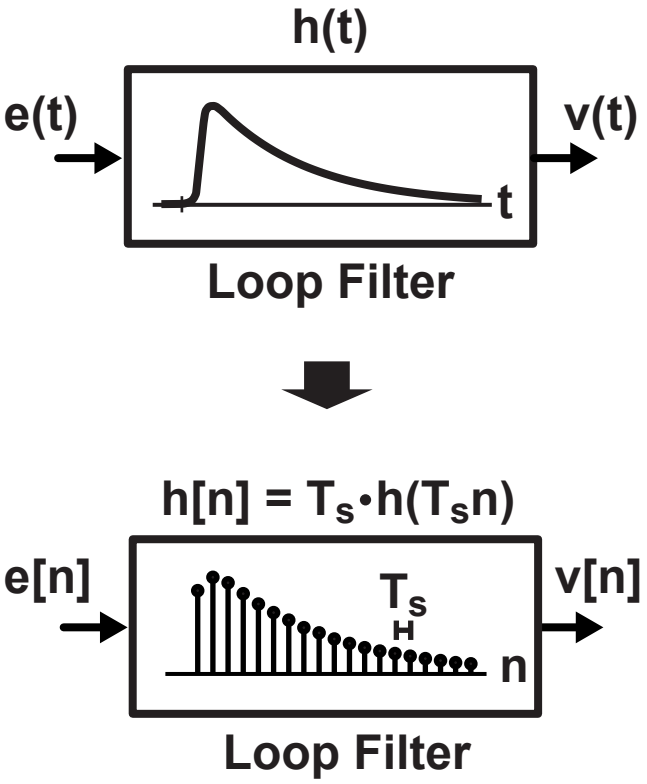


(Johns and Martin, Analog Integrated Circuit Design)

- Directly sample the PFD output according to the simulation sample period
  - Simple, fast, readily implemented in Matlab, Verilog, C++
- Issue – quantization noise is introduced
  - This noise overwhelms the PLL noise sources we are trying to simulate

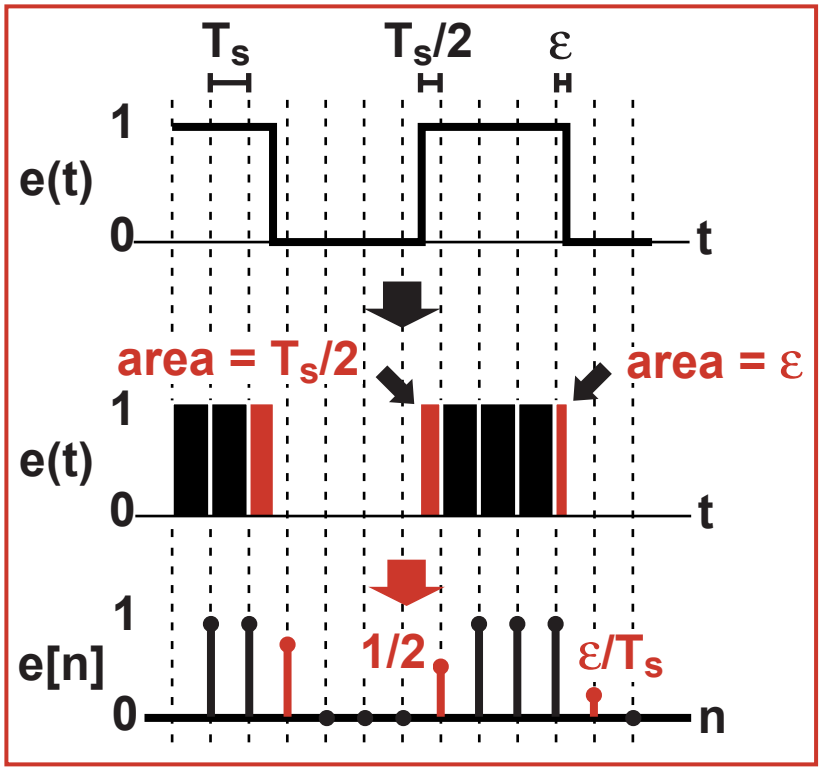
Proposed Approach: View as Series of Pulses

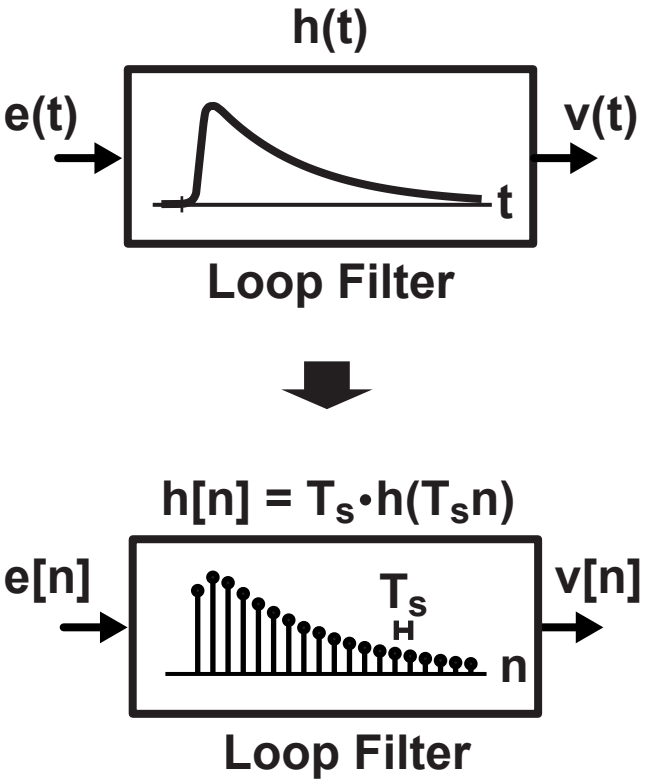




- Area of each pulse set by edge locations
- Key observations:
  - Pulses look like impulses to loop filter
  - Impulses are parameterized by their area and time offset

Proposed Method





Perrott, DAC, **June 2002**

- **Set e[n] samples according to pulse areas**
  - **Leads to very accurate results and fast computation**
- **Implemented in the CppSim Behavioral Simulator**

Does C++ behavioral simulation really help?

Some Fabricated Examples to Consider:

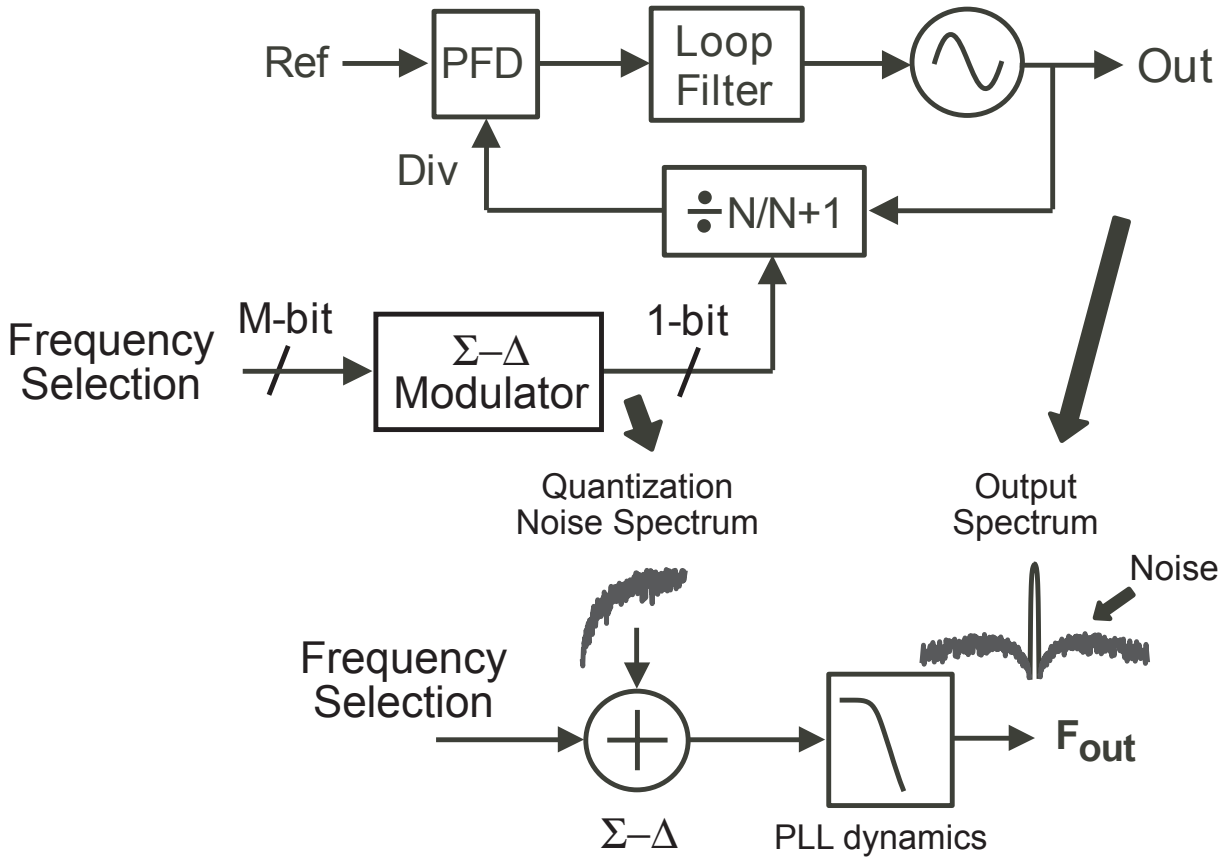
1) A 3.6 GHz, Low Noise, Wideband Fractional-N Synthesizer

2) A 2.5 Gb/s, Low Jitter, Clock and Data Recovery Circuit

3) A 3.1 GHz Limit Amplifier with Fast Offset Correction

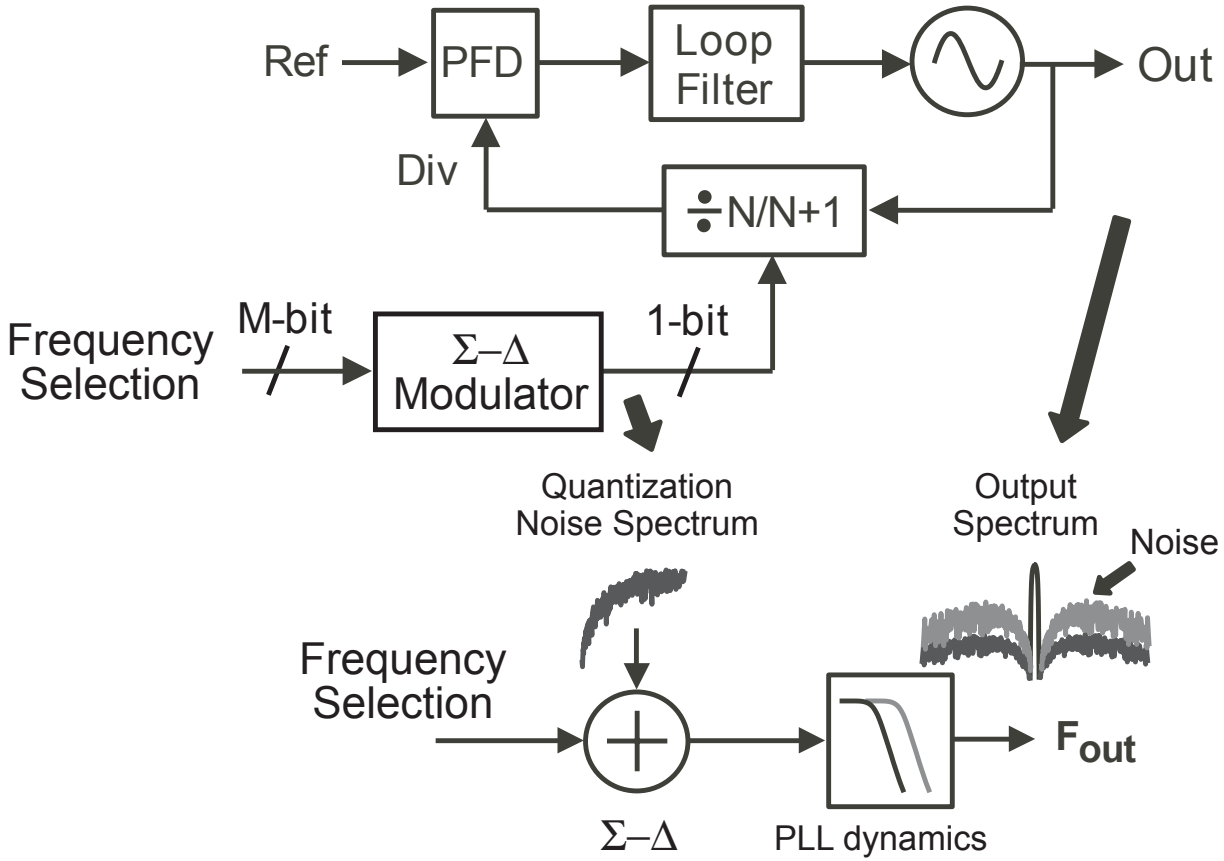
4) A 2 GHz VCO-based A/D Converter

The Issue of Dithering Noise



- Dithering of divide value introduces quantization noise
- PLL lowpass dynamics suppress this noise

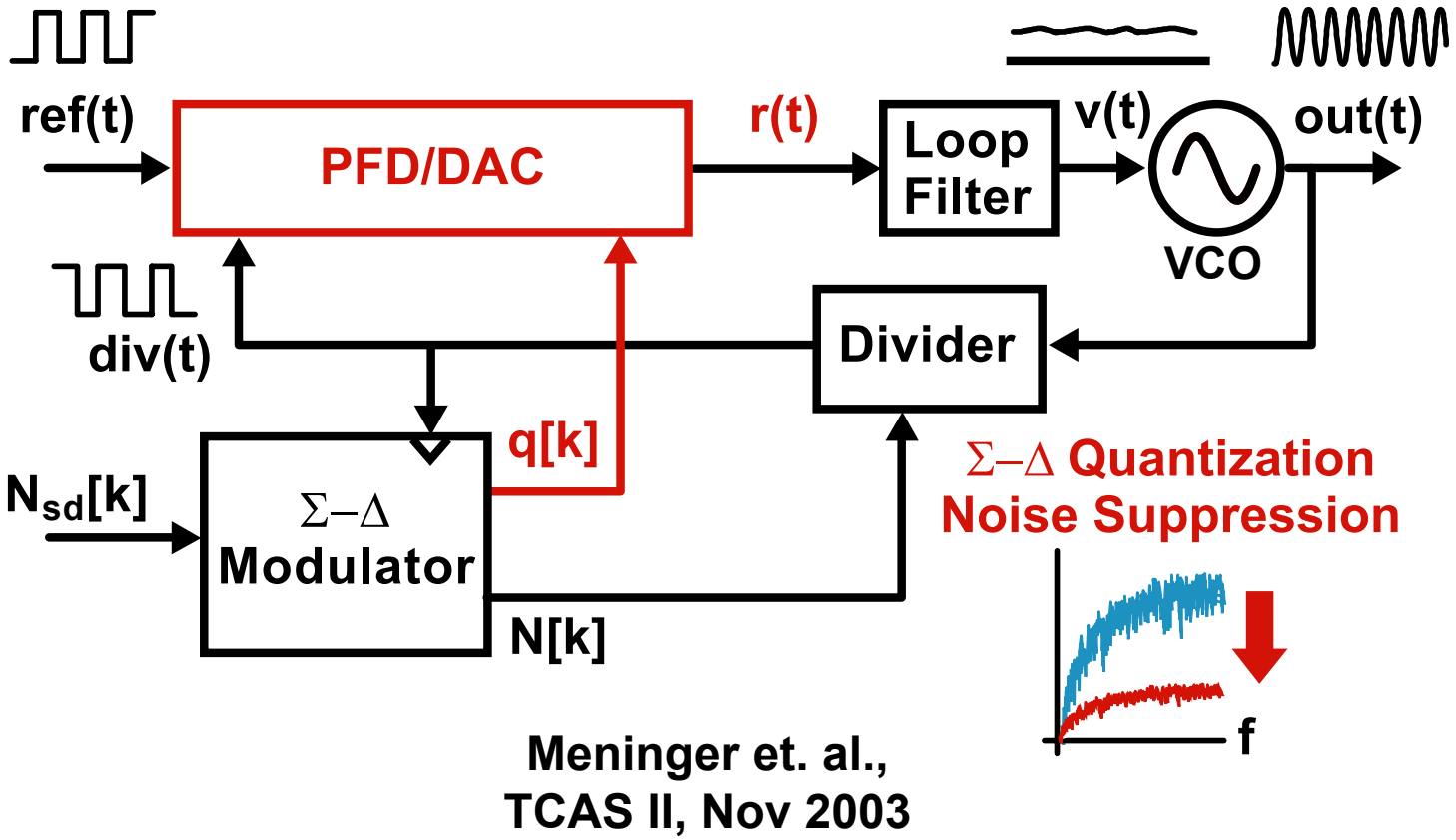
Impact of Increasing the PLL Bandwidth



- **Higher PLL bandwidth leads to higher noise**

Tradeoff: Noise performance vs PLL bandwidth

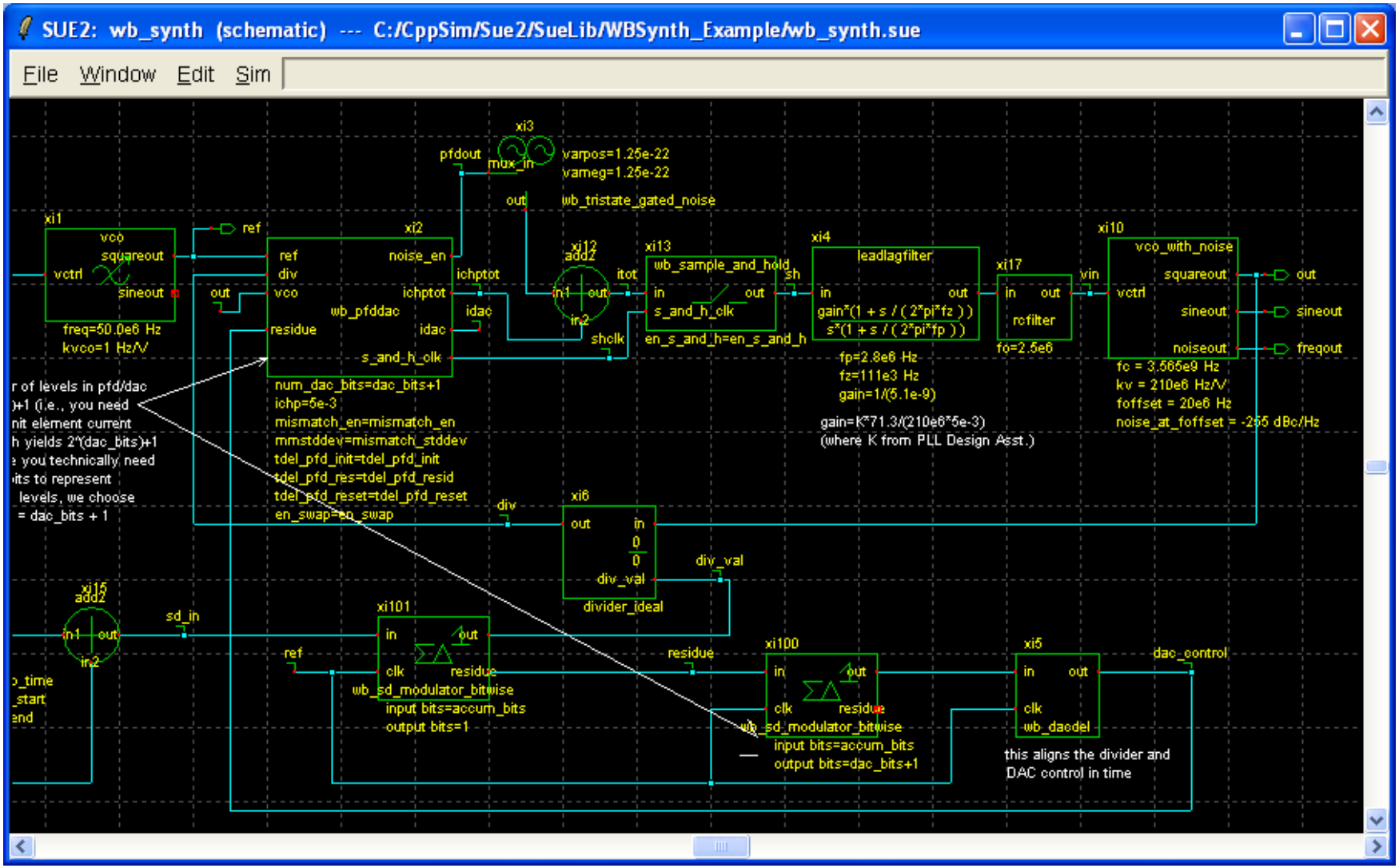
Goal: Use Noise Cancellation to Improve Performance



Meninger et. al., TCAS II, Nov 2003

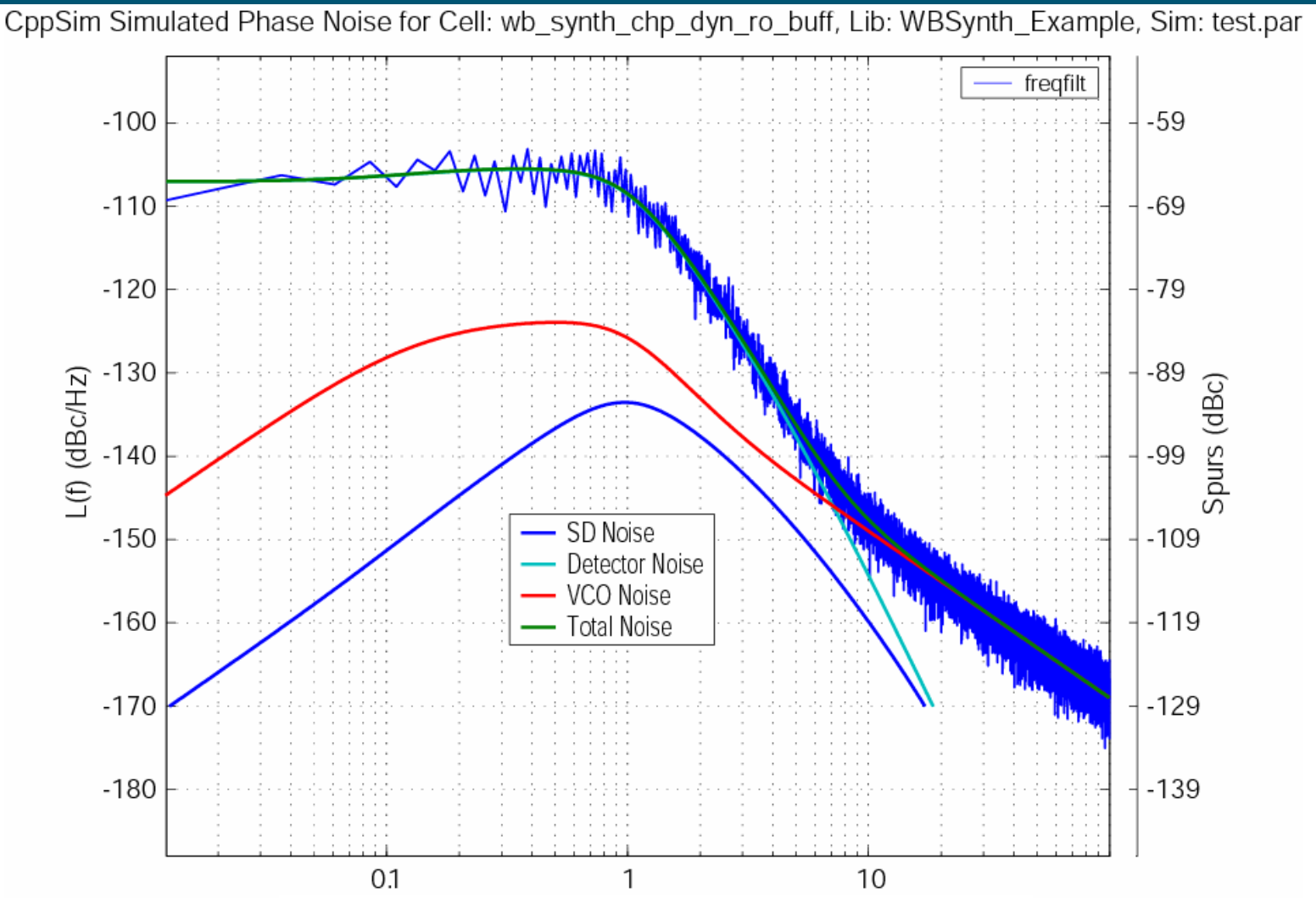
- Key challenge: understanding impact of various circuit mismatches
  - Sensitive to both timing and DAC current mismatch

Simulation of PFD/DAC Synthesizer using CppSim



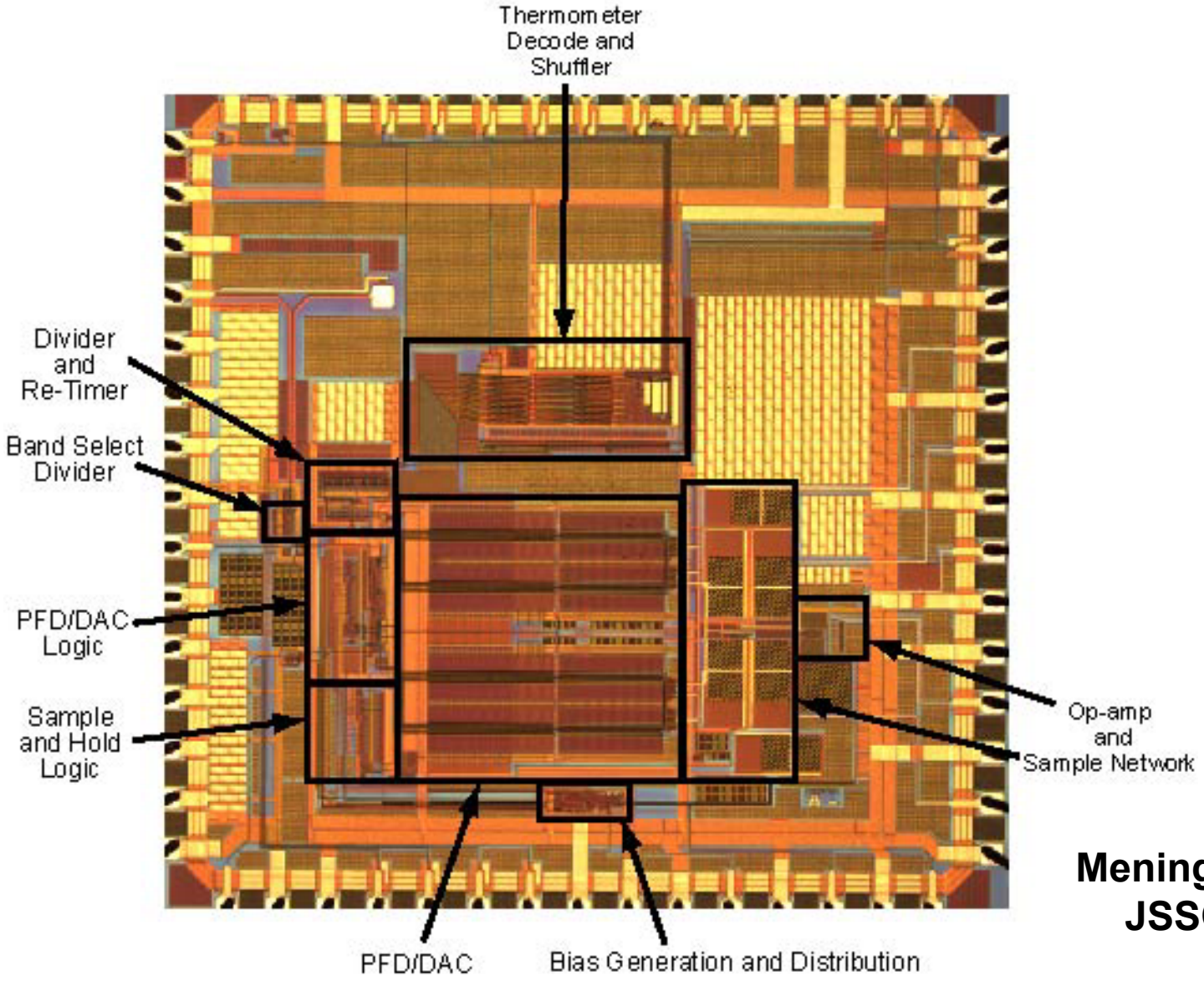
- **Simulation speed: 40 million time steps in 11 min**

Simulated PLL Phase Noise of 7-bit PFD/DAC



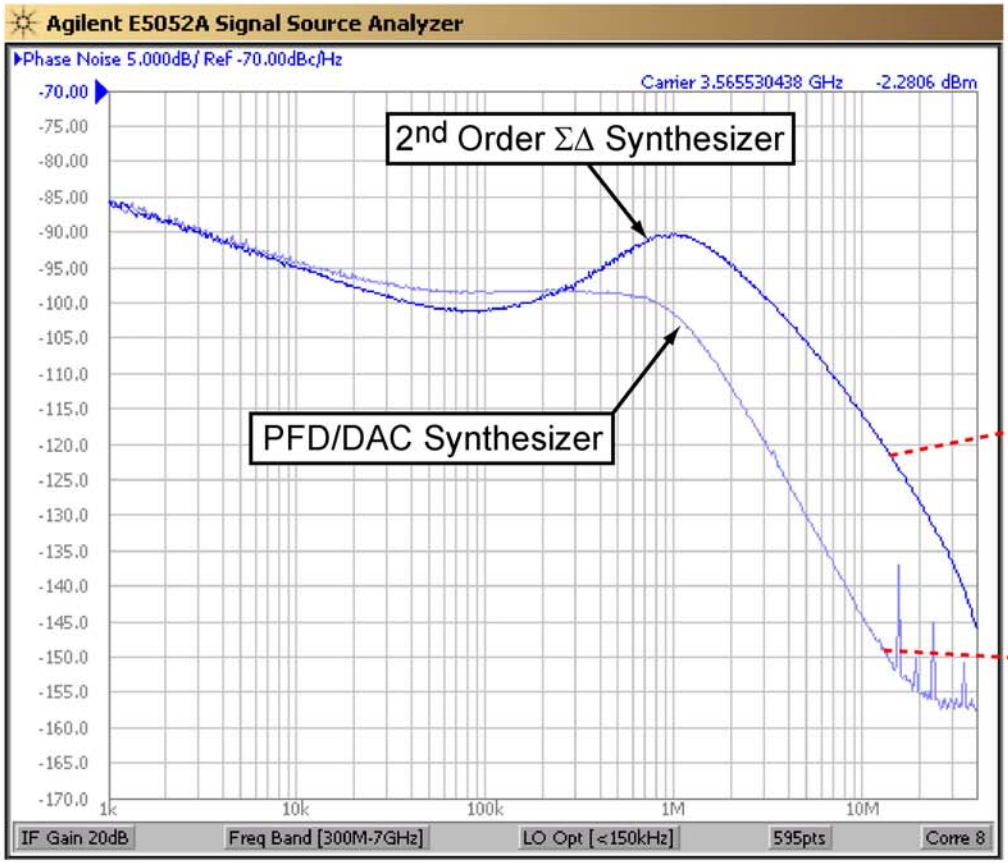
- **Simulation results allowed verification of an analytical model identifying impact of timing and current mismatch**

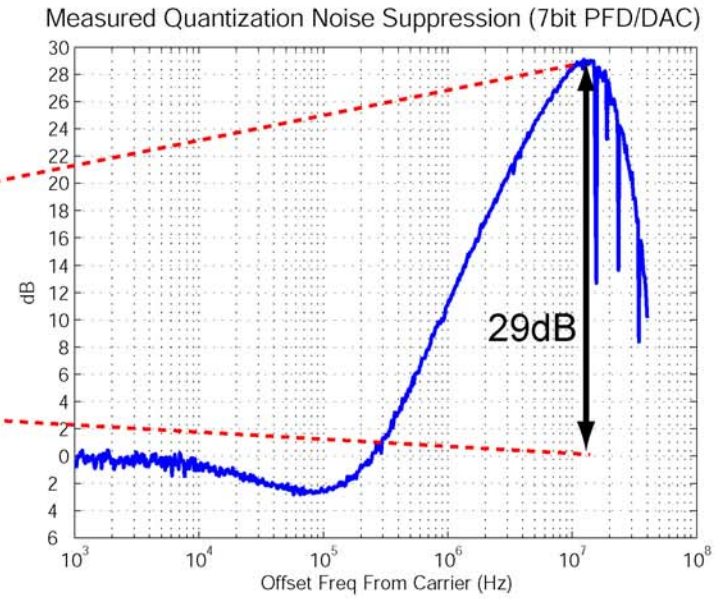
Custom IC Prototype (0.18u CMOS)



Meninger et. al., JSSC 2006

Measured Noise Suppression

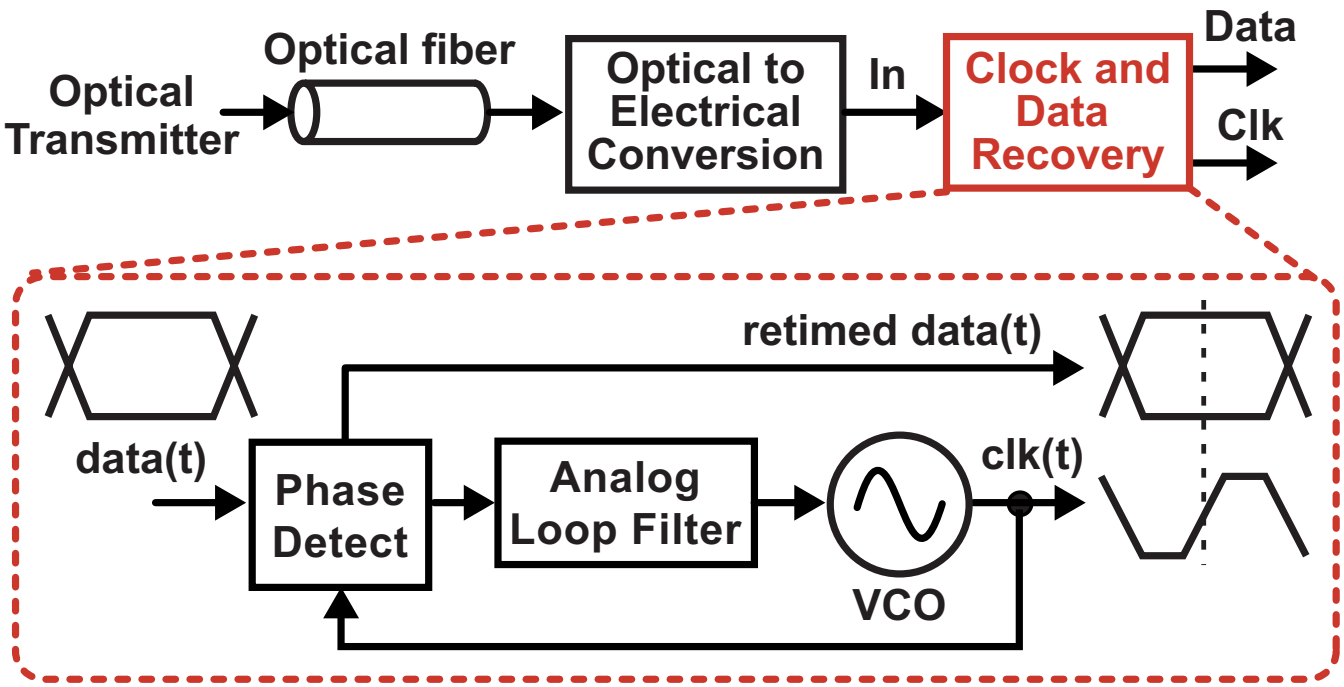




- **29dB quantization noise suppression measured at 10MHz !**

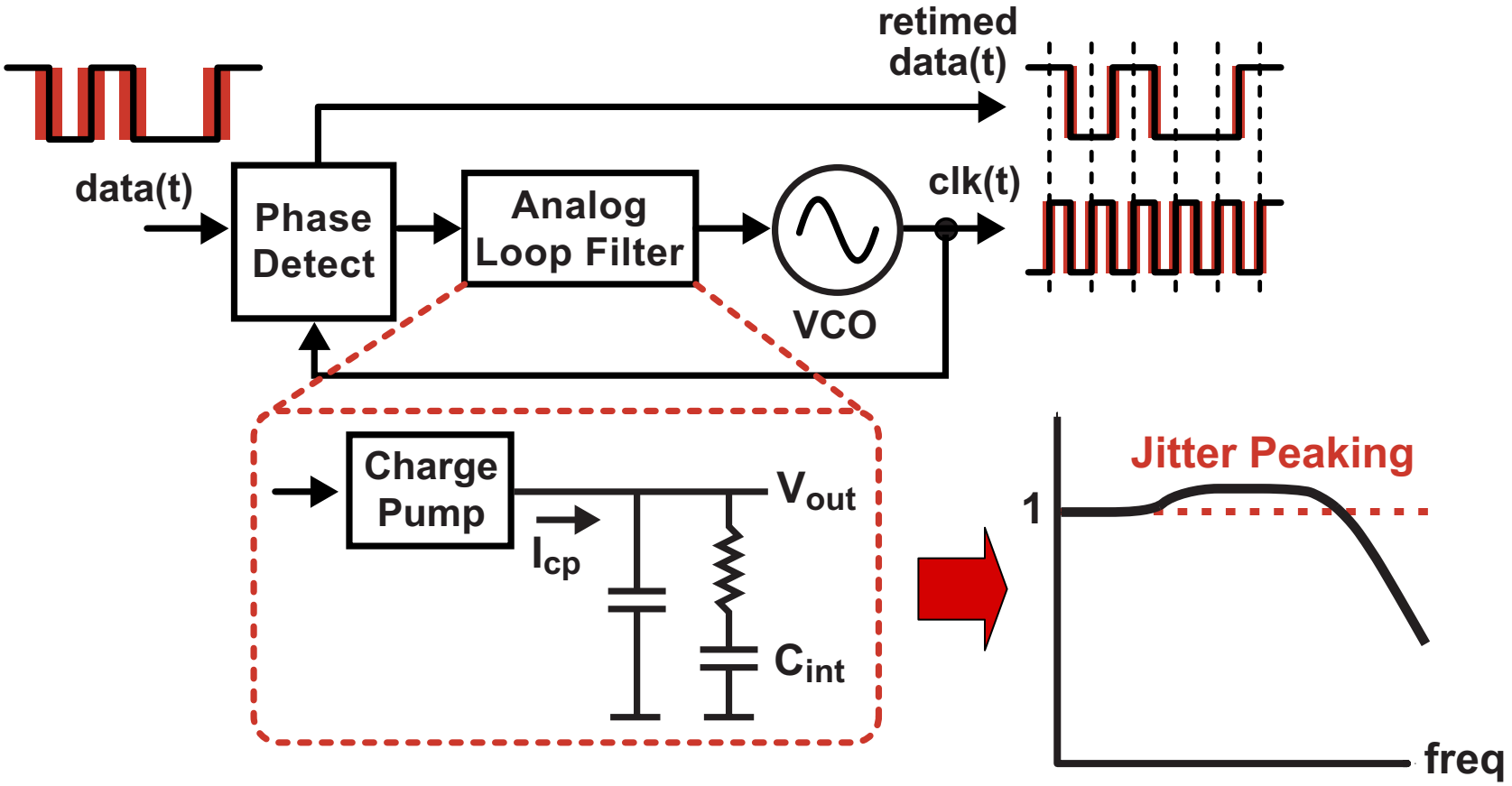
First pass silicon success with a new synthesizer architecture!

Clock and Data Recovery Circuits for SONET



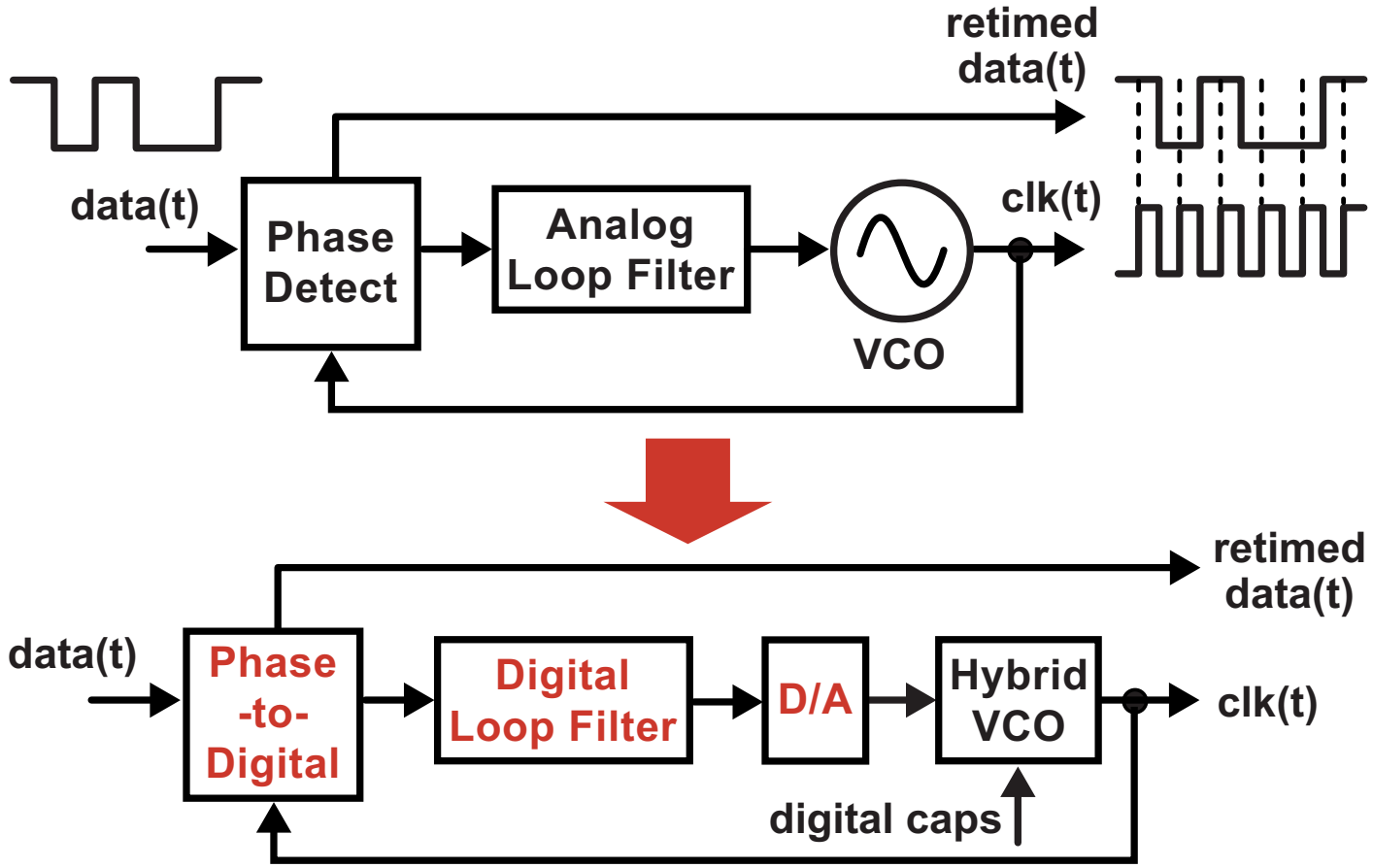
- Function: recover clock from input NRZ data stream
  - Want to support multi-rates: 2.5 Gb/s, GbE, 622 Mb/s, 155 Mb/s
- Structure: phase-locked loop with an appropriate phase detector
- Focus: fully integrated implementation

The Woes of an Analog PLL Implementation



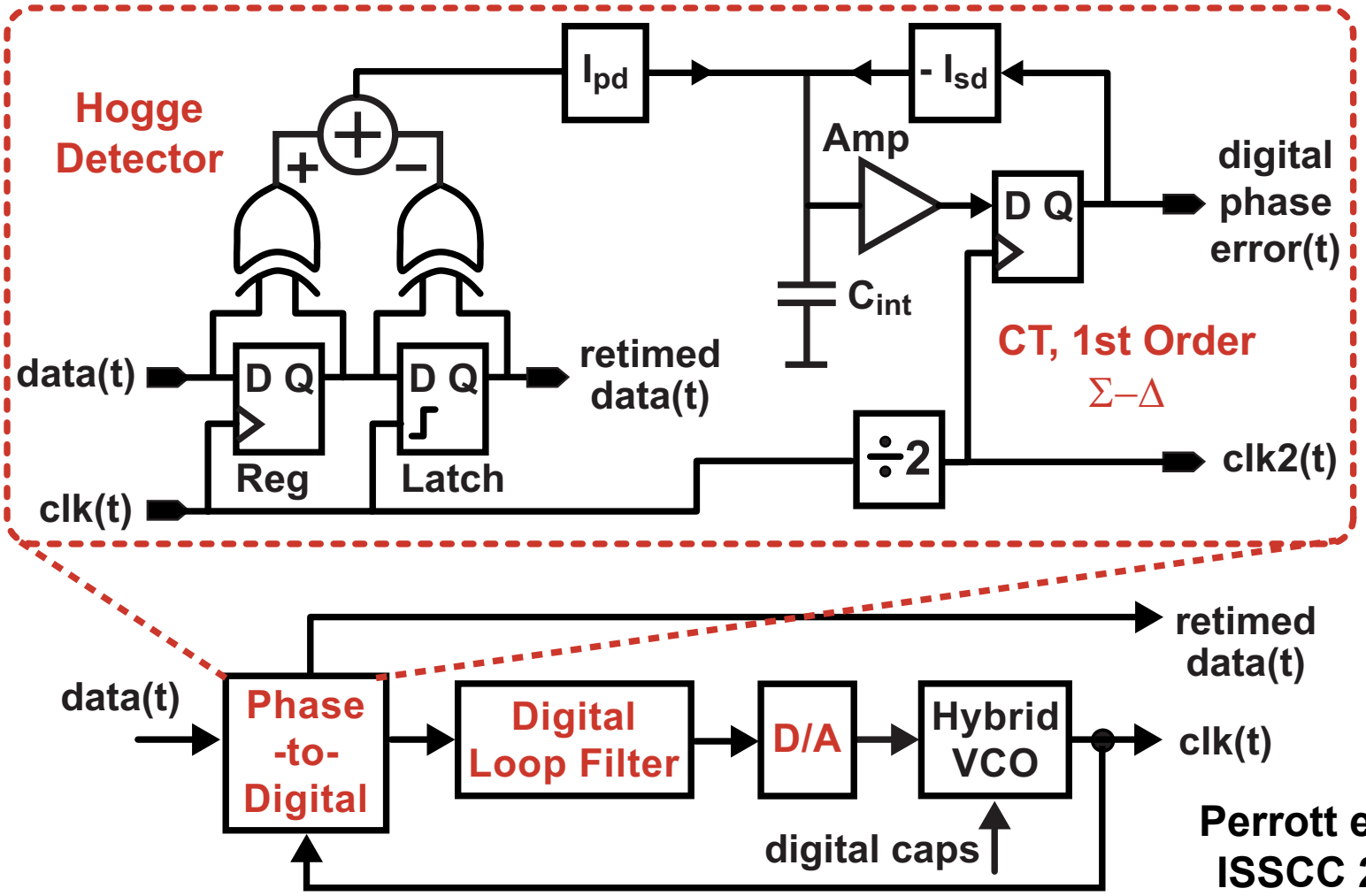
- Goal: integrated loop filter
- Issue: required cap value too large (100 uA charge pump):
  - 2.5 Gb/s (OC-48): 4 nF
  - 155 Mb/s (OC-3): 64 nF

Consider A Digital CDR …



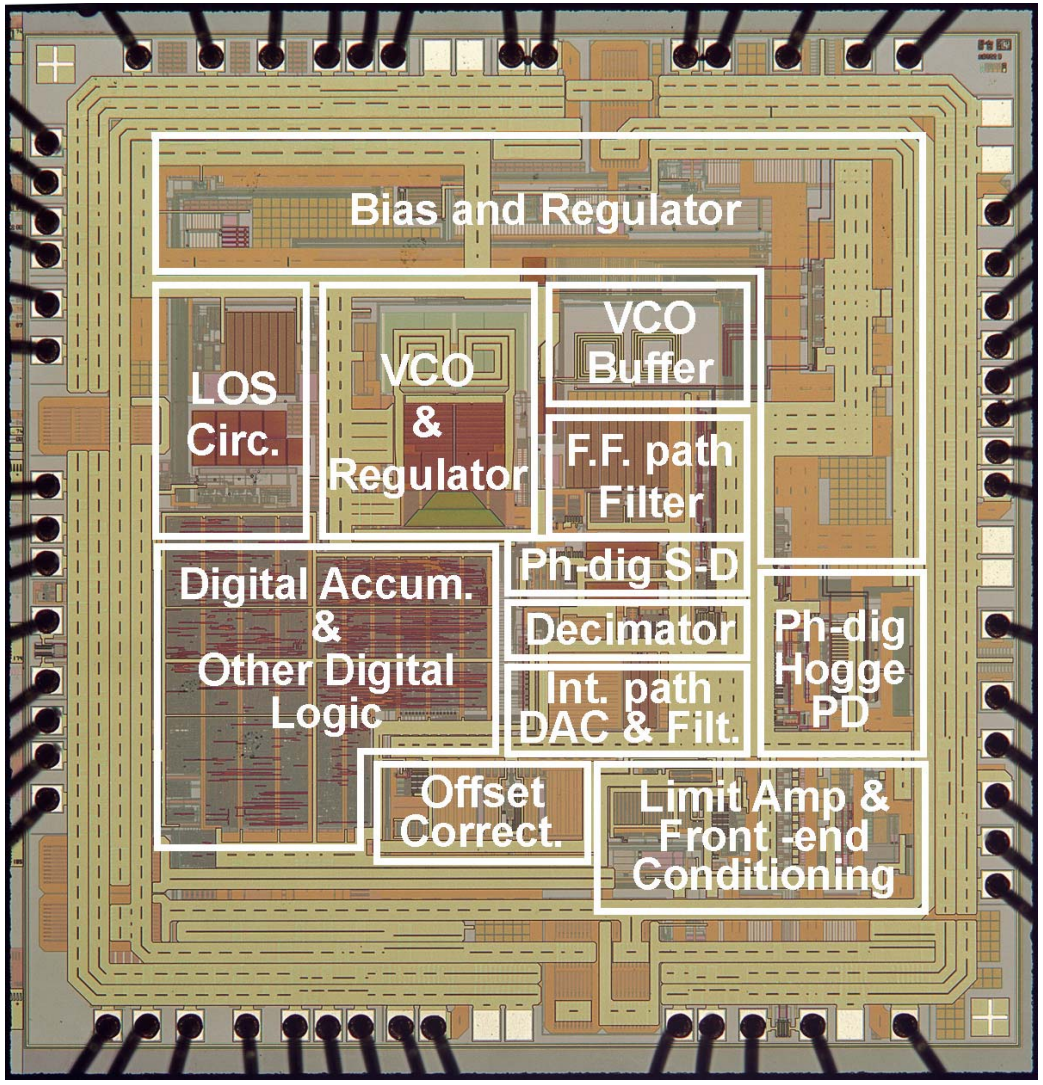
- Digital loop filter allows:
  - Easy integration of loop filter
  - Easy configurability for multi-rate operation

Key Component: A New Hybrid Phase Detector



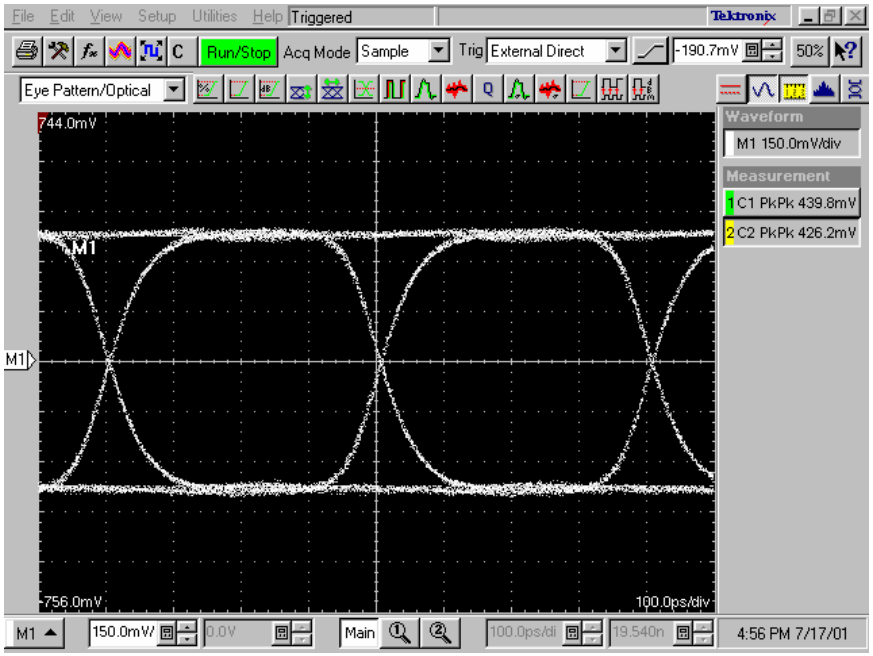
- Combines analog phase detector with a simple A/D
- Use simulation to verify CDR performance with this structure

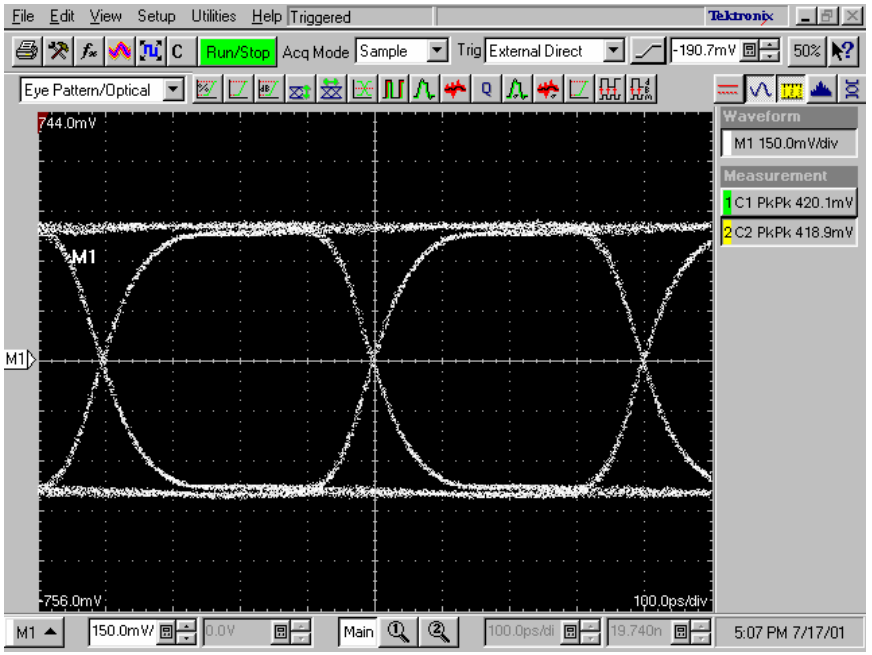
Custom IC Implementation (0.25u CMOS)



- Achieved all SONET performance requirements on first pass silicon
- Measured results in line with behavioral simulations

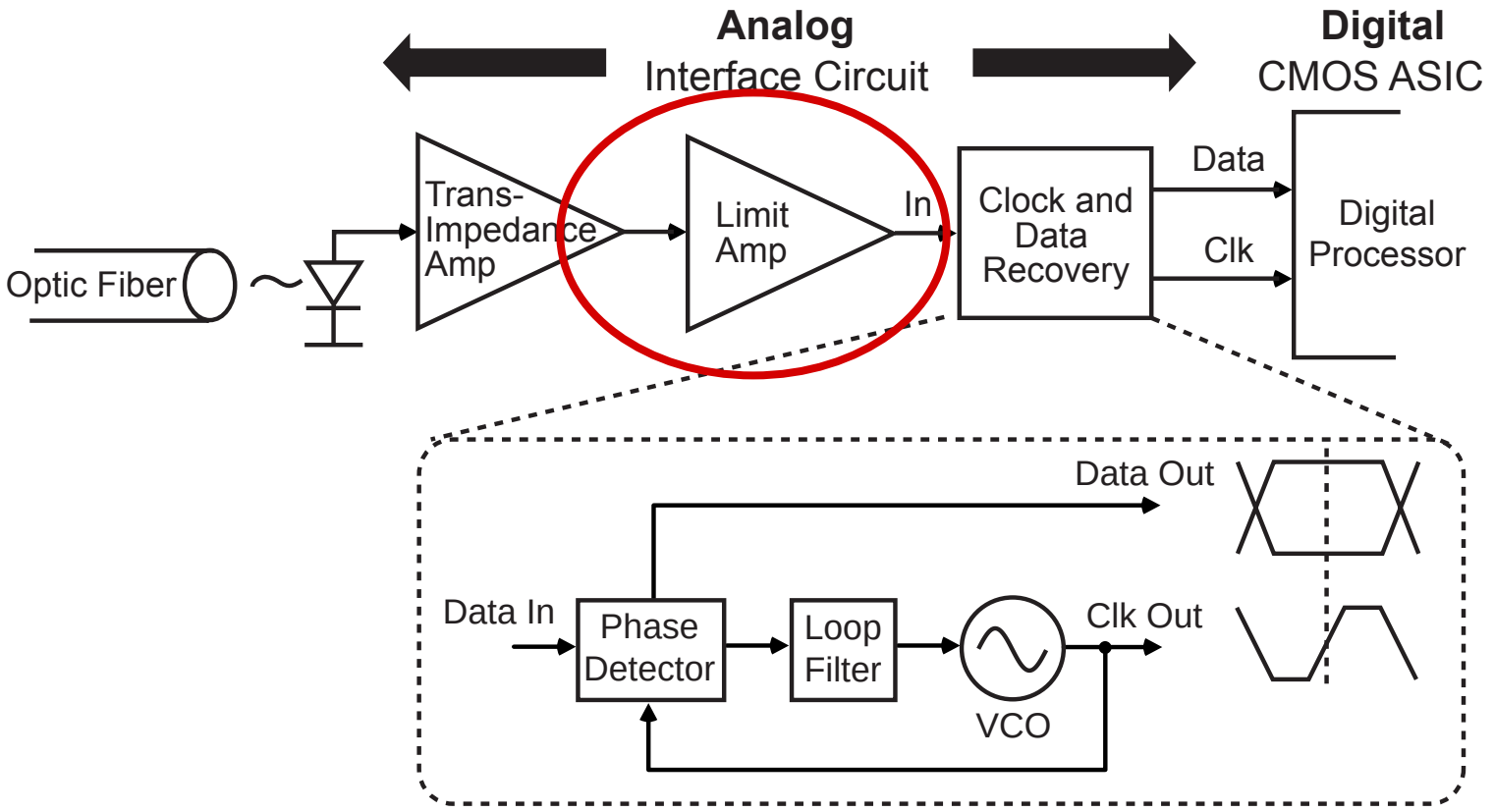
Measured Eye Diagrams at 2.5 Gb/s





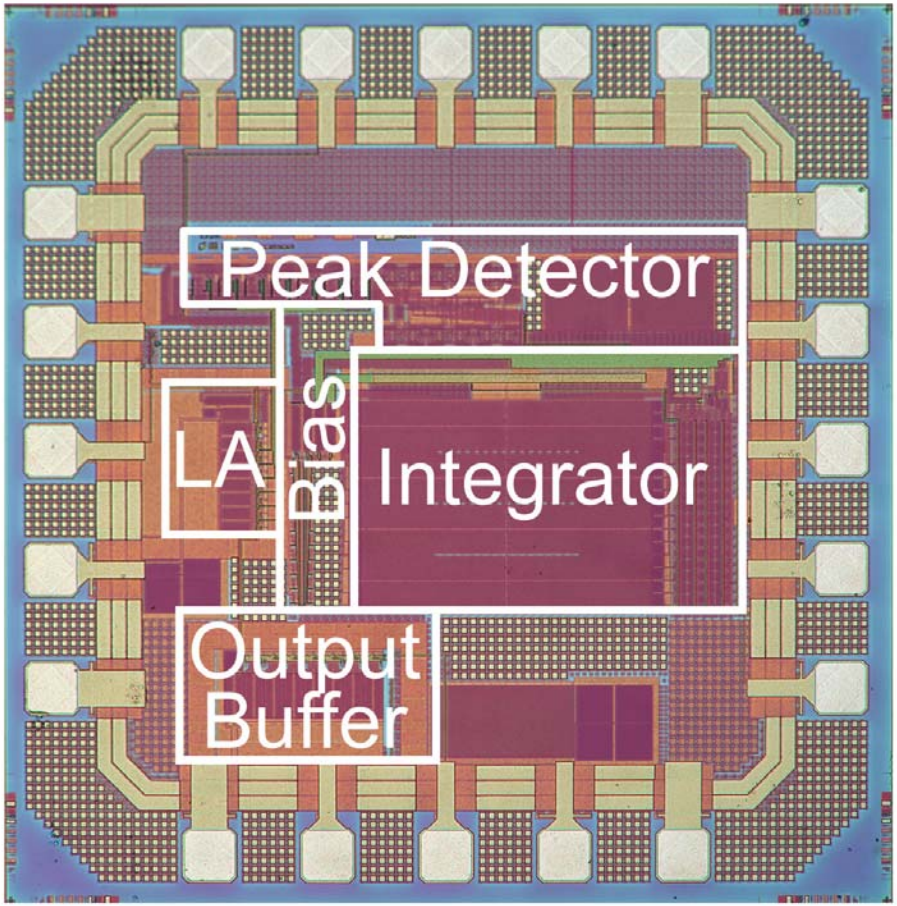
- **Best Case Conditions**
  - Input data: 2 Vp-p diff.
  - Pattern: PRBS 2^7-1
  - Temperature: 25° C
  - Jitter: 1.2 ps RMS
- **Worst Case Conditions**
  - Input data: 10 mVppd
  - Pattern: PRBS 2^{31}-1
  - Temperature: 100° C
  - Jitter: 1.4 ps RMS

A Fast Acquisition Limit Amp



- **Acquisition time of CDR is limited by slow response of limit amp offset correction loop (typically milliseconds)**
- **Research focus: improve speed of offset correction**

Custom IC Implementation (0.18u CMOS)

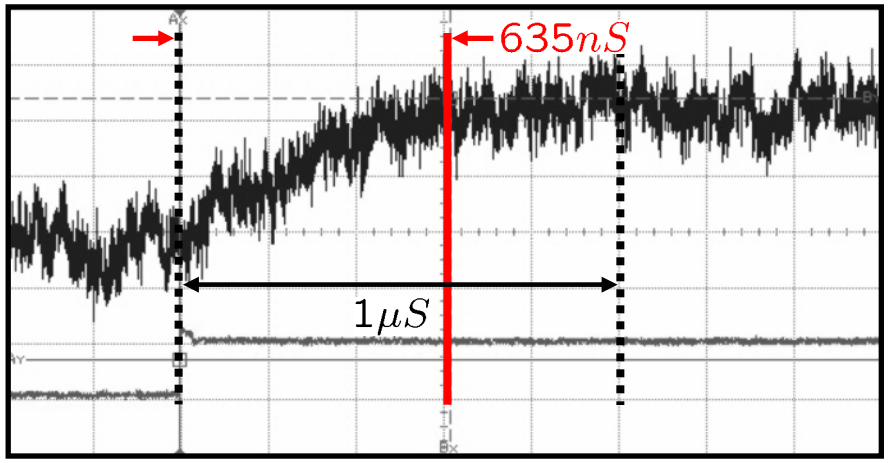


Crain et. al., ISSCC 2006

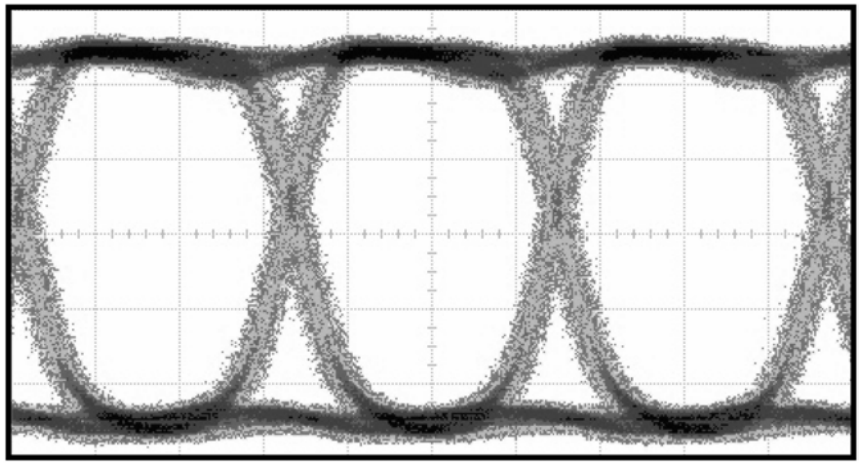
- Achieved *3 orders* of magnitude improvement in offset compensation time over conventional approach!
  - Key innovation: a high speed, low droop CMOS peak detector
- CppSim used to understand impact of peak detector on system
- First pass silicon!

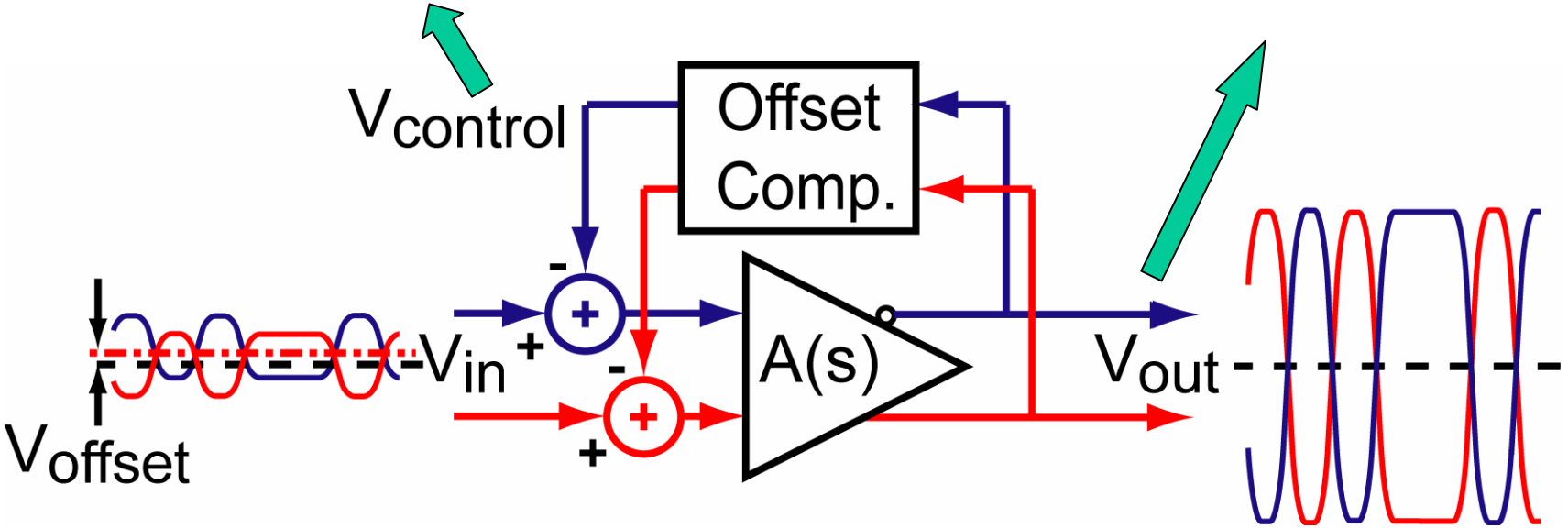
Measured Results at 3.125 Gb/s

Settling time (< 1 microsecond)

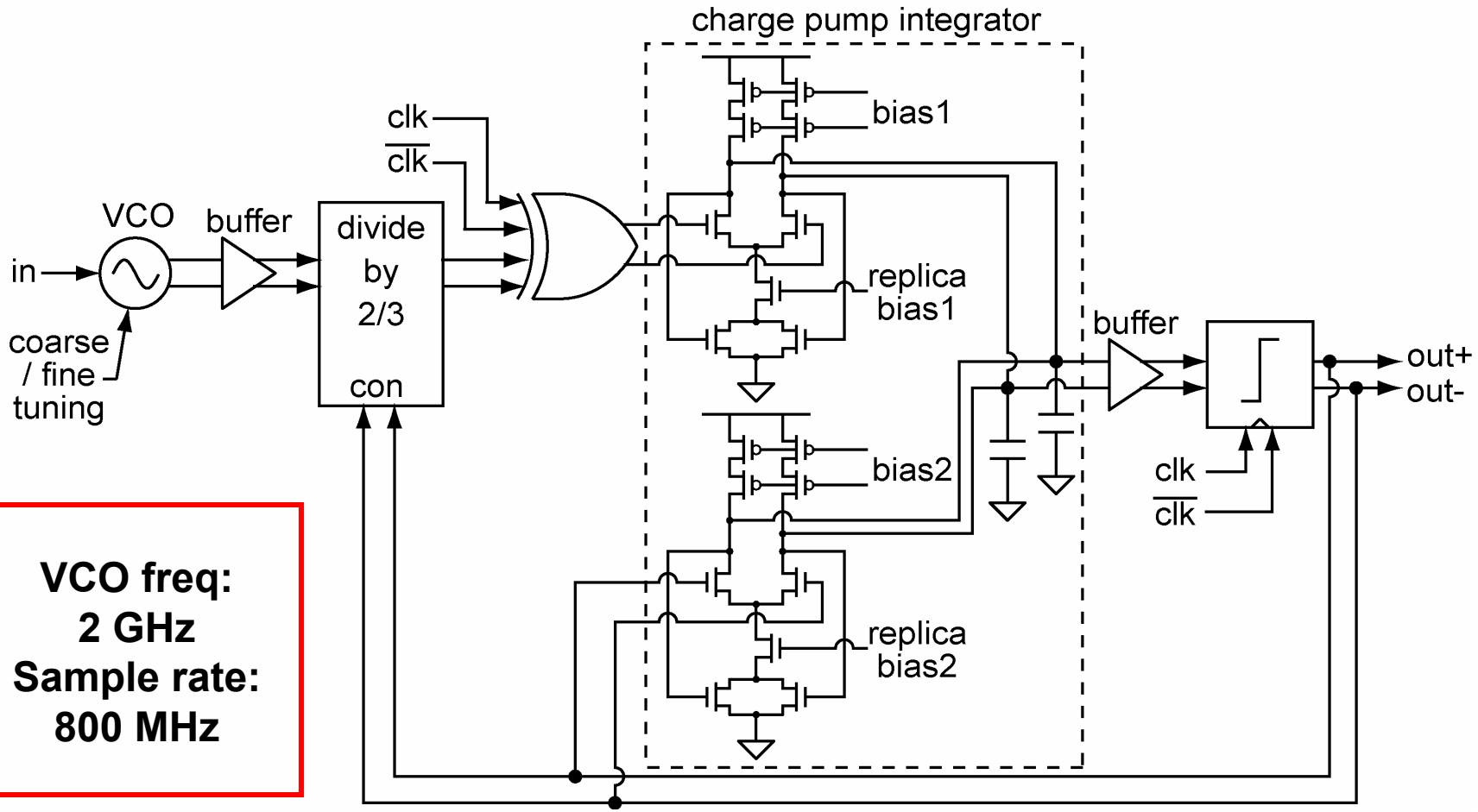


Eye Diagram





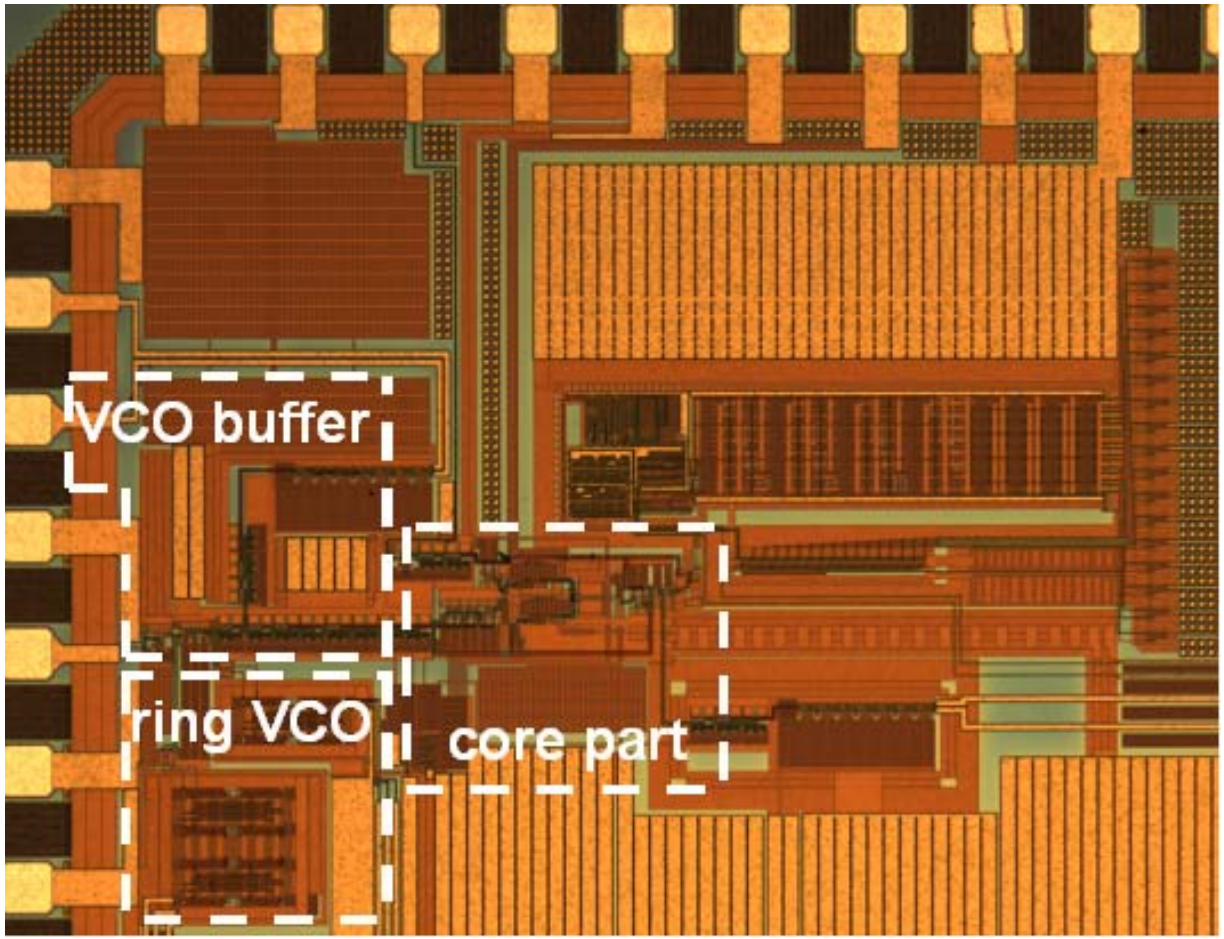
A Novel VCO-based A/D Structure



VCO freq:
2 GHz
Sample rate:
800 MHz

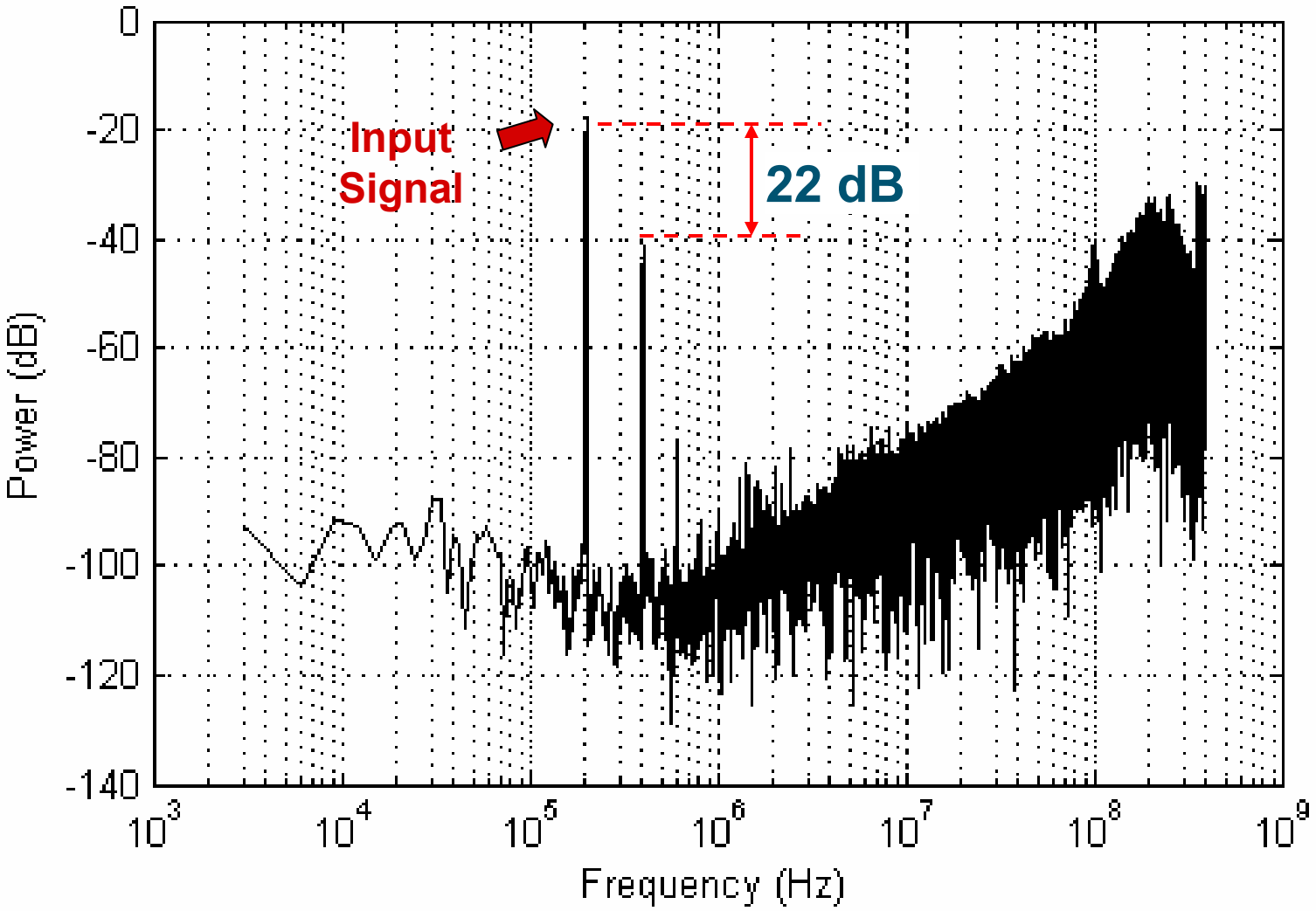
- Output frequency/phase of VCO set by input voltage
- Measure phase of VCO to infer input voltage

Custom IC Prototype (0.18u CMOS)

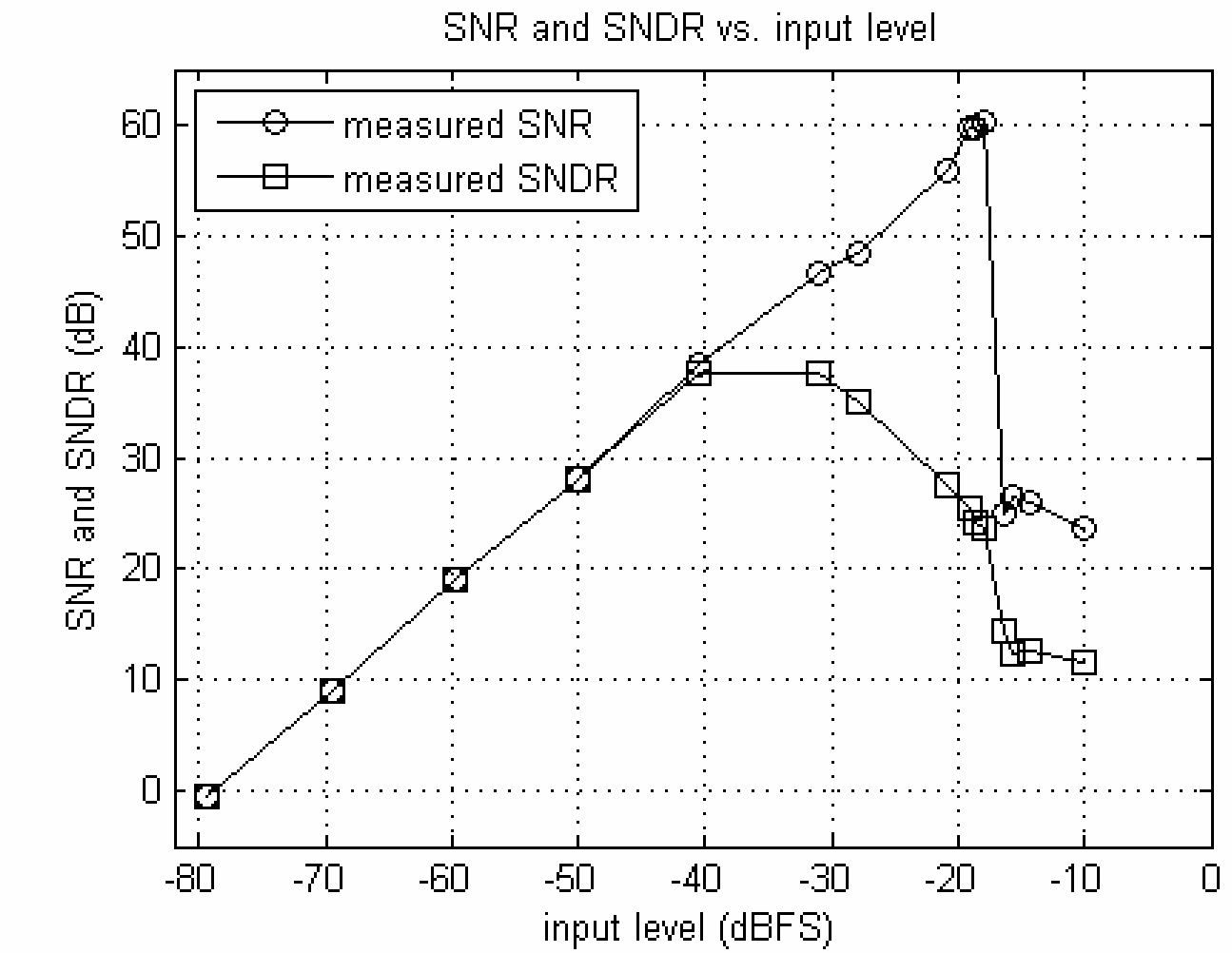


- **Designed by Min Park**

Measured output spectrum (Hann Window)



SNR and SNDR (over 1 MHz) vs. input amplitude



- **Peak SNR: 60 dB Peak SNDR: 39 dB (Hann window)**
- **Measured performance in line with behavioral simulations**

Conclusion

- C++ behavioral models allow fast and accurate simulation of novel mixed-signal circuits
  - Allows the designer to understand key issues *before* spending time on transistor level circuits
- C++ models allow seamless bootstrapping into a variety of simulators
  - AMS Designer
  - NCVerilog
  - Matlab
  - CppSim
- VppSim dramatically simplifies the integration of these models in each of the above simulators