On-chip samplers for test and debug of asynchronous circuits

Frankie Liu, **Ron Ho**, Robert Drost, Scott Fairbanks
Sun Microsystems Laboratories

Chip test is a major part of design

- Testing is expensive in headcount and equipment
  - For example, the time to debug ("root-cause") a chip problem [1]

Circuit or layout error, discovered on automated test equipment (ATE)	2 person-weeks
Circuit marginality, discovered on ATE	2 person-months
Logic error, or any error discovered on system (after ATE)	2 person-years

- …and the stakes are even higher
  - Intel's FDIV problem was a $500M mistake
- Test is a critical area for asynchronous circuits
  - Test/debug methodology will decide production worthiness

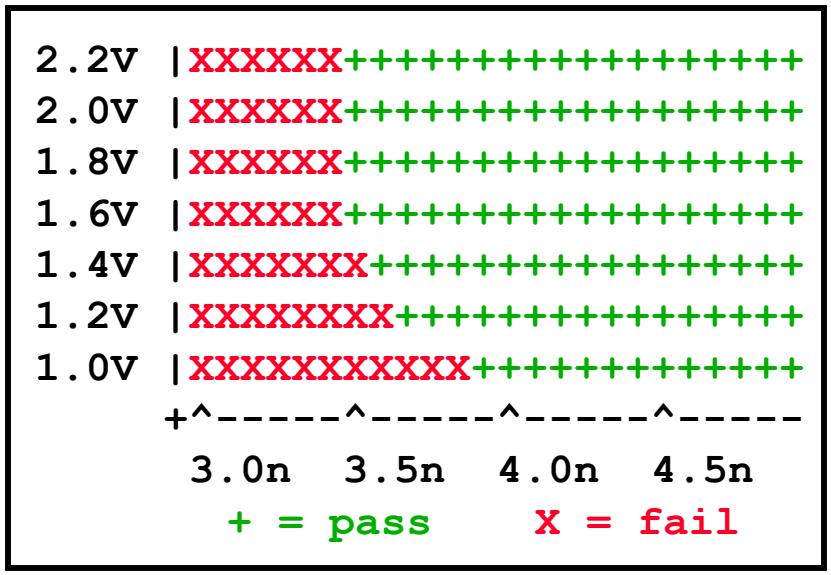
[1] Jason Stinson, Intel

1. Traditional (synchronous) testing

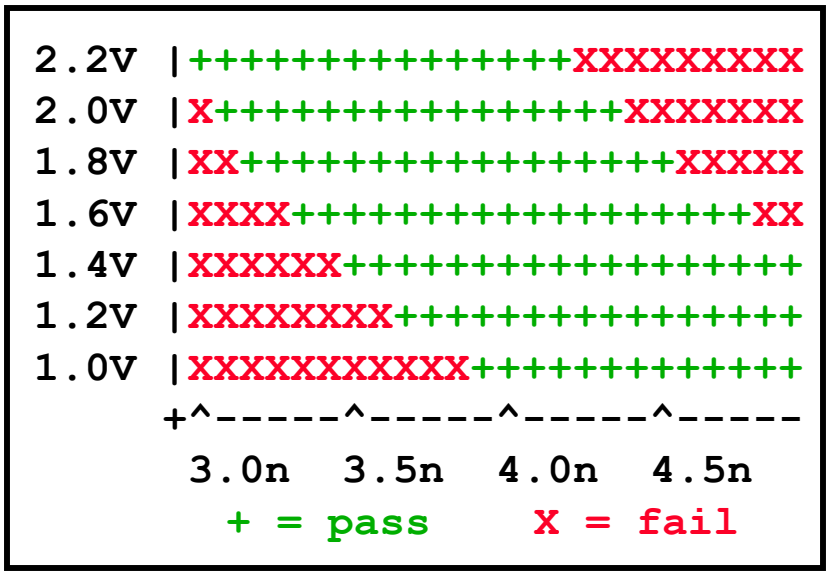
"If you didn't test it, it didn't work."

Inference testing: Shmoo plots

- Vary two parameters, typically voltage and period
  - Run tests on the chip and examine result profiles



RC-dominated path

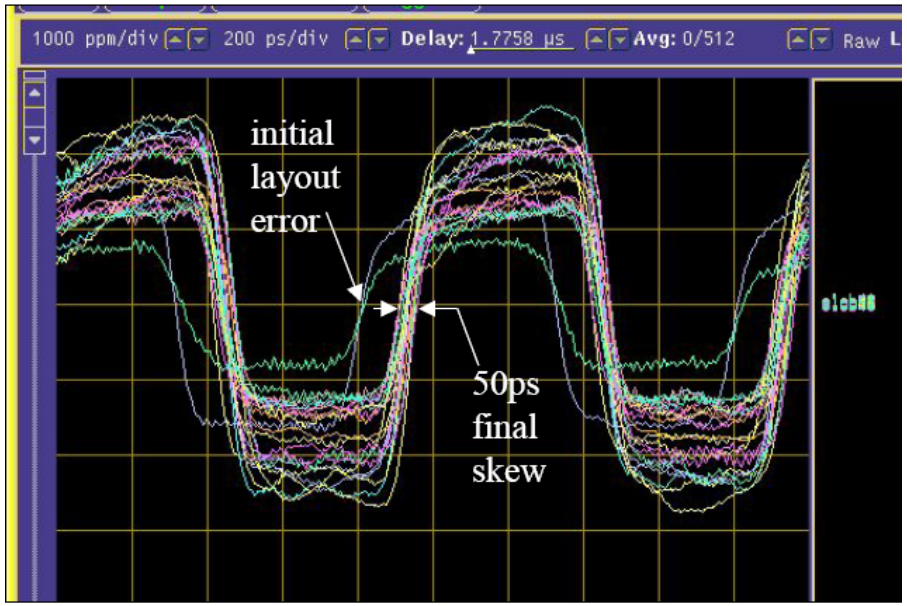


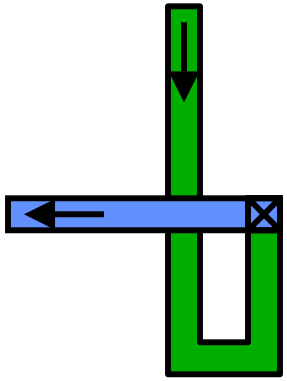
Leakage problem

- Limitation: cannot easily check specific circuits

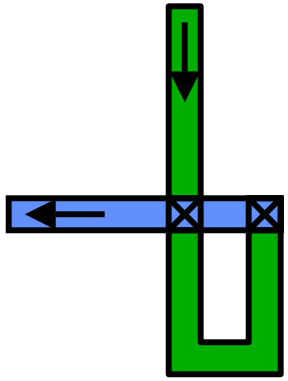
Probed testing: E-beams or lasers

- Aim beams of electrons or photons at a chip
  - > Measure the beam reflectivity, which depends on voltage bias





Intended layout:
Match delays
using RC length



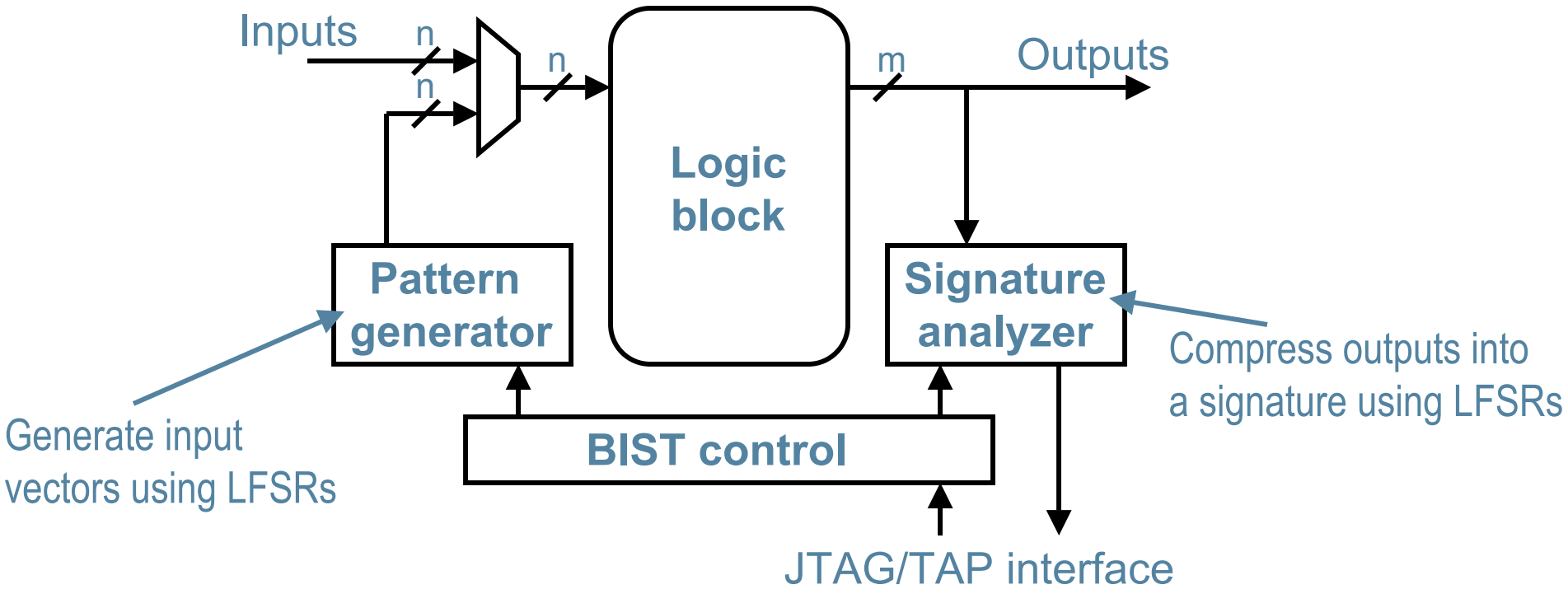
Actual layout:
After "robust
VIAs" CAD flow

- Limitation: Invasive (unpackaged chips, thinned Si)
- Limitation: Captures transitions but not voltages

Waveforms from Ferd Anderson, ISSCC 2002

BIST testing: Signature analysis

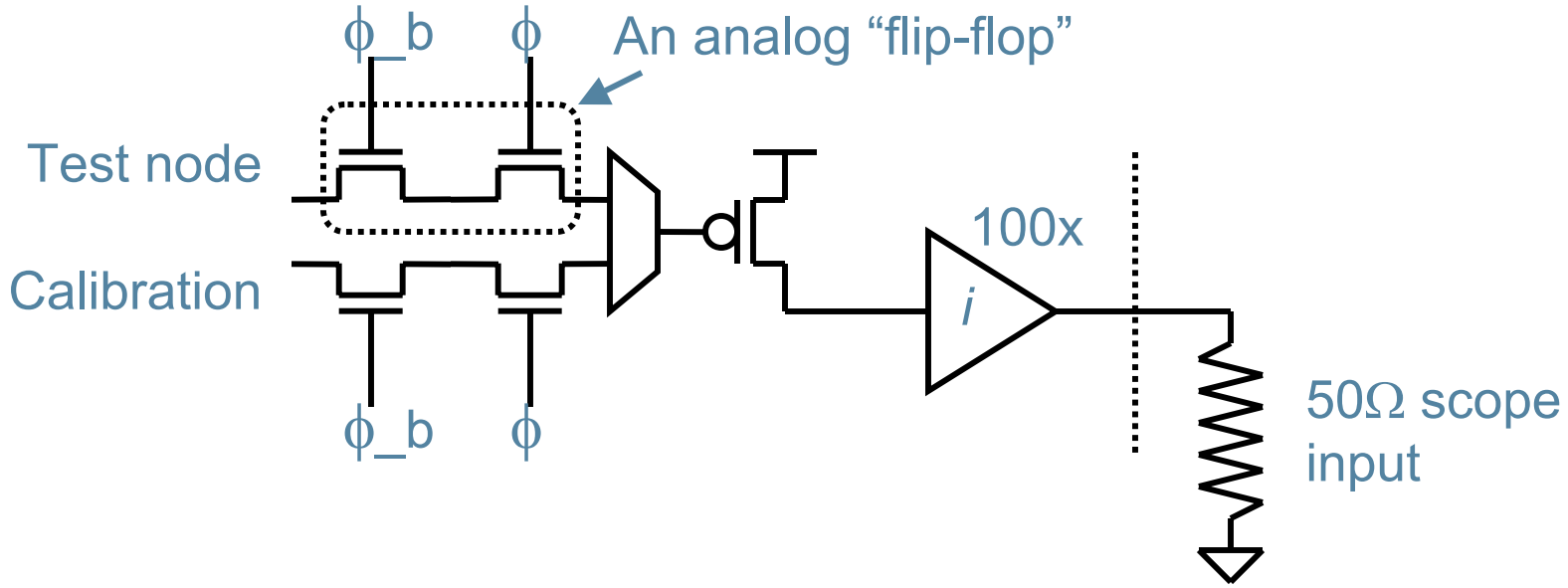
- Surround a logic block with input and output LFSRs



- Limitation: Generally restricted to digital signals

BIST testing: Analog samplers

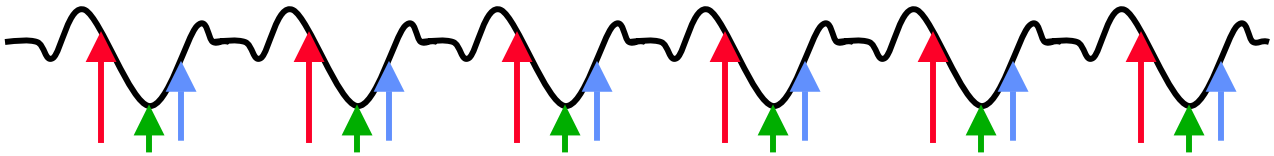
- Exploit transistors' high bandwidth and low load
  - > Use an analog "flop" to bring input voltage to the PMOS gate
  - > Calibrate for V-I transfer function and per-sampler variations
  - > Build the circuit *in-situ* to test nodes of interest



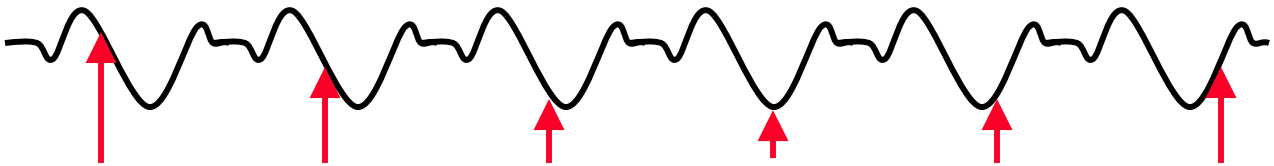
  - > Clock this type of sampler using one of two methods…

Two ways to clock samplers

- $T_{clk\text{-}sample}=T_{clk\text{-}chip}$: Get the same point each cycle
  - > After you take the measurement, change the phase and repeat



- $T_{clk\text{-}sample}=T_{clk\text{-}chip}+\delta$: Walk through the waveform
  - > Over many cycles you *subsample* the entire waveform



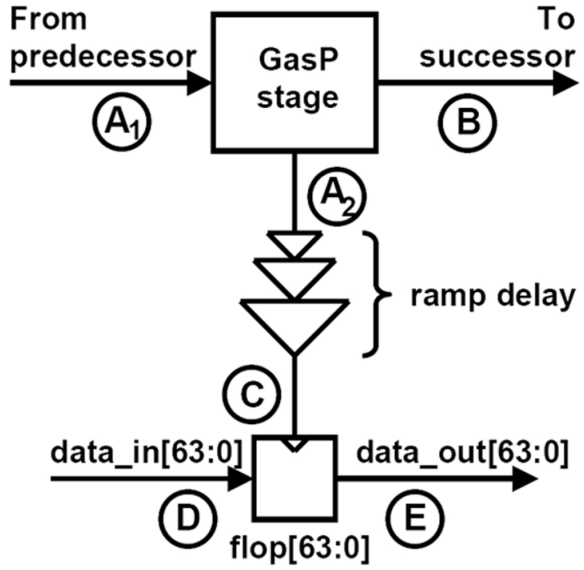
- Limitation: Best used for synchronous circuits
  - > Question: how could we sample asynchronous circuits?

2. Asynchronous samplers

"If you don't build a testability circuit, you will need it.
If you do build it, you won't."

A very simple idea

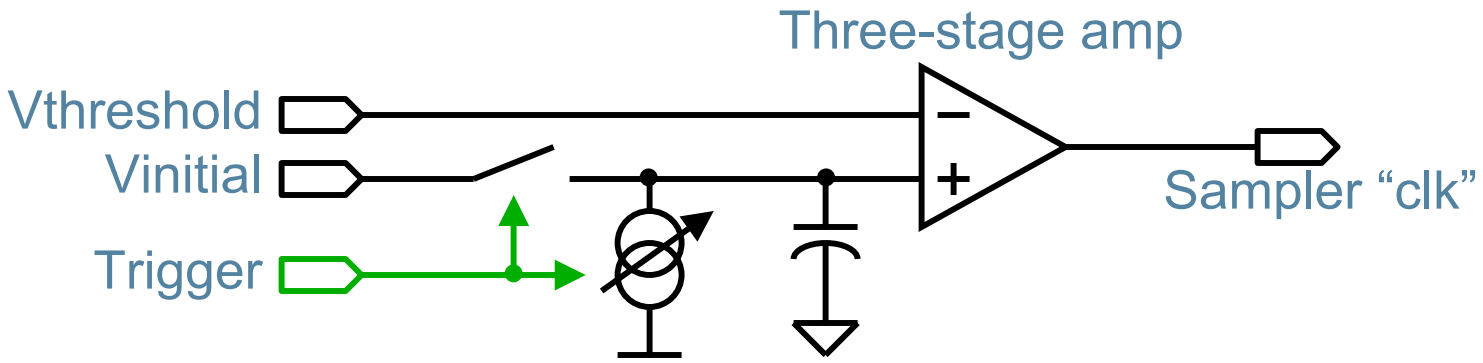
- Keep the sampler body but generate a strobe
  - Triggered off an (asynchronous) *leading* event
  - From the trigger, wait a fixed (adjustable) delay, then strobe
  - After you've taken the measurement, adjust the delay
- Can use A2 as trigger
  - To sample B, C, D, and E
  - Or use A1 as the trigger



- The trick is generating a stable adjustable delay

Timer-generated delays

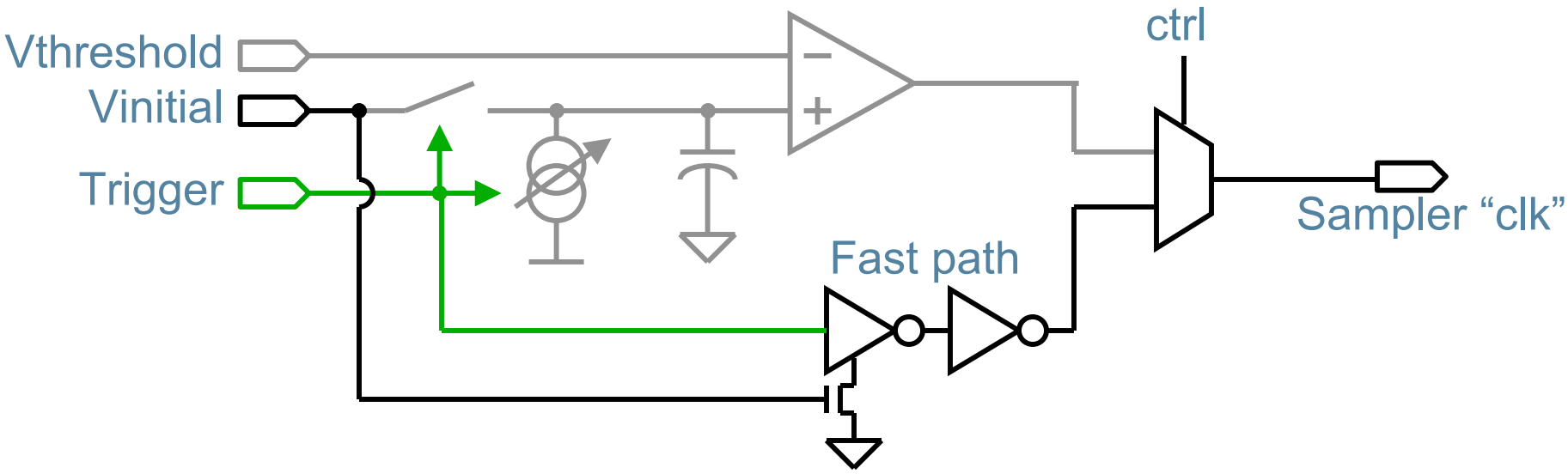
- Before trigger, precharge a cap to an initial voltage
- At trigger, start pushing charge onto the cap
  - When voltage crosses a threshold, fire the sampler



- Can adjust the delay via several parameters
  - Vinitial, Vthreshold, charging current (but beware slope issues)

Timer-generated delays, con't

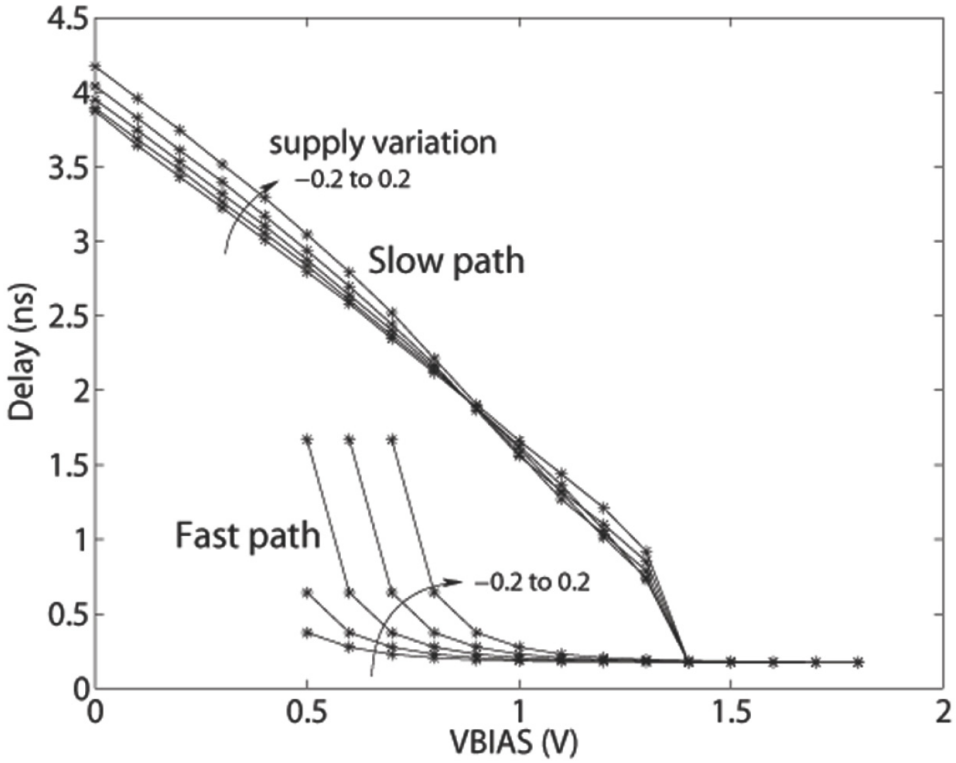
- Insufficient dynamic timer range at low (fast) end
  - > So add a fast path in parallel, and pick one of the two paths



- Time calibration as important as voltage calibration
  - > Simple (if inelegant) method: sample the delay line output

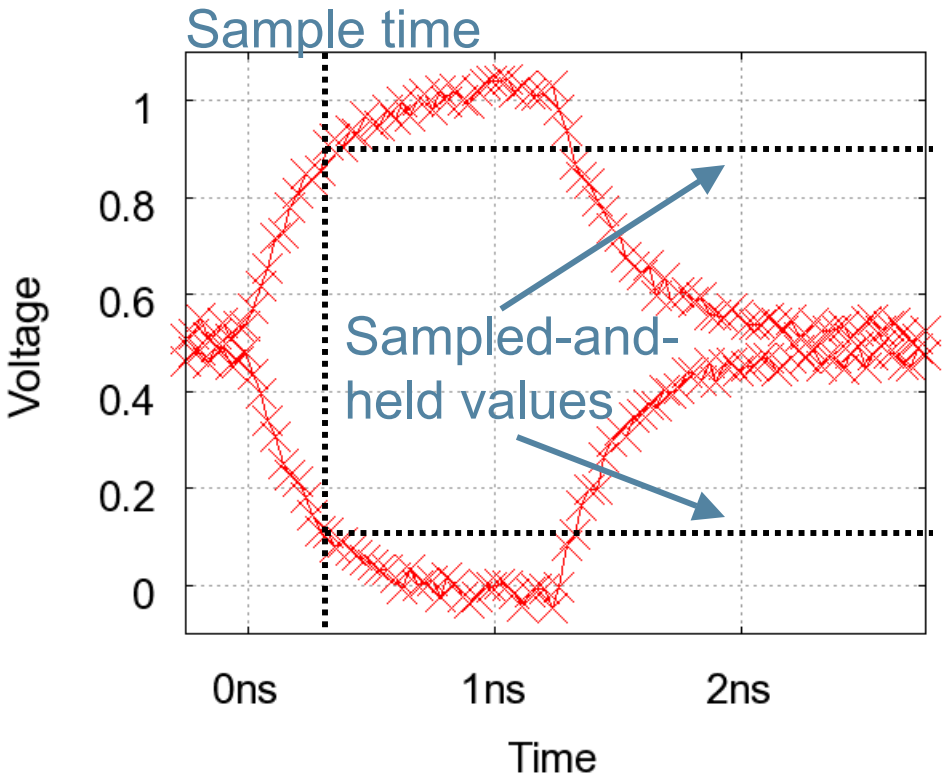
Timer-generated delays, con't

- Make efforts to reduce (not eliminate) noise effects
  - > Feed in external currents and external voltage references
- Simulated sensitivity to power supply noise



Repeated one-shot sampling

- Often useful to sample-and-hold ("-for-a-long-time")
  - > Allows sampling of multi-value voltage paths (e.g., data eyes)
- Logically gate the timer clock with a user signal
  - > Take only a single sample…
  - > …then wait until user resets
- Gets multiple trajectories
  - > Sampler will track either path
  - > …not average them together

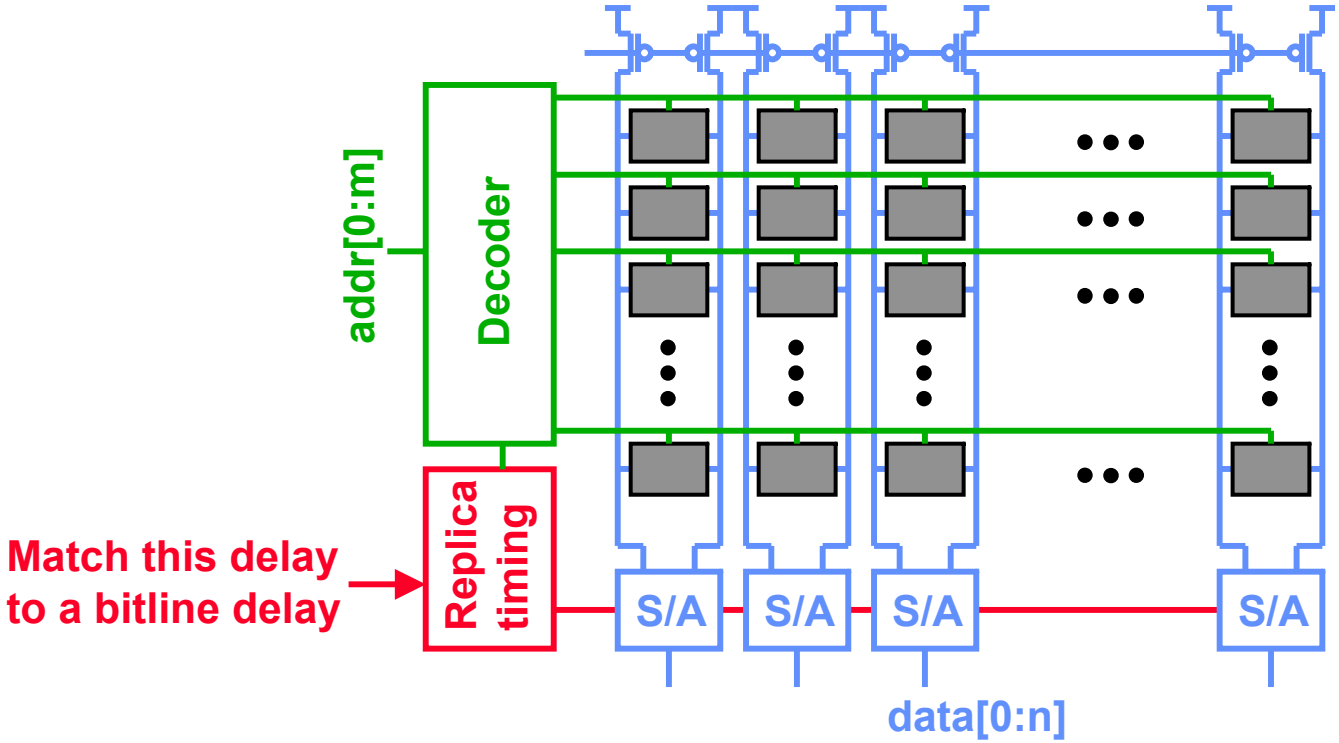


3. Applications

"In theory, there is no difference between theory and practice. In practice, there is."

Application: probing SRAM bitlines

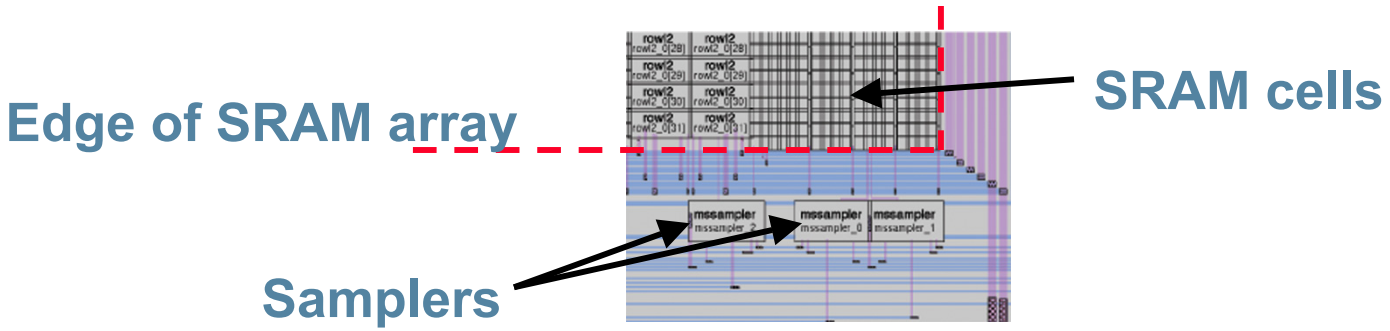
- An SRAM's read path is typically self-timed
- A typical question: What is the margin on that path?
  - Transistor mismatch and wire variation can become critical



SRAMs: Measuring margins

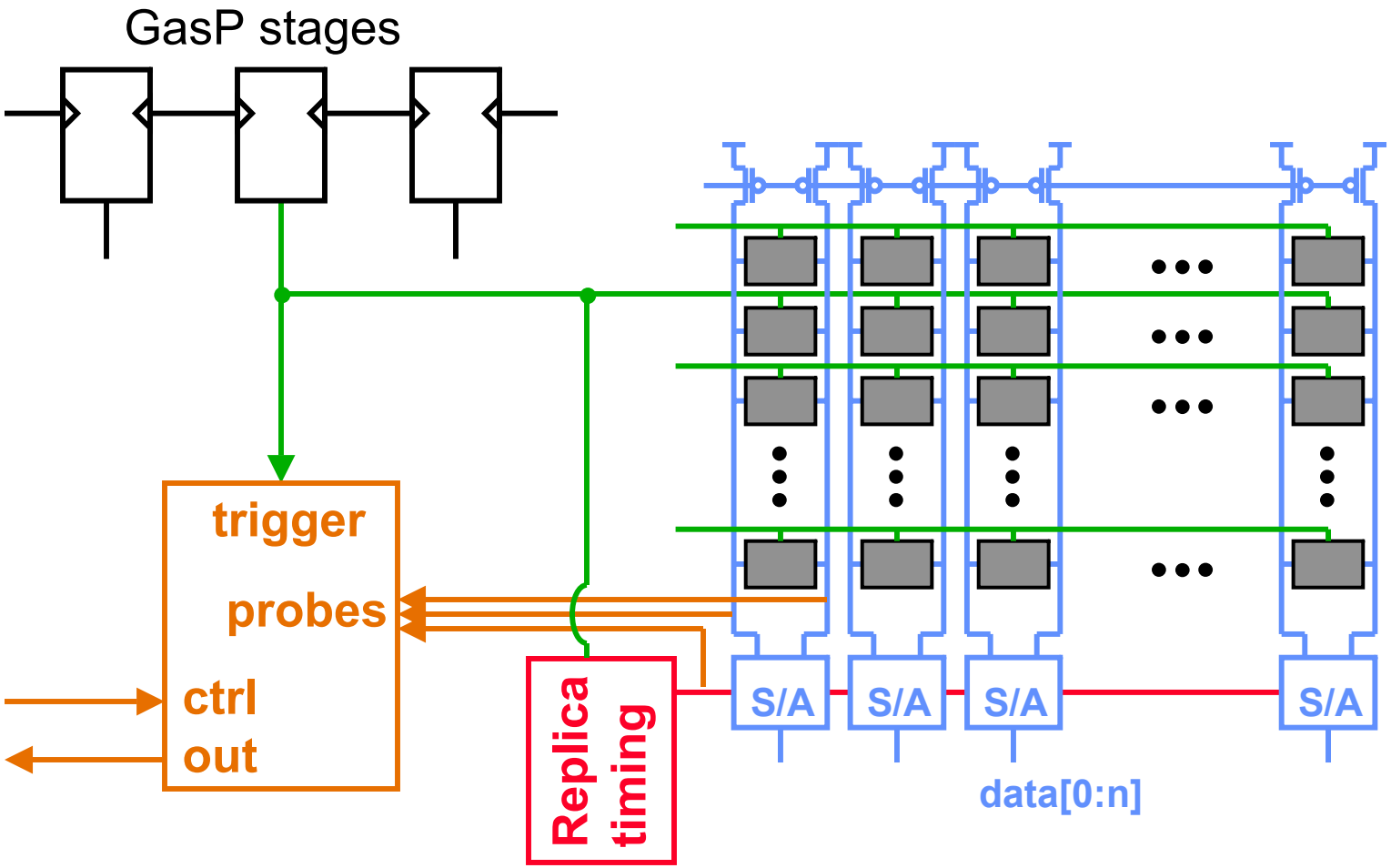
- We could make the replica delay adjustable
  - …And shmoo the replica delay with the sense-amp strobe
  - But this introduces additional variation in the path

- Better: asynchronously sample bitlines and strobe
  - Keep sampler load + routing < 5% of the bitline capacitance
  - Probe bitlines at the edge of the memory array (for area reasons)



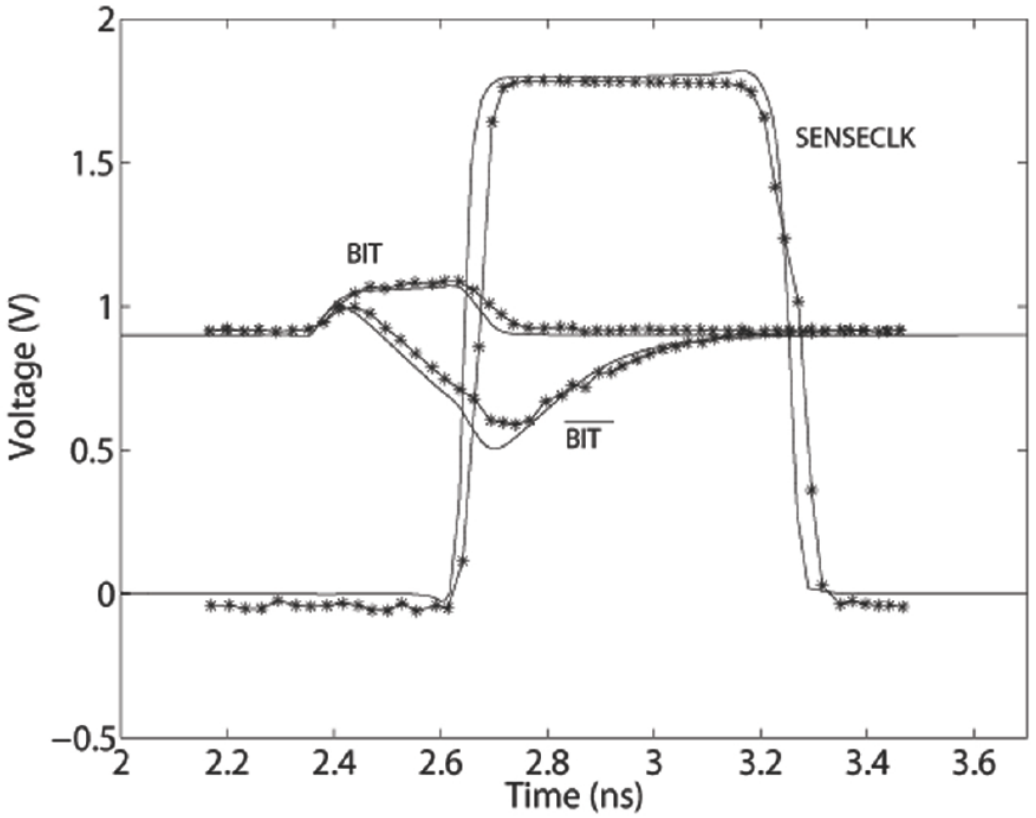
SRAMs: A first experiment

- Use a GasP chain to trigger an SRAM and samplers
  - Activates: WL, replica WL, and three samplers (bit, bit#, strobe)



SRAMs: Simulated results

- We ran 50 read cycles of a 128-word, 8Kb memory
  - > Estimated parasitics in a 180nm 1.8V TSMC technology
  - > Uneven arrival times emulate an asynchronous memory

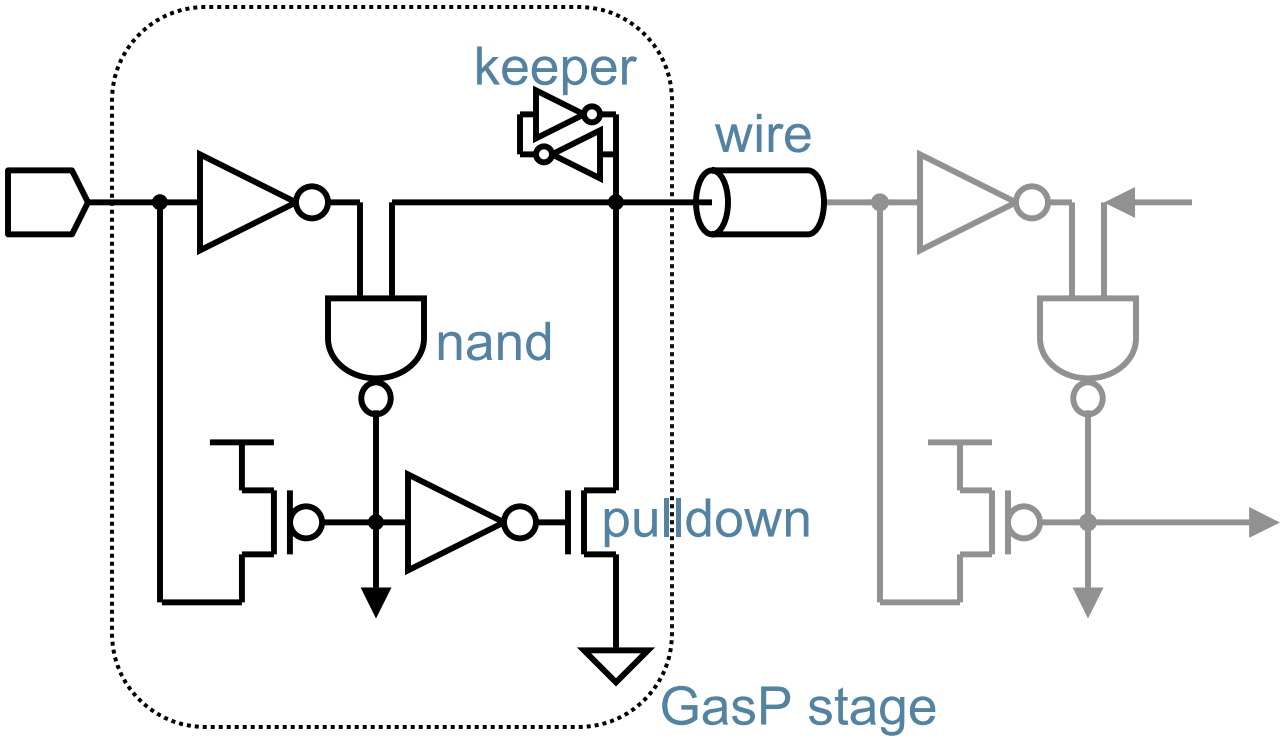


Solid lines are plotted directly from a waveform viewer

Dots are reconstructions from a full sampler path (measuring the output current to a 50W off-chip resistance) after calibration

Application: checking GasP margins

- GasP handshake circuits store token state on wires
- A question: what are the margins on these wires?
  - Careless design can lead to pathological circuit behavior

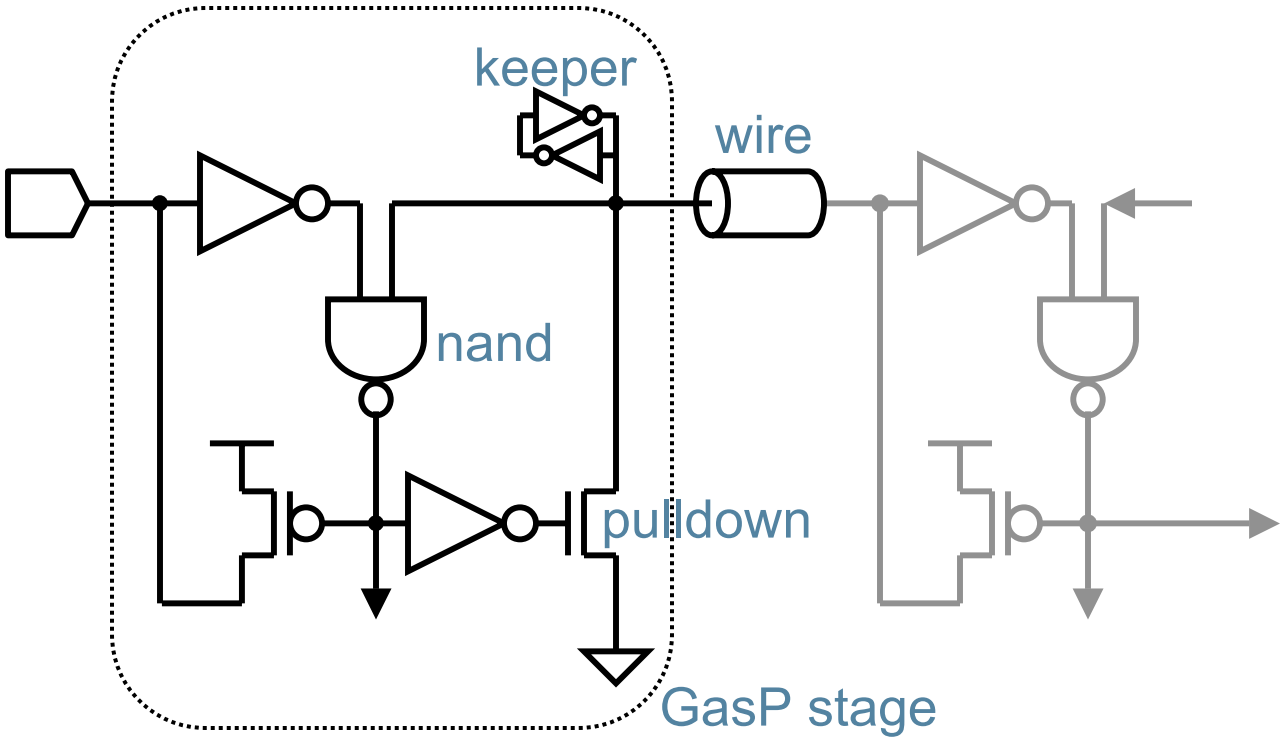


A "perfect storm" of careless design:

1. Weak pulldown
2. Long wire
3. High nand Vth
4. Low keeper Vth

Application: checking GasP margins

- GasP handshake circuits store token state on wires
- A question: what are the margins on these wires?
  - Careless design can lead to pathological circuit behavior



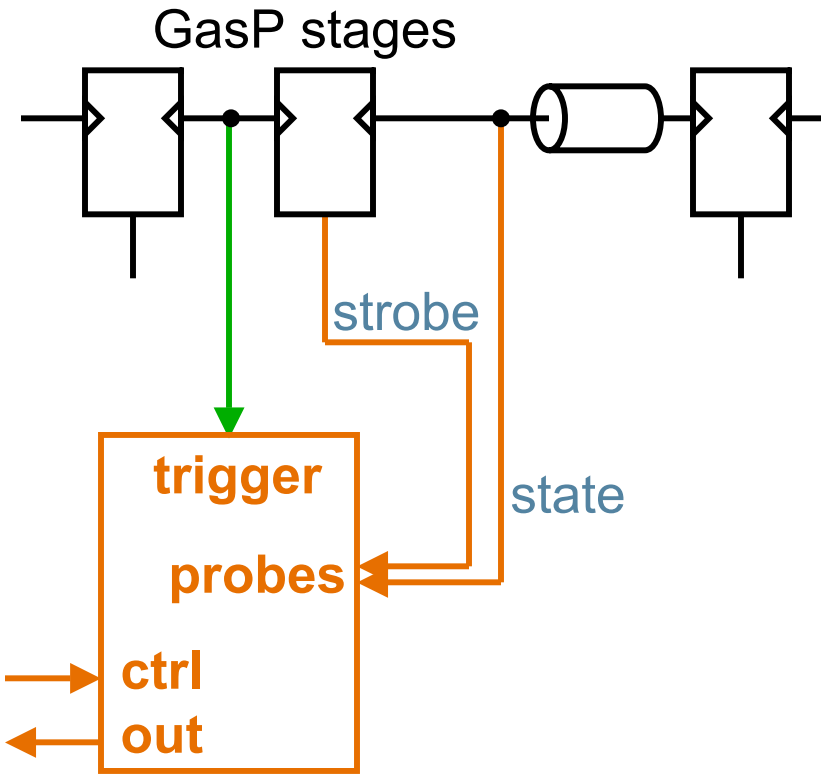
1. Pulldown pulls state wire LOW
2. Far end of long state wire stays HI
3. Nand (high Vth) trips but keeper does not
4. Nand cuts off the pulldown
5. Keeper pulls state wire back HIGH
6. Far end of state wire never went LOW

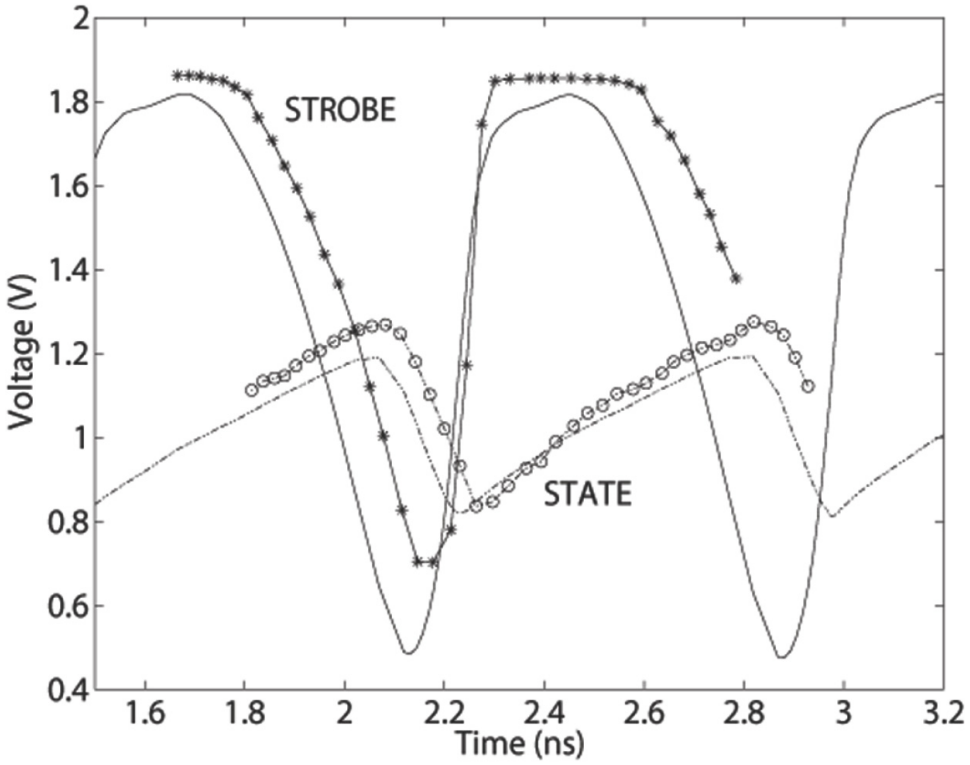
Application: GasP failure mode

- We show a pipeline dropping tokens in simulation
  - Estimated parasitics in a 180nm 1.8V TSMC technology
  - 300μm minimum-width wire between GasP stages
  - Keeper 1/8th the size of the pulldown transistor
  - Logic threshold of keeper and Nand different by 13% Vdd (0.24V)
  - Can arise from slightly greater than 6σ transistor mismatch
- Yes, it is a contrived example
  - But just barely
  - Mismatch gets worse (as fraction of Vdd) with technology scaling

Application: GasP simulation

- Trigger the sampler from prior state wire
  - > Uneven arrival times emulate an asynchronous pipeline
  - > Third GasP stage never sees a token arrive





Summary

- Current test methods inadequate for async circuits
  - Notably for subsampling analog voltages with *in-situ* circuits
- We've studied asynchronously-triggered samplers
  - A relatively simple extension to currently-used sub-samplers
  - Use charge-up timing circuit for triggered variable delays
  - Calibration and external sources reduce (not eliminate) noise
- Simulated two applications to test out the ideas
  - Basic architecture shows promise for further development

On-chip samplers for test and debug of asynchronous circuits

Ron Ho, Ph.D.
ron.ho@sun.com