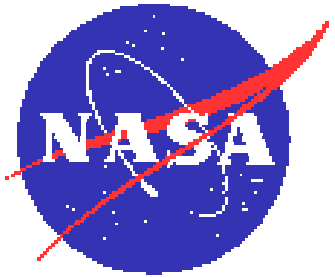
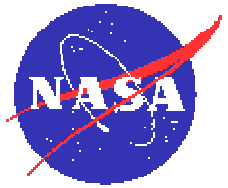


SEE Tolerant Self-Calibrating Simple Fractional-N PLL

Robert L. Shuler, Avionic Systems Division, NASA Johnson Space Center, Houston, TX 77058

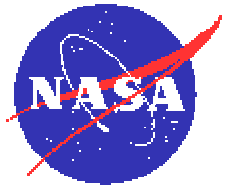
Li Chen, Department of Electrical Engineering, University of Saskatchewan, Saskatoon, SK Canada

SEE Symposium, April, 2010



Importance of PLL Error Tolerance

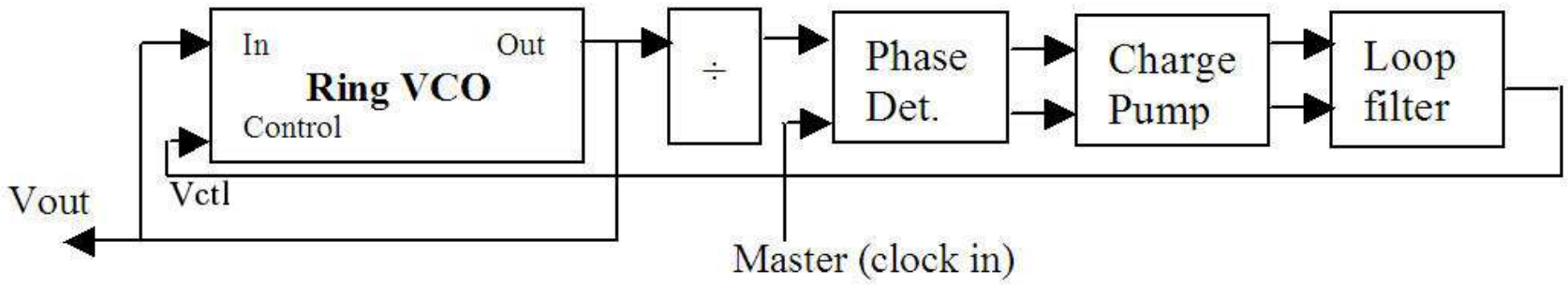
- On-chip clock generation (clock multipliers)
  - Clocks over a few hundred MHz are on-chip generated
  - Phase Locked Loop (PLL) used in clock multiplier configuration
  - PLL errors affect all logic, including redundancy schemes!
- Communication applications
  - PLLs used for channel tuning in many types of devices
  - PLL errors affect bit error rates and overall performance
- Research in Single Event Effects & mitigation
  - Need to evaluate SEE and mitigation at high clock rates
  - Errors from clock can mask or skew results



Background to current effort

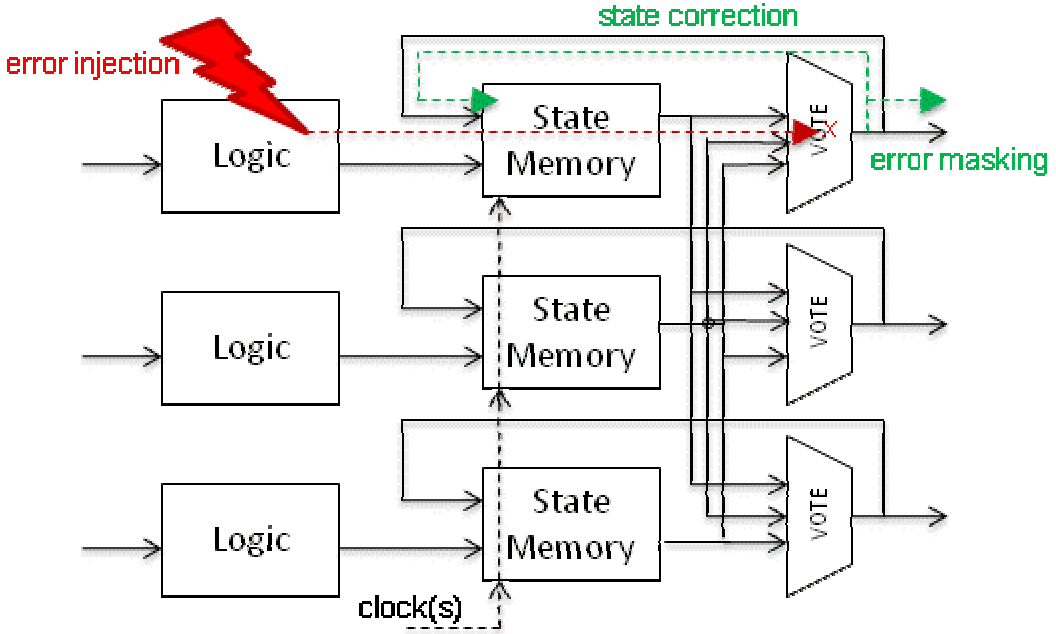
- High reliability clock source needed for SEE testing
- Interest sparked by work of collaborators at U. Saskatchewan and Vanderbilt
  - From a fault-tolerance perspective, a more comprehensive solution seemed feasible and simpler
  - U. Saskatchewan became collaborator on this project
  - Vanderbilt fabricated similar circuit in 40nm SOI, not yet tested
- Acknowledgement of support for this project
  - NASA grants provided heavy ion test funds and .35u fab.
  - U. Saskatchewan provided 90nm fabrication

Typical PLL configuration



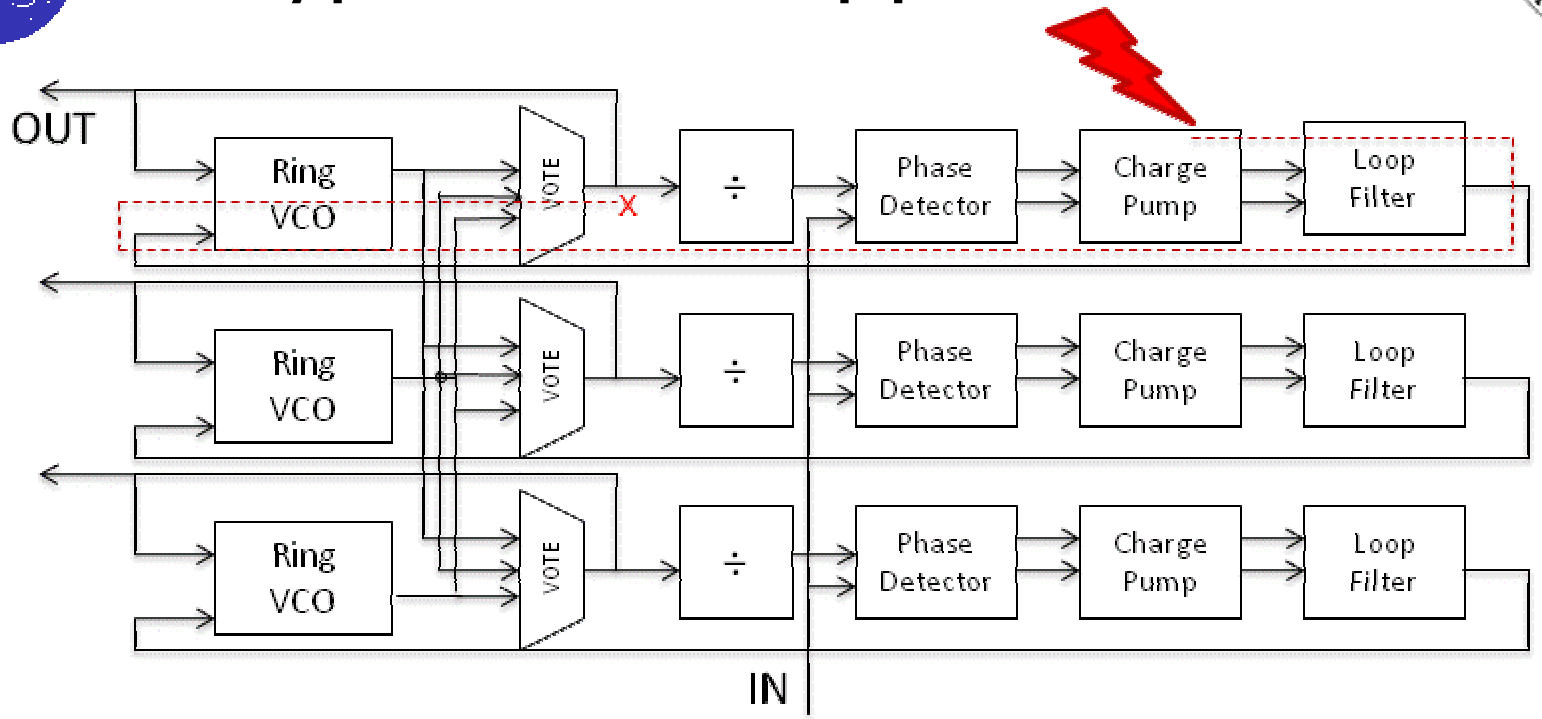
- Circuit characteristics
  - Mixed signal, not all parts are digital
  - Runs at clock-speed and generates the clock
  - State memory and feedback
- Difficulties in SEE mitigation
  - Analog elements are difficult to vote and correct
  - Feedback loops will conflict with voter correction

Typical Voter configuration



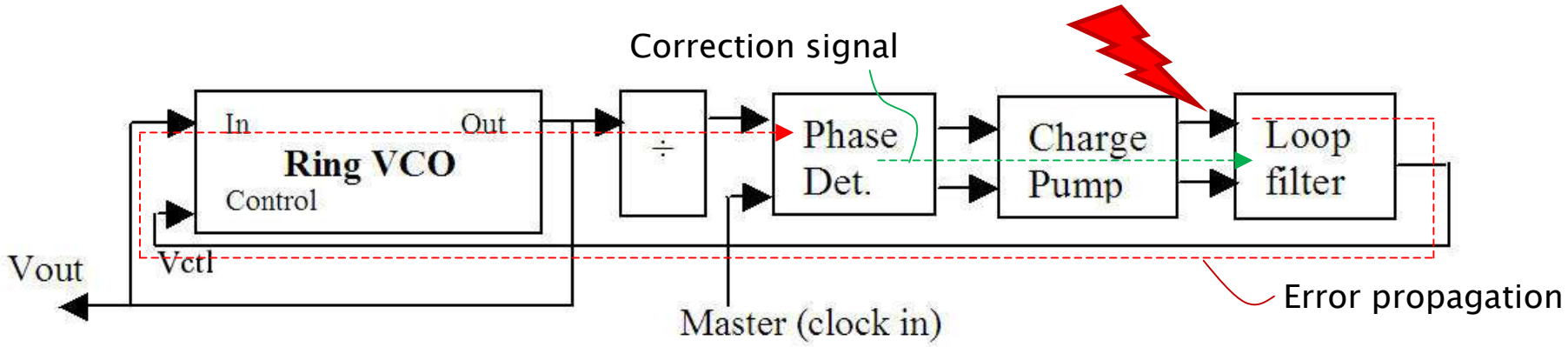
- Circuit characteristics
  - Synchronous voting on clock edge
  - Correction signal feeds back to correct state memory
  - Leaving off the state correction can allow errors to accumulate, resulting eventually in an invalid vote

Typical voter applied to PLL



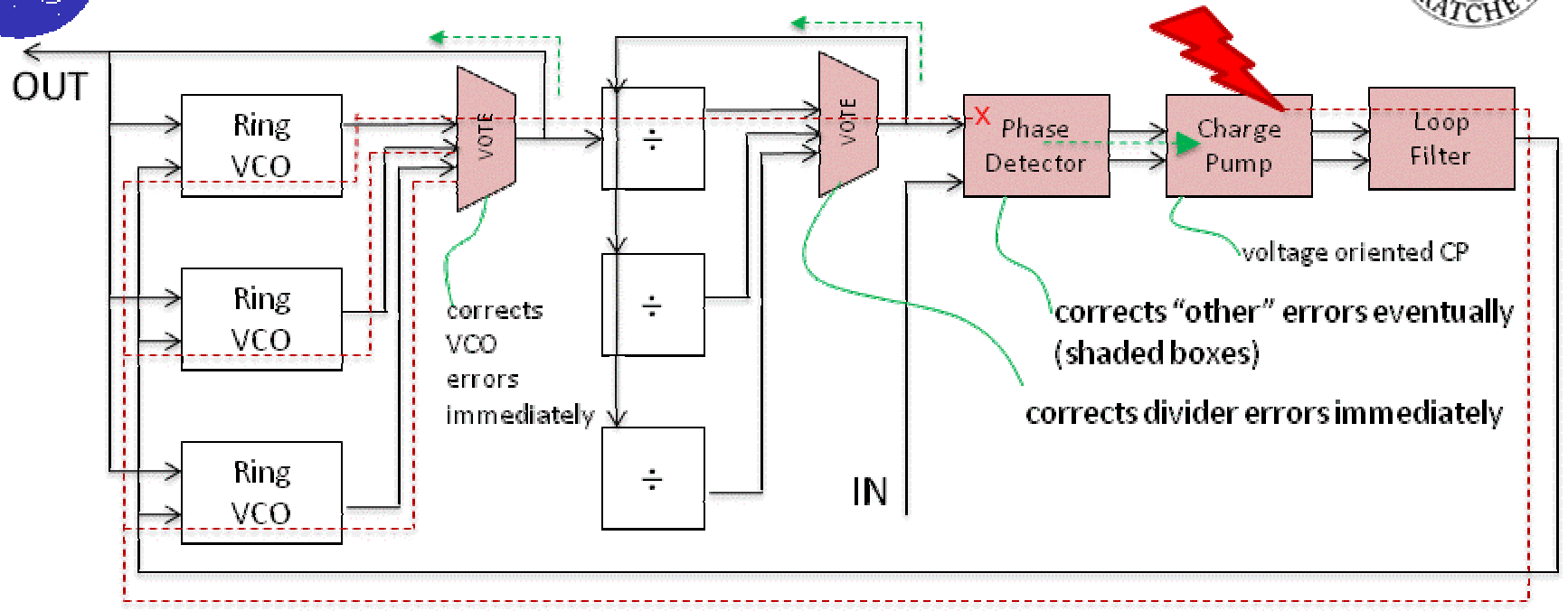
- If output is voted, logical place to feed this back is VCO input
- Error masked by voters, but correction signal also masked!
  - Original error persists indefinitely
  - Reduces redundancy to 2 strings, next error causes fault
  - Voter must not remove error-correction from feedback loop

PLL error response



- PLL is already an error-correcting circuit
  - Phase detector generates correction signal from reference
  - However errors may persist for many cycles
- In theory one could harden all the parts
  - Parts with analog output would require fast analog voting
  - For a PLL charged with generating a high speed clock, many hardening or voting techniques are too slow

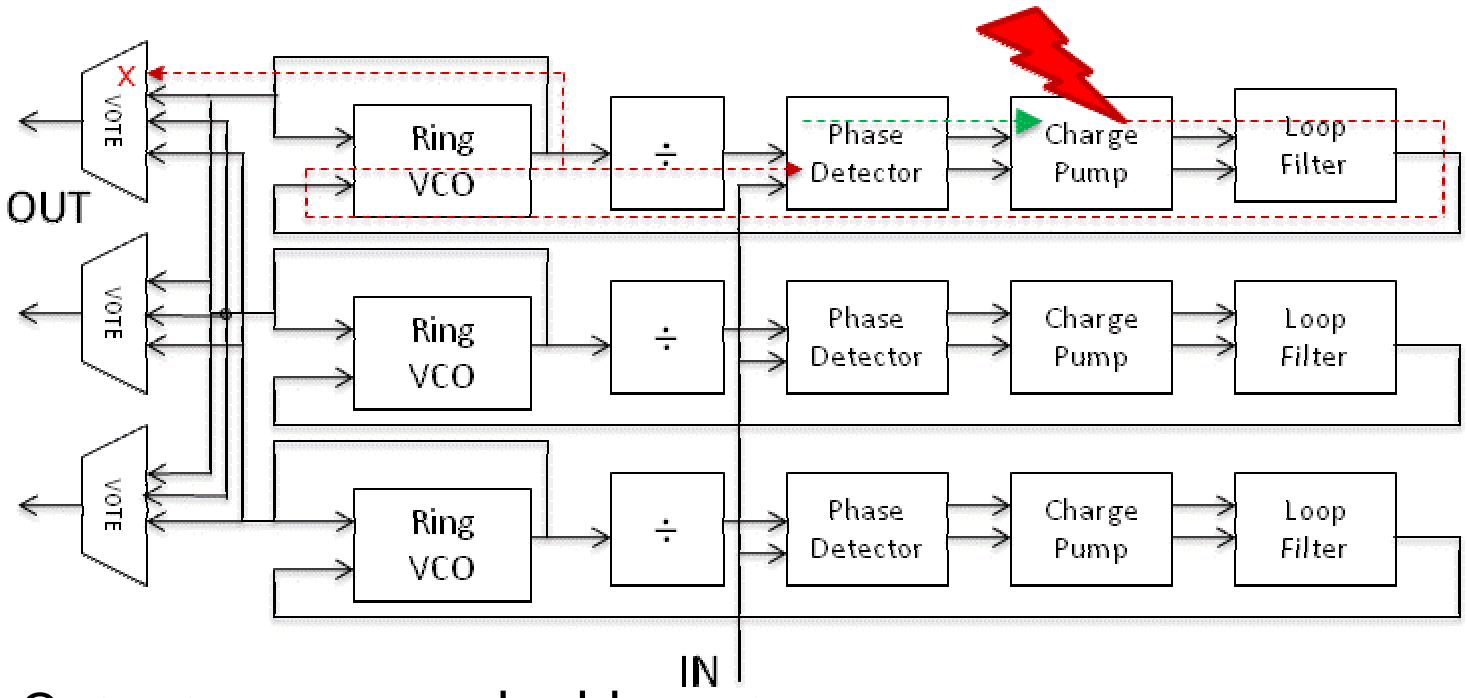
Best practices from recent literature



- VCO and logic vote & correct errors within themselves
  - Single string in rest of PLL prevents correction signal masking problem
- Natural (slow) PLL action corrects other errors (eventually)
- Techniques to reduce vulnerability used if available (e.g. voltage CP)

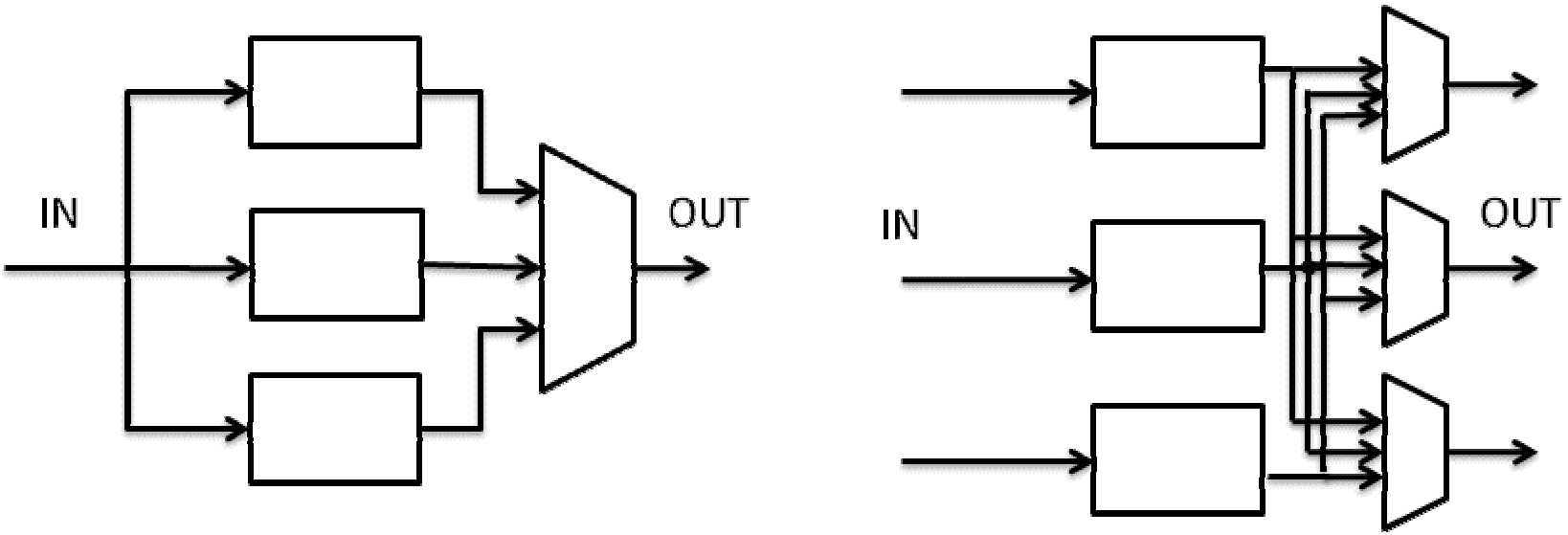
Errors reduced but significant vulnerability remains – what can we do?

Output-only voting



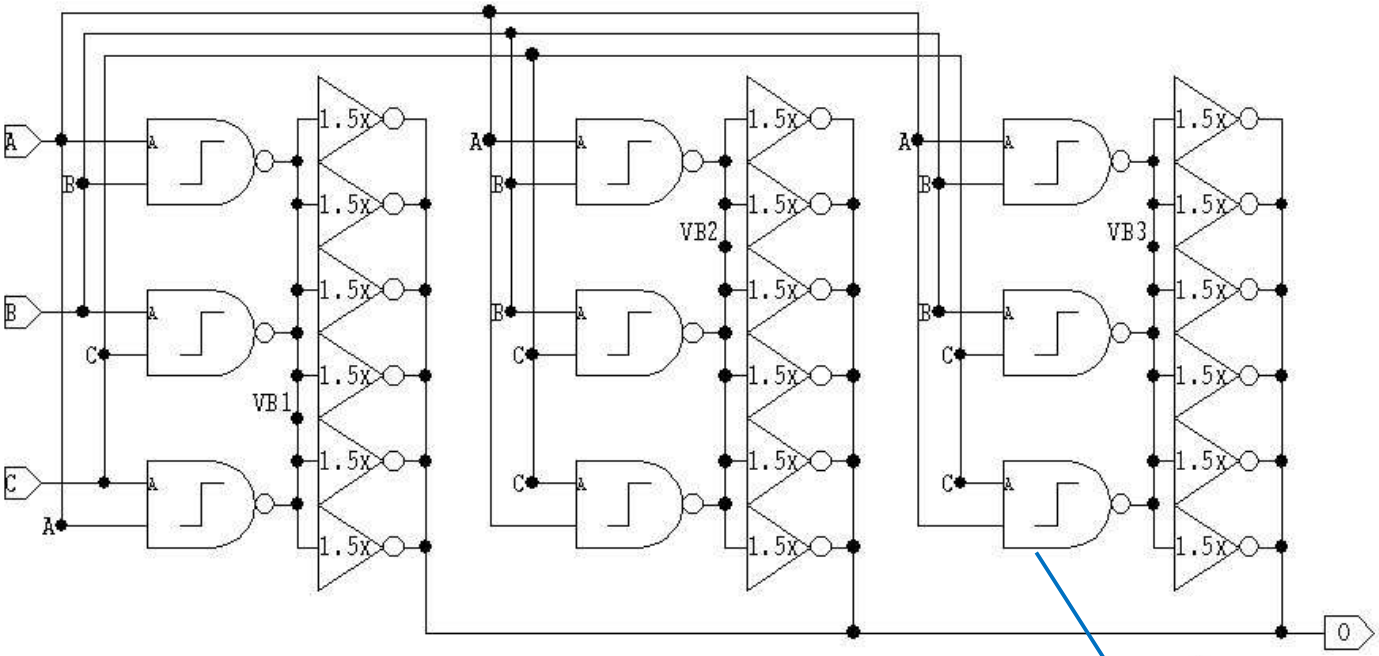
- Output errors masked by voter
- Correction signals remain internally
- Some questions:
  - What if only one output is wanted?
  - Can we vote asynchronous clock signals this way?
  - Will the 3 PLLs always re-sync?

Types of Voters

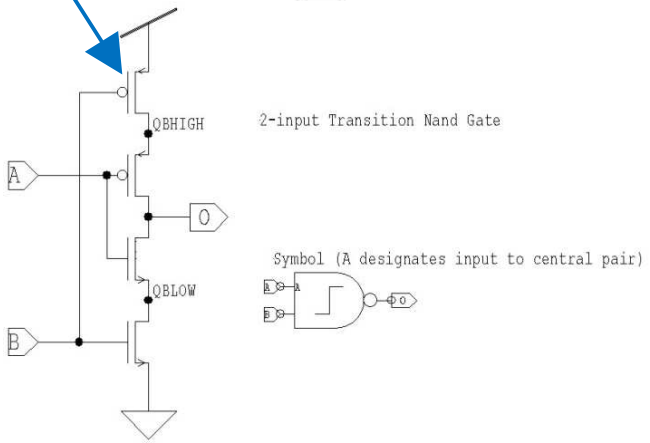


- The single output voter may have a vulnerability at its output
- The multi-output voter is only useful if the client circuit is designed for redundant clocks
- For many applications would prefer a reliable single output

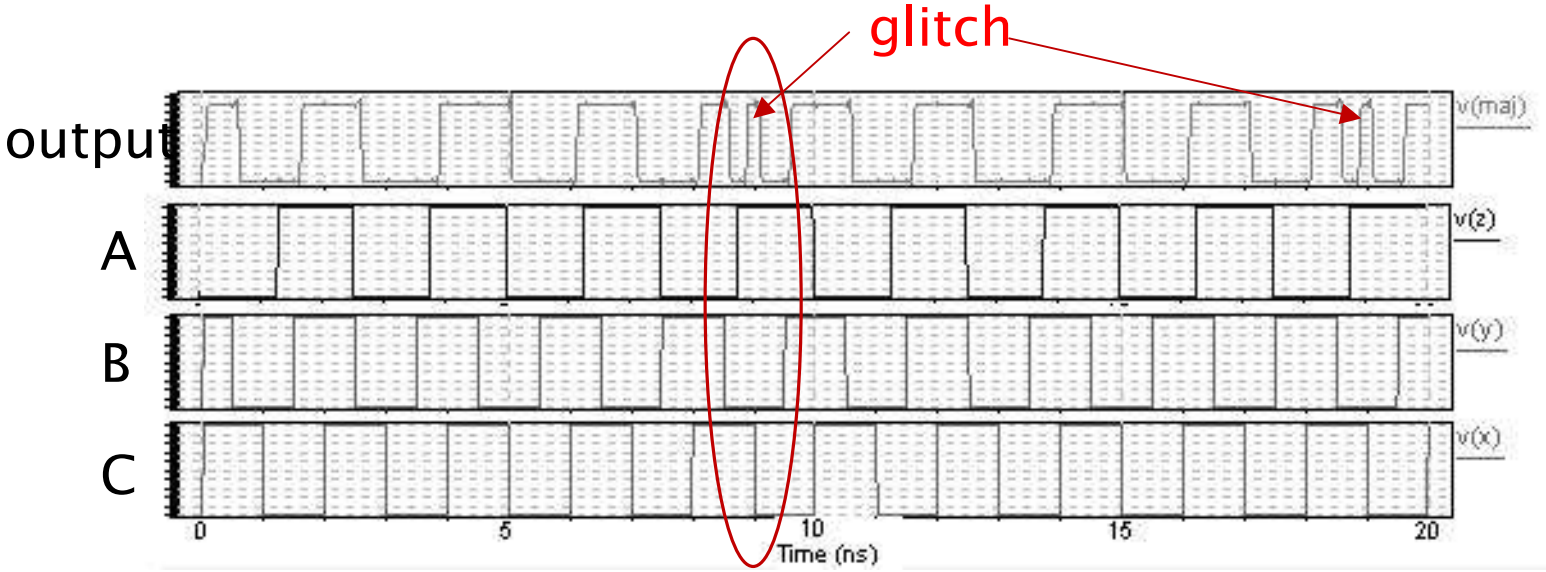
SET resistant single-output voter



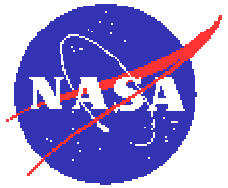
- Single instance high drive capacity circuit
- Not constrained in area like a logic circuit
- Multi-element force voter
- Layout separation to reduce multi-gate SET
- Follow with large buffers in clock tree, or distribute 3 clocks and vote at lowest branches



Voting asynchronous signals (clocks)



- A fast voter can glitch when input phase error exceeds its switching time
  - Example: B & C are correct frequency but there is some phase error
  - When a transition on the incorrect "A" signal occurs while B & C disagree, glitch can occur
- If glitch exceeds logic switching threshold, errors will result
  - I call this PHASE INDUCED VOTING ERROR
- In a locked low-jitter Integer-N PLL, can be fixed by attention to voter speed
- In a Fractional-N PLL, the dividers may not re-sync to the same state!



Re-synchronization

- Integer-N (ordinary) PLLs
  - Normally PLLs are designed for low phase error/jitter
  - Phase offset from device parameter variation must be small
  - A PLL with seek and tracking modes may require tweaking
  - Make sure feedback loop does not have un-responsive states (e.g. false lock at unusual voltages caused by SET)

- Fractional-N PLLs
  - Dividers and delta-sigma logic will not re-sync to same state
  - Spur elimination alone is not sufficient
  - Cycle by cycle phase error must be tolerable to voter
    - Loop filter adequate to remove phase error may be too slow
    - Divider output compensation may be required

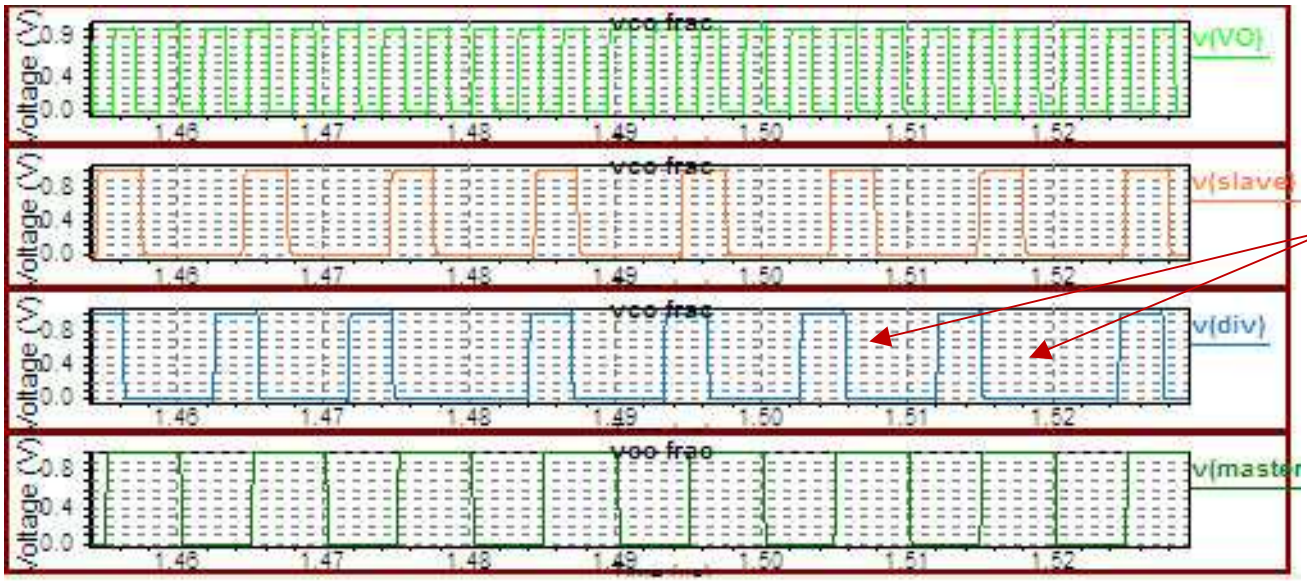
Fractional-N PLL signals

VCO output

compensated divided output

raw fractional divided output

reference

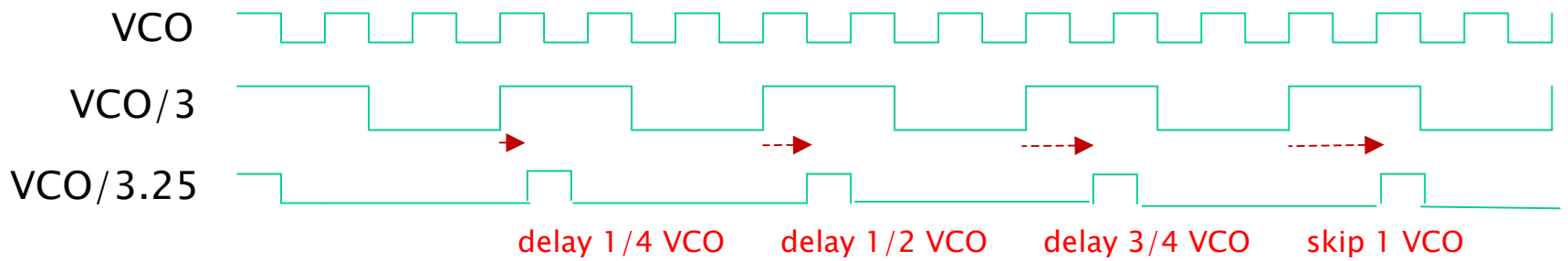


- These devices are needed for clocks (e.g. 3.5x multiplier) and comm.
- Idea is to avoid noise issues with very low reference frequencies
- A fractional-N divider alternates between N and N+1 division
- Delta-Sigma schemes randomize alternation to reduce spurs
- Compensation adjusts the divider output timing to approximate true rate
- Without compensation, loop filter to reduce jitter becomes unrealistic

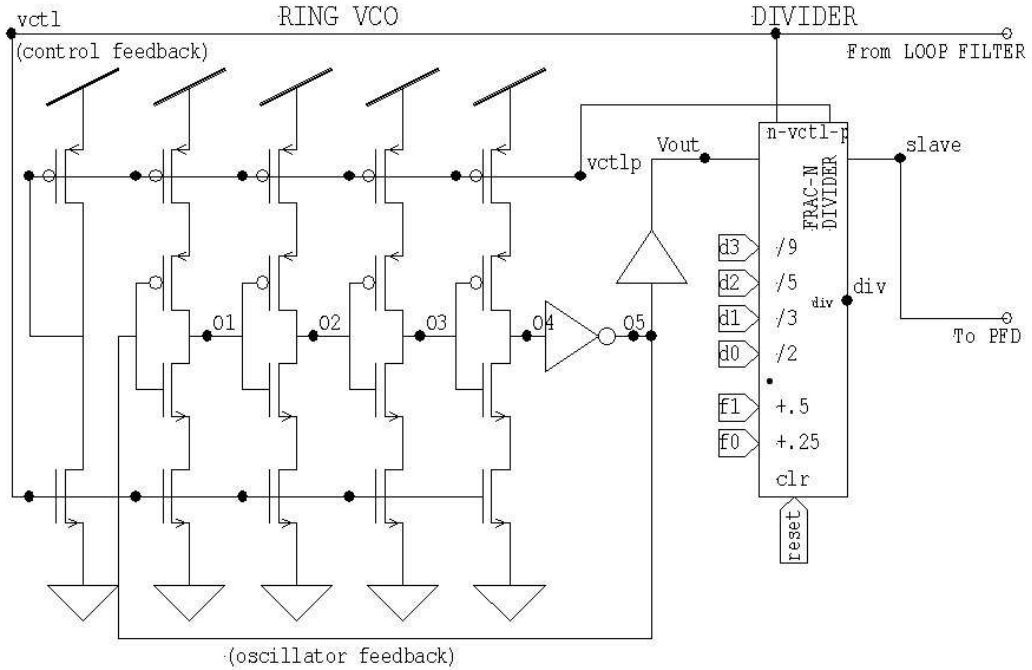
Compensation calibration issues

- Types of compensation
  - Analog compensation adjusts the voltage delta which the phase-frequency detector (PFD) applies to the loop filter
    - Difficult calibration problem, requires temperature compensation
  - Delay compensation directly adjusts the timing of divider output pulses to “true rate”
    - Must be able to calculate accurate fractional cycle delays
  - Both types are somewhat complicated and require calibration

Example of delay compensation:

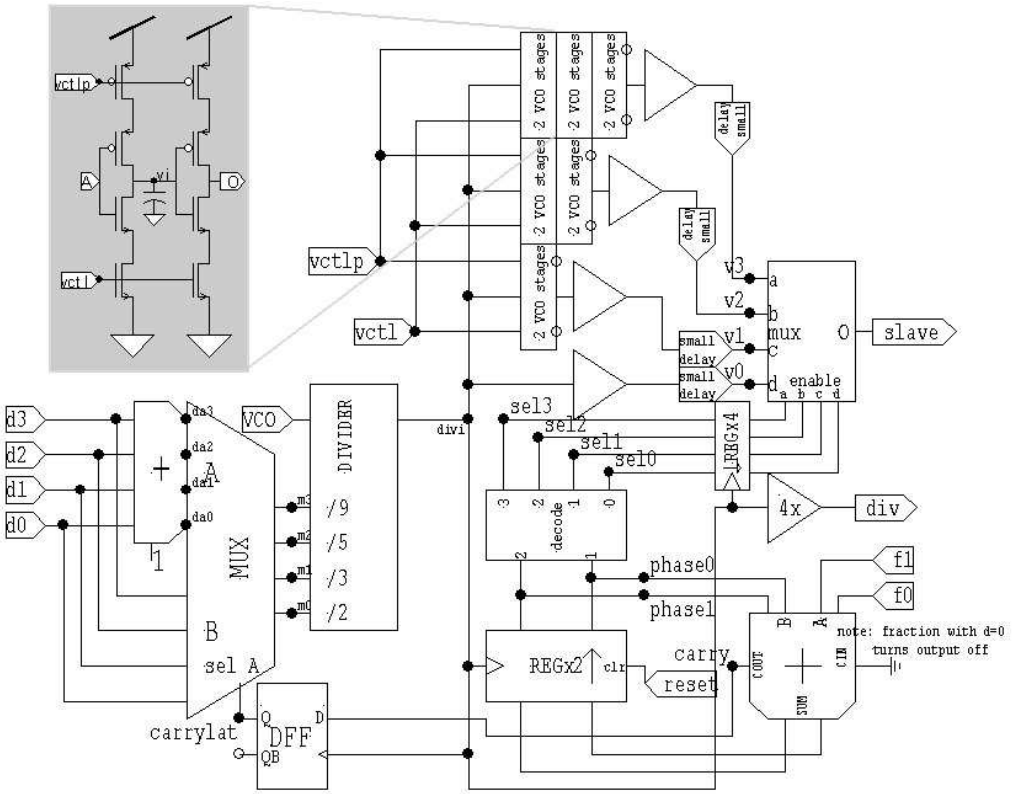


VCO for ¼ fractional divider

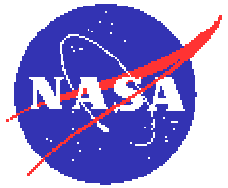


- In a Ring VCO, adjustable delay stages are already present
  - PLL feedback voltage adjusts the delay
  - One pass through VCO delay gives ½ cycle (must be inverting)
- If near-identical delays used for compensation, calibration is done!
  - Some tweaking may be needed if stage loads are not the same (final load)

¼ fractional divider



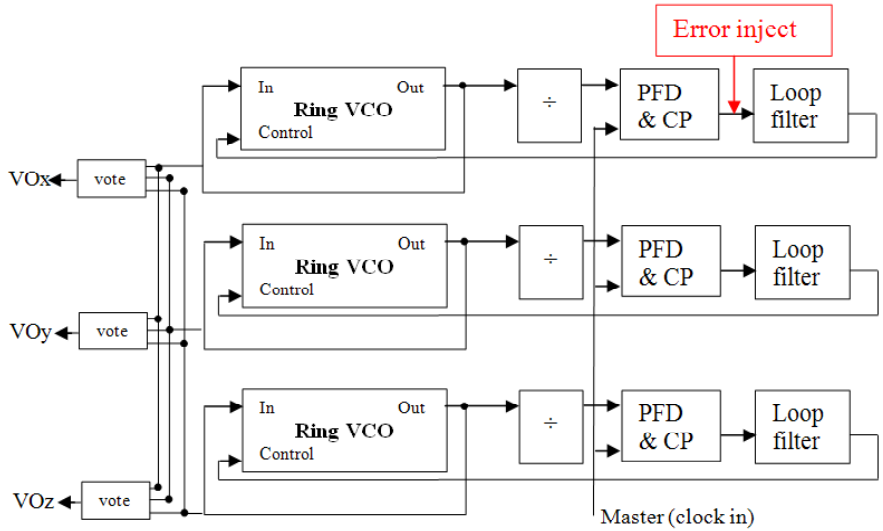
- Alternating N / N+1 divider as usual
- Remainder accumulator selects amount of delay
  - Extra small delay added to allow for computation time
  - M-stage VCO can support 1/2M fraction
  - Stages can be grouped to support lesser fractions
  - Only need enough granularity to avoid voting errors, higher fractions via delta-sigma



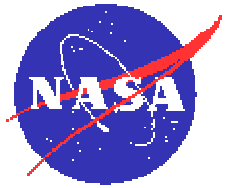
Implementations

- 0.35μm proof of concept PLL (TSMC process)
  - Integer-N
  - 50 MHz reference with up to 8x multipler
  - Bench testing completed
- 90nm proof of concept PLL (STM process)
  - Integer-N
  - 50-100 MHz reference with up to 8x multiplier
    - Basically identical to 0.35um circuit
  - Fabrication complete, test rig development in progress
- 90nm Fractional-N PLL (TSMC process)
  - 50-100 MHz reference with up to 9x multiplier in ¼ increments
  - Capable of 200 KHz channel spacing
  - In fabrication currently

Bench Testing of 0.35µm Part



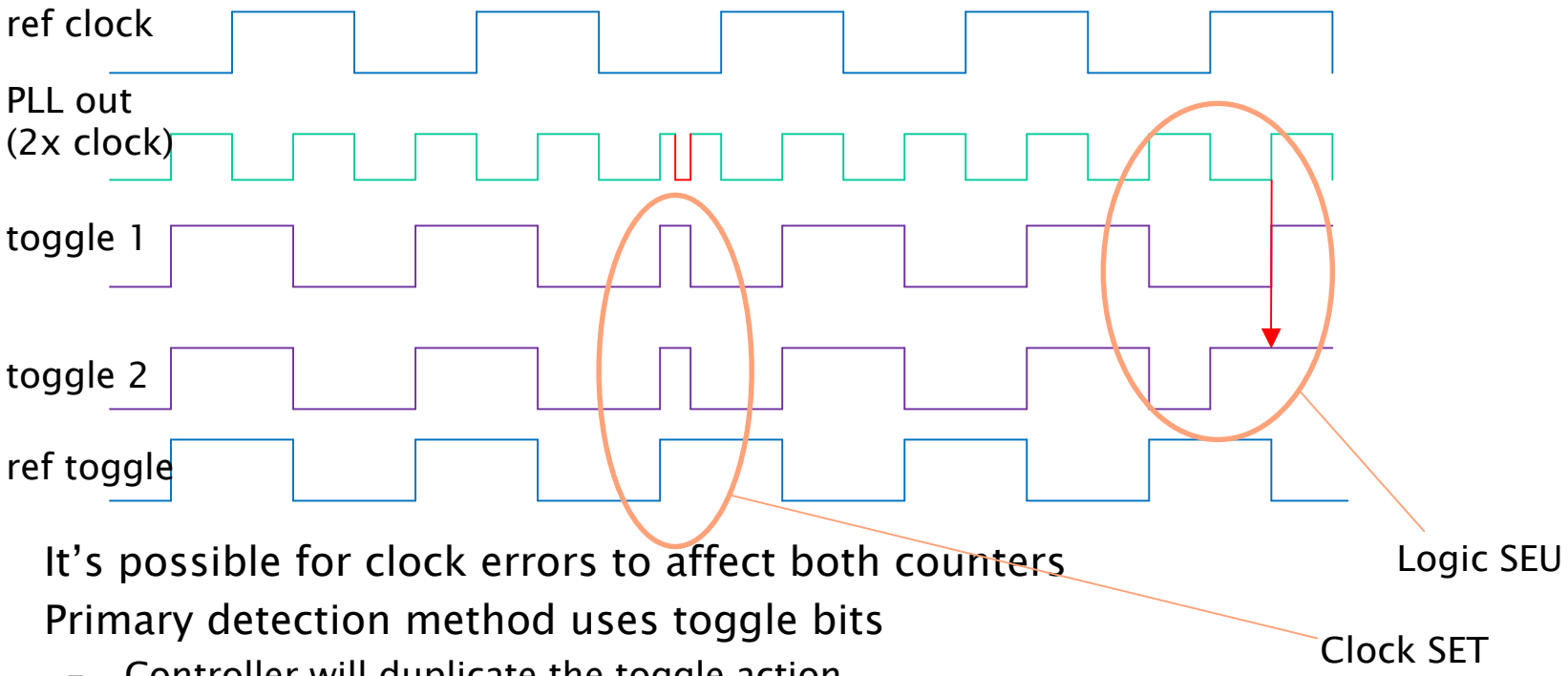
- Error injection was provided for bench testing
  - Error injection is a simulated CP false pulse in one string of arbitrary length
  - Ability to shut down other two strings to verify error injection
  - Laser pulse testing could also be used (possible future test)
- Test results
  - No discernable deviation in output timing after error injection



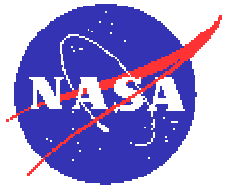
Plans for Heavy Ion Testing

- Testing TSMC 0.35μm and STM 90nm integer-N PLL
- TAMU in May
  - range of angles and ions
  - 1e7 fluence
  - TSMC 0.35mm and STM 90nm integer-N PLL in this test
- PLL provides clock for SEU detection experiments
  - Range of flip flop types & speeds, unprotected and RHBD
  - Pairs of counters with XOR compare and latch error
  - Least significant toggle bits available to FPGA controller

Detection of PLL / clock errors



- It's possible for clock errors to affect both counters
- Primary detection method uses toggle bits
  - Controller will duplicate the toggle action
  - Discrepancy between controller and chip indicates error
  - If simultaneous on several experiments, must be PLL error
  - If SEU error not reported, must be PLL error
- Will use error injection to test error detection
- Expectation is we may see few or no errors



Summary

- A robust method of PLL SEE tolerance shown
  - Not previously evident in literature, though at least one vendor claims an unpublished SEE tolerant PLL
- Fractional-N divider compensation shown
  - Seems to be simpler than previously described methods
- Range of uses
  - Soon to be in use by present authors and Vanderbilt for clock generation for high speed SEE testing
  - Radiation tolerant ASICs, CPUs or FPGAs
  - Small and effective enough for use in high reliability commercial chips (transportation, routing, communication)
- Applicable to most types of PLLs
  - Fractional-N PLL designed and in fabrication