Implementation of ADPLL Networks on FPGAs
Final Presentation

Conor Dooley - 14321611

Supervisors: B. Mulkeen & Dr. E. Blokhina

3 May 2018

Motivation

- Want low power, high frequency clocking system for SoCs.
- Want closely sync'ed clocks, characterised using:
  - Average difference *skew*.
  - Gaussian random process *jitter*.

- Existing solutions limited by power usage at high frequency.
- ADPLL networks can potentially solve this.
- FPGAs can be used to provide a testbed for ADPLL networks.
- **Goals**:
  - Produce such a test platform.
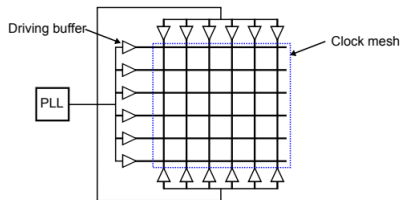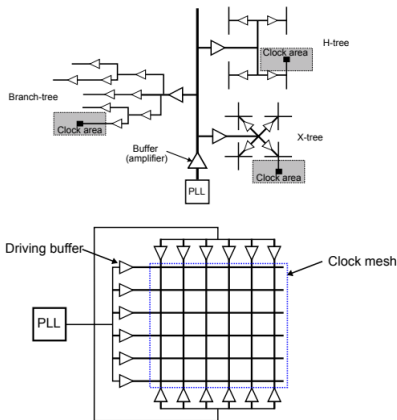 - Investigate its performance.

Existing Solutions

Branch, H, X trees
- Use buffers & delay symmetry.
- Fabrication mismatch problems
  $\rightarrow$ skew, high power usage.

Clock Mesh
- Great timing accuracy.
- Redundancy $\rightarrow$ very high power draw.

Skew Compensation
- Centralised/Decentralised methods.
- Increases power consumption.
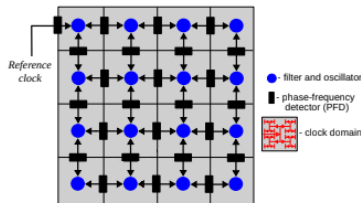

[Zianbetov, 2013]

ADPLL Network

ADPLL network:
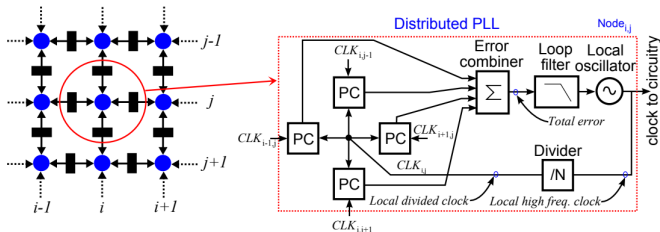
ADPLLs generate clock in an area of chip.

Synced via lower freq. error signal between neighbouring PLLs.

Reduces complexity of synchronisation system.


[Zianbetov, 2013]

Example node:

Why FPGAs?

- Common verification stage for conventional ASIC designs.
    - Hardware validation of digital circuitry.
    - Detection of any potential flaws/errors.
- Zianbetov & Shan - Used FPGA to validate ADPLL network:
    - Unable to replicate mixed-signal blocks.
    - Operating frequency much lower.

- Lose precise control over layout.
- Restriction placed on available hardware.
    - No gates, only LUTs & primitives.

- Limited mixed-signal circuits *are* possible.

- Can examine system performance and dynamics.

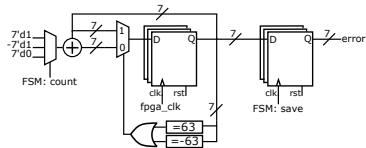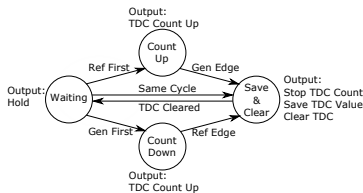- Rapid prototyping, minimal cost.

My ADPLL Implementations
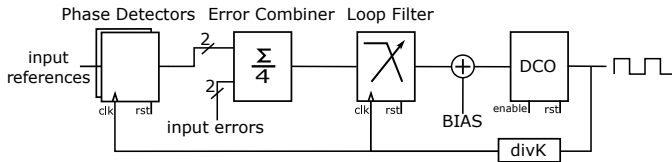
- 3 different ADPLL designs examined.
- 5 MHz centre frequency.
- Range from entirely FPGA clock driven to entirely inverter delay based.
- Number of blocks stay the same between designs:
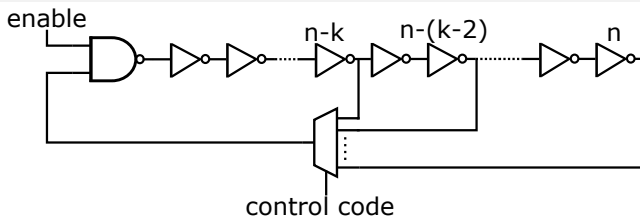 Error Combiner, Loop Filter, Divider.
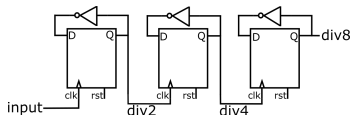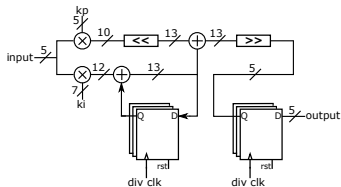
Design 1

- FPGA clocked oscillator and phase detector.

- Oscillator implemented by accumulator.

- Phase detector implemented by a statemachine controlling an up-down counter.

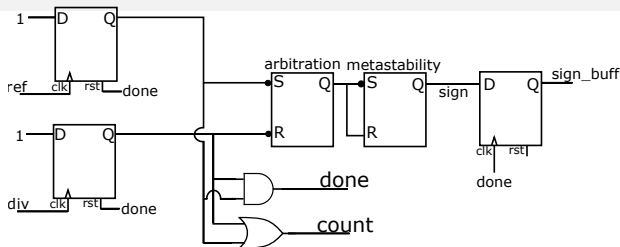- Worst detector/period resolution of the three designs.
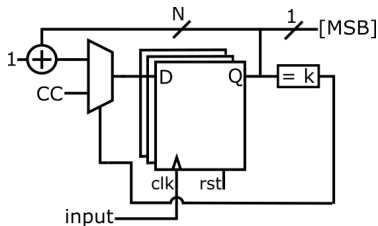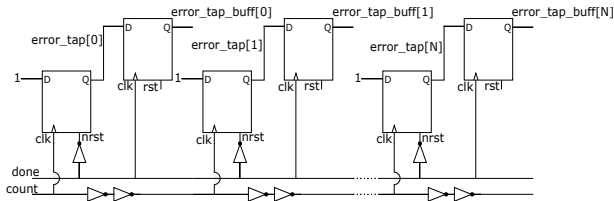
Design 2

- Retains FPGA clocked Phase Detector

- Oscillator replaced by inverter chain using primitives.

- Better approximation of mixed-signal circuits.

- Drawback = loss of control.

- Significant improvement in period resolution,
 3.875 ns $\rightarrow \approx$ 1.1765 ns.
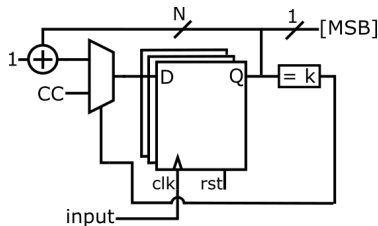
Design 3

- Entirely inverter based, retains RO from ADPLL2.
- SigNum phase detector using inverter primitive TDL.
- Better approximation of mixed-signal circuits.
- Loss of control affects range, not centre freq.
- Detector resolution: 3.875 ns $\to\approx 0.5883$ ns.

ADPLL/Network Performance

- 2x2 & 3x3 implemented with all designs.
- Indep. PLL, uni- & bi-direction mode.
- Compared all three designs in each mode.

- Main comparators: Time Interval Error, C2C jitter & skew.
- ADPLL 3 top performer overall, but greater C2C jitter.
    - C2C jitter $\rightarrowtail$ D1: 1.95 ns, D2: 0.65 ns, D3: 0.75 ns
    - Peak TIE $\rightarrowtail$ D1: 19.7 ns, D2: 19.1 ns, D3: 9.77 ns
- ADPLL 2 has less C2C jitter as PDet has no variability.
- ADPLL design 1 suffers most from significant skew issues.
    Propagation delay due to resolution?
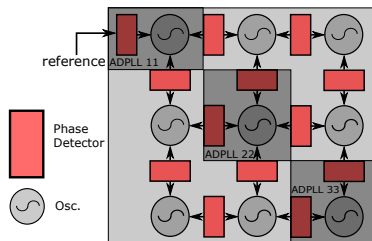
ADPLL/Network Performance
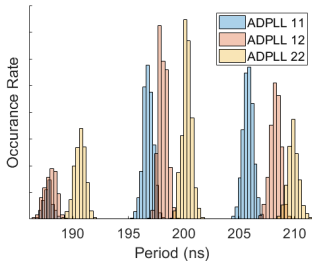
Ring Oscillator
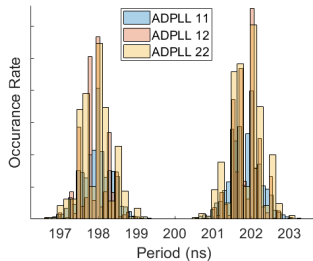
FPGA Clocked Oscillator

– Period distribution highlights variability missing in clocked oscillator and phase detector designs.
– Variability of inverter based designs $\rightarrow$ ideal for behavioural investigations.

     Pierre & Eugene.

– FPGA clocked $\rightarrow$ greaater control, better for design validation.

     Zianbetov & Shan.

Minor Variations

– Investigated a number of minor variations in the design.

– Some to justify decisions:
  - Lack of loop filter input delay.
  - Accumulator Width.

– Also to test for expected results:
  - DCO Width Variation
  - LF Comparison with Eugene's results.
  - Impact of feedback divider.
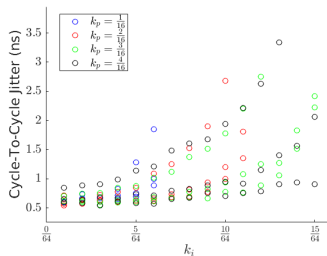
LF Gain Variation
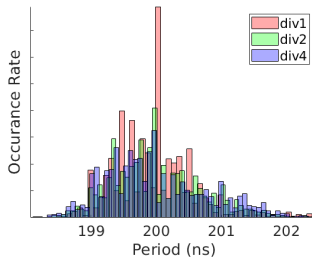
Impact of Divider

Summary

– FPGA based analysis platform for ADPLL network designs.

– Implemented a three ADPLL designs.

– Compared the designs and impact of minor changes.

– Implemented 2x2 and 3x3 ADPLL networks using each of the three designs.

– Proposed suitable use cases for each design.

  – **Future Work**:
    Larger network.
  - TDL characterisation.
  - New FPGA clocked DCO (Period Linear).
  - Procedural network instantiation.