A Multiplying Delay-Locked Loop For A Self-Adjustable Clock Generator

Gary Choi
Borivoje Nikolic, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-108
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-108.html

May 16, 2017

Acknowledgement

Borivoje Nikolic, John Wright, and Benjamin Keller

A Multiplying Delay-Locked Loop For A Self-Adjustable Clock Generator

by Gary Choi

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Borivoje Nikolić
Research Advisor

(Date)

* * * * * * *

Professor Elad Alon
Second Reader

(Date)

Contents

Abstract

Using multiple cores on SoC's are a well accepted solution to improving performance without using as much power as scaling up frequency but each core is a digital circuit that needs to be driven by a clock. Global clock distribution networks are responsible for delivering a reference clock signal to local clock generators that generate local clocks for each core. Current clock distribution and generation circuits tend to consume a lot of power and don't have a set phase relationship, creating a need for synchronizers that inherently have some latency that limits throughput. This work aims to present a self-adjustable clock generator that multiplies the reference frequency to allow for slower references, which reduces power in clock distribution, and periodically injects the reference frequency to better establish a phase relation between the reference and output frequency. The main focus of the design is on a multiplying delay-locked loop whose frequency is managed by a digital control circuit. The circuit generates 16 phases of 2GHz using a 500 MHz injected reference signal. The injection allows the phase error to reset with every injection.

1. Introduction

1.1 Clock Generation in Multi-core SoC's

Power is one of the most important metrics when designing integrated circuits. While improving clock frequency was initially used to improve performance, such a solution eventually hit its limits as problems occurred the higher the frequencies went with power being one of the biggest. The tradeoffs between frequency and power became more unbalanced as a linear increase in frequency resulted in cubic increase in power. Ignoring overhead costs, by having multiple cores the relationship between processing speed and power becomes more linear instead of cubic.

Using multiple cores requires a clock distribution network to provide a clock to the multiple cores. A global clock is typically distributed to different local clock generation blocks that drive the multiple processors [2]. One way to make these distribution networks more power efficient is to lower the global clock frequency. However, to maintain performance, the local clock generation blocks need to generate the same frequency from before lowering the global clock frequency. This requires a clock generator that can multiply a reference frequency.

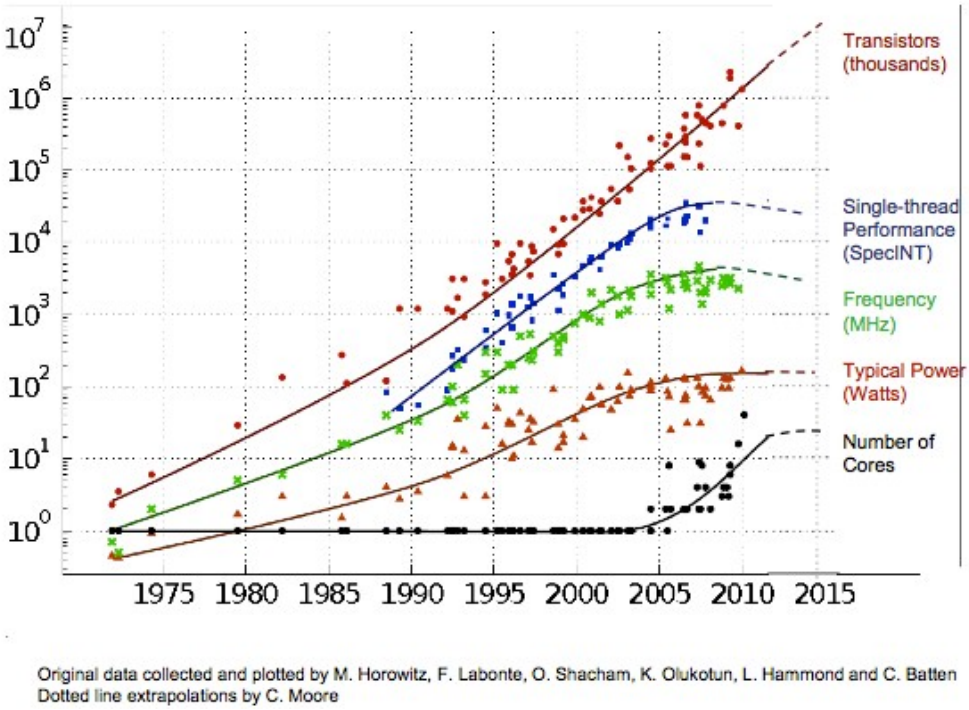

Figure 1: Moore's law indicates a growing number of transistors over past years but frequency has stayed the same [1].

Power efficiency can also be improved using dynamic voltage and frequency scaling in which power and frequency are adjusted based on the operating conditions of the chip. While a power management unit is important in maintaining the supply, an adaptive clock generator is also needed such that the frequency closely follows the voltage disturbances (voltage droop) [3] and no additional power is wasted from excessive frequency while being able to recover frequency to avoid timing violations.

1.2 Synchronization Across Clock Domains

Another concern of this work is synchronization across clock domains. Clock generators that use digital control take several clock cycles to calculate their new target frequencies [4][5] resulting in longer clock source latencies. This makes it harder to establish any phase relationship between two different clock domains, causing unguarded communication between them to be vulnerable to issues such as metastability. These metastability issues can be circumvented using circuits like asynchronous FIFO's but they tend to have a series of cascaded flip-flops to resolve metastability that results in latency.

Injection-locking clock generators inject the reference signal directly into their oscillator circuits and generate their output clock signal from the oscillator, resulting in low clock source latency. However, some current approaches to injection-locking circuits do not have a control circuit that self-adjusts their frequency if the reference frequency jumps around [6][7]. Another circuit uses fractional multiplicity that makes it harder to establish a phase relationship because of a non-integer multiplicity and uses additional area to for a DAC and fractional samplers for its control circuit, increasing the complexity of the circuit [8]. Other circuits may use additional resources such as another oscillator circuit in order to control adjust their frequency [9].

1.3 Proposed Circuit

This paper aims to present the self-adjusting clock generator that adapts to rapidly varying supply voltage [3] with one of its components replaced by a multiplying delay-locked loop. The multiplying delay-locked loop allows for use of a slower reference (500MHz) while maintaining the original desired frequency (2GHz). It injects the reference signal every time it sees a rising edge from the reference, and has a self-adjusting feedback control mechanism to adjust its frequency while using a simple digital control to turn on phase detection.

2. Clock Generation Overview

2.1 Phase-Locked Loops and Delay-Locked Loops

Two common circuits for generating a clock signal are phase-locked loops and delay-locked loops. Most current clock generators use one of these circuits or a variation of one.

A phase-locked loop typically consists of a phase detector, a loop filter, voltage-controlled oscillator (VCO), and sometimes a frequency divider. Its operation can be described as follows:

1. The VCO is responsible for generating an output clock signal. The frequency of the signal is controlled by an input voltage or current level such as adjusting the rail voltages of a ring oscillator.
2. The phase of the oscillator's output is compared to the phase of a reference signal. The phase detector is responsible for finding the phase error between these two signals.
3. This error signal is converted into a control signal and fed through a loop filter. The loop filter is responsible for filtering out unwanted high frequency components that may appear in the circuit.
4. After being filtered the control signal is sent back to the VCO. The VCO's frequency is adjusted which in turn also adjusts its phase. The phase error is adjusted until it approaches zero and the system reaches equilibrium.
5. A frequency divider is sometimes added in between the phase detector and output of the oscillator or between the input reference and phase detector. The phase detector detects an error using a slower version of the reference or VCO output and adjusts the control signal using these divided frequencies.

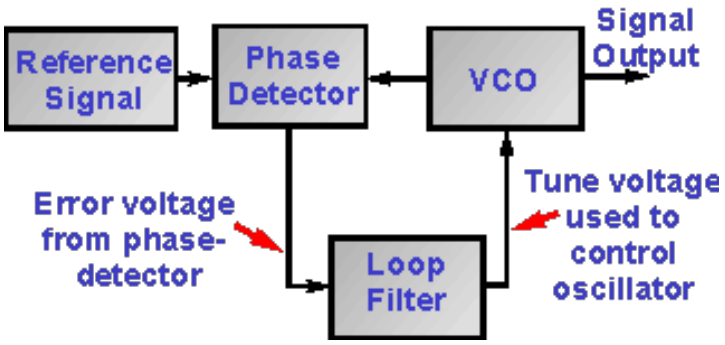

Figure 2: Basic structure of phase-locked loop [10].

A delay-locked loop is very similar in structure except instead of a VCO, a voltage-controlled delay line is used instead. The delay line usually consists of a chain of inverting gates, and phases are compared between the first and last stage of the chain.

The clock generator that this work is based on originally consists of a delay-locked loop, tunable replica circuits, and a digital controller. The tunable replica circuit and digital controller used phases from the delay-locked loop for a frequency scaling technique. The structure of the clock generator will be discussed in more detail.

2.2 Clock Distribution

Most digital circuits use registers that are updated synchronously with a clock signal. Getting the clock signals to these registers across the whole chip is accomplished by using clock distribution networks. These networks typically arrange a cascading tree of buffers throughout the chip with the goal of optimizing skew and wiring resources. Some arrangements include spine, where the global clock is distributed along is a central wire, and H-tree, where wires are hooked up similar to a fractal tree pattern which minimizes wiring resources but tends to have bad skew.

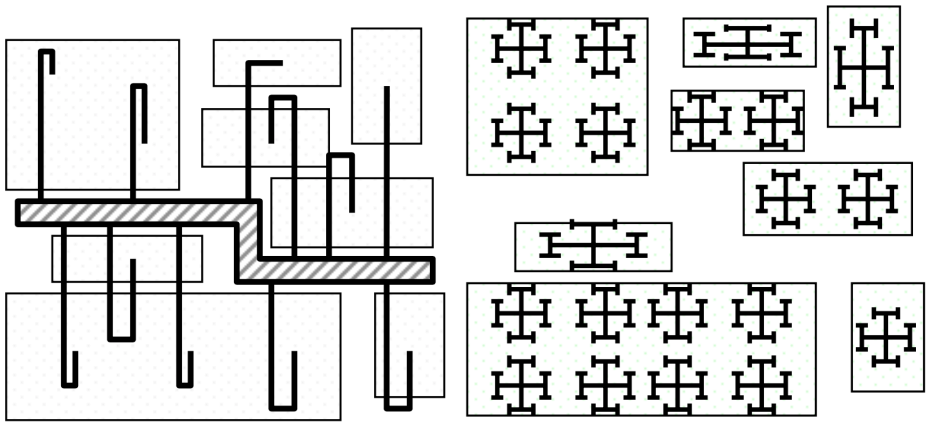

Figure 3: Example of spine (left) and H-tree (right) clock distribution networks [11].

Each distribution network inherently comes with a delay, or clock latency, needed to deliver the source clock to a register usually stemming from the delay going through a network of buffers. Because of variations and non-idealities in a clock tree, differences in latency for each register results in skew. Skew can cause setup and hold time violations and metastability if not accounted for. Across different clock domains, this skew is made even worse, resulting in the need for synchronizers or deskewing.

2.5 Synchronizer

The most basic example of a synchronizer consists of two flip-flops in series under the same clock. Asynchronous data is fed into the first flip-flop. Setup or hold time violations from the asynchronous data can cause the flip-flop to enter a metastable state. However, the probability of it staying in that state decreases as time passes. As

a result, by the time the next clock edge reaches the second flip-flop, since it is synchronous with the first flip-flop, it should receive valid data as opposed to a metastable signal. More robust synchronizers will use more than two flip-flops to further minimize the chance of metastability occurring.

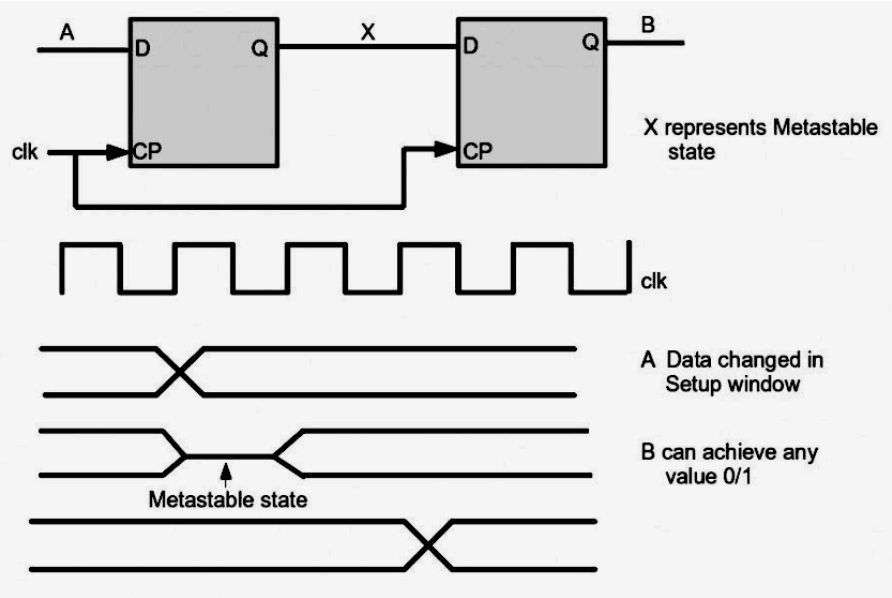

Figure 4: Example of synchronization circuit and how it resolves metastability [12].

2.4 Evaluating a Clock Generator

Important metrics for clock generation circuits are the locking range, its output frequency, and jitter. Some more general considerations would include power and area.

The locking range is the range of frequencies of the input reference signal that the circuit can track and remain locked; any frequency outside this range can cause the circuit to not behave as intended such as never reaching a consistent frequency. The output frequency is crucial in deciding how fast the rest of the circuit can operate.

Jitter is any deviation from the true periodicity of a clock signal and needs to be accounted for when meeting timing requirements.

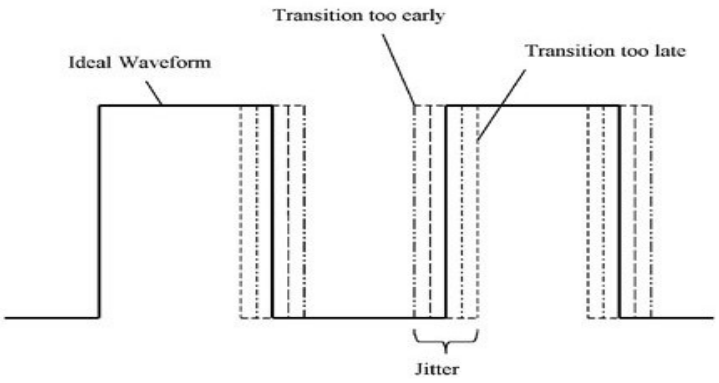

Figure 5: An illustration of jitter.

3. Clock Generator

3.1 Overview

The clock generator [3] adapts frequency based on the supply voltage. Its main components can be divided as the controller, tunable replicable circuits, and delay-L loop.

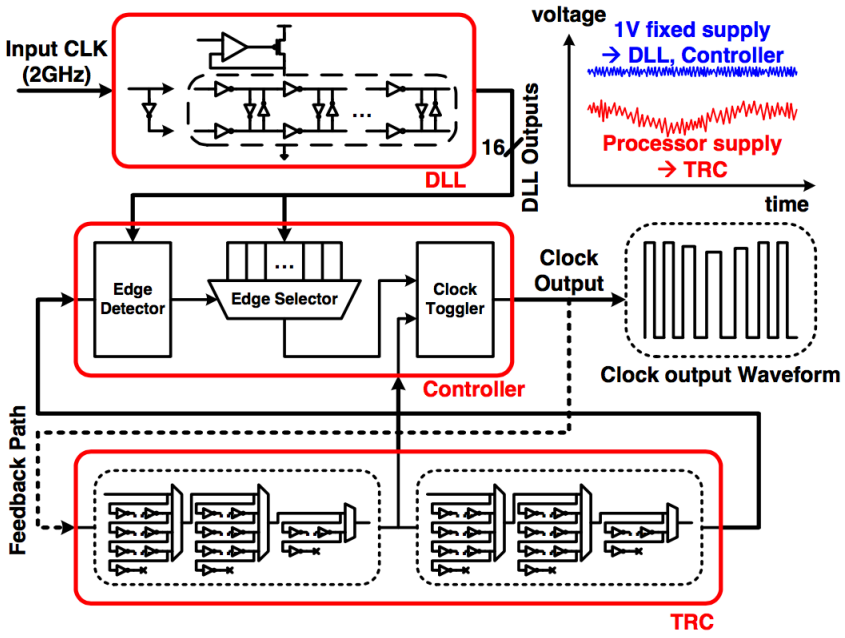

Figure 6: An overview of the clock generator's components and how they are connected [3].

3.2 Multiplying Delay-Locked Loop

3.2.1 Overview

The original delay-locked loop in [3] follows the general structure of a delay-locked loop mentioned earlier. The delay line is implemented using 18 stages of inverters. The phases of the first and last stage are compared using a phase detector.

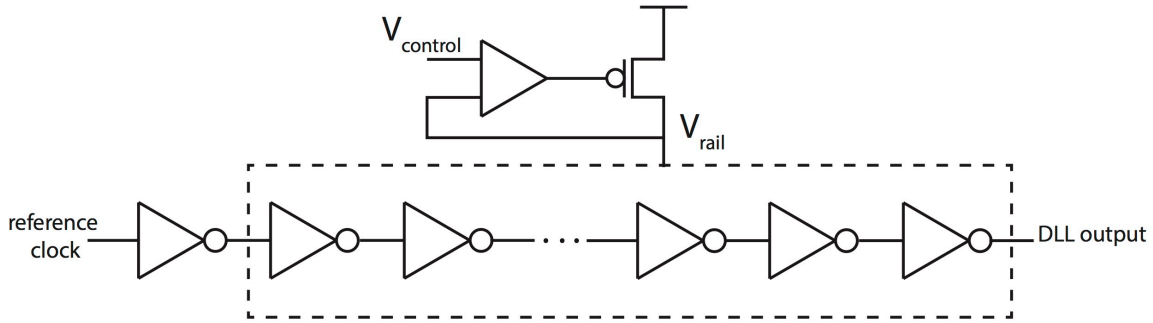

Figure 7: The original voltage controlled delay line. The low dropout regulator adjusts the delay of each stage.

The phase detection circuit outputs an UP or DN signal that is fed into a charge pump. An UP signal tells the charge pump to increase its output voltage while the DN signal tells it to lower its output voltage. The output of the charge pump is fed into a filter that consists of some capacitance and the output of the filter is used in a low-dropout regulator (LDO) circuit. The LDO controls the rail voltage of the delay line and adjusts the delay of each stage until the first and last stages' phases match and there are 16 equally spaced phases between the two.

In order to be able to use a lower reference frequency, we replaced the original delay-locked loop with a multiplying delay-locked loop. Its components are somewhat similar but the difference in generation is substantial. The reference signal is injected periodically, indicating that the reference frequency and oscillation frequency do not be the same, similar to a phase-locked loop with a frequency divider. This allows us to use a slower reference clock while still generating the same frequency as the original delay-locked loop and also resets the phase error with each rising edge of the reference signal.

The structure of the circuit is shown in Figure 10. The ring oscillator is responsible for generating the output frequencies and one of them is fed to the phase detector to be compared to the reference input signal for a phase error. The charge pump and loop filter convert the error into a control voltage used by the LDO. The LDO controls the frequency of the ring oscillator. The control logic is responsible for managing the periodic injection of the reference signal and disabling the phase detector when it shouldn't be looking for phase error.

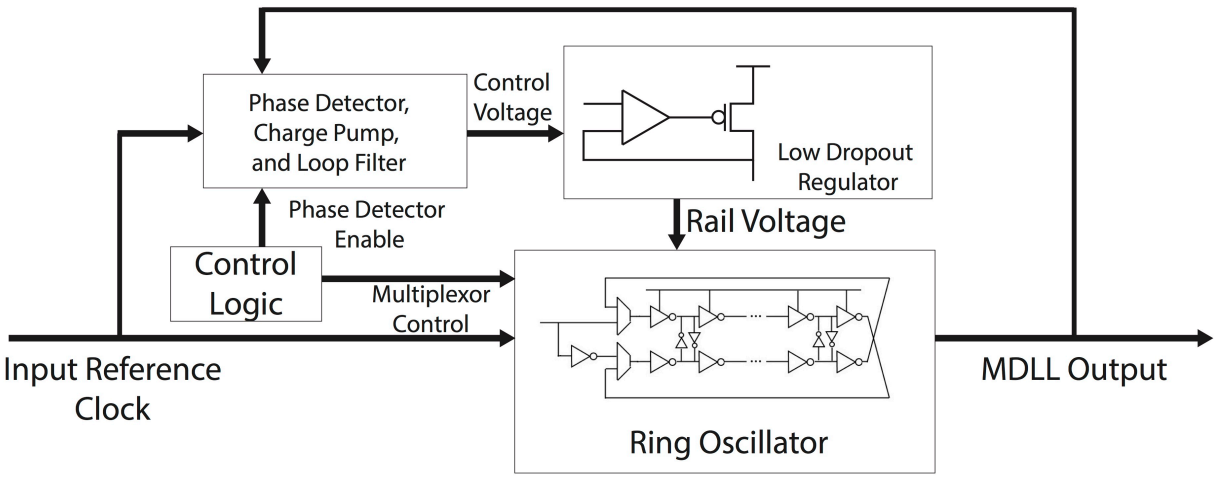

Figure 8: Overview of the multiplying delay-locked loop.

3.2.2 Low-dropout Regulator, Charge Pump, and Loop Filter

Some of the multiplying delay locked loop's components accomplished the same task in equilibrium as they did in the delay-locked loop so they remain the same or had little change. The LDO remained the same since the voltage input and output ranges and frequencies were similar to the Cassia's original delay-locked loop's case.

The charge pump still uses UP and DN signal to adjust its output voltage but the slew rate was slightly adjusted to better account for the new architecture and the loop filter's capacitance was adjust accordingly to ensure the system was still stable after the changes to the charge pump.

3.2.3 Delay Loop

To maintain the original architecture, the multiplying delay-locked loop is intended to replace the delay-locked loop so it needs to generate 16 equally spaced phases of whose frequency is equal to the original's, which is 2GHz. A 500MHz reference signal was desired so the multiplying delay-locked loop needed to oscillate four times faster than the reference signal. Eight stages of cross-coupled inverters arranged into a differential ring oscillator were used to generate the necessary phases, with two from each stage.

The reference signal is injected into the loop using multiplexers. The multiplexers choose between completing the loop or injecting the reference signal. Ideally, after each injection into the first stage, the injected stage will experience three other rising edges before the next injection, effectively resulting in a frequency four times faster than the reference. A controller circuit is responsible for making sure the multiplying delay-locked loop reaches this equilibrium state.

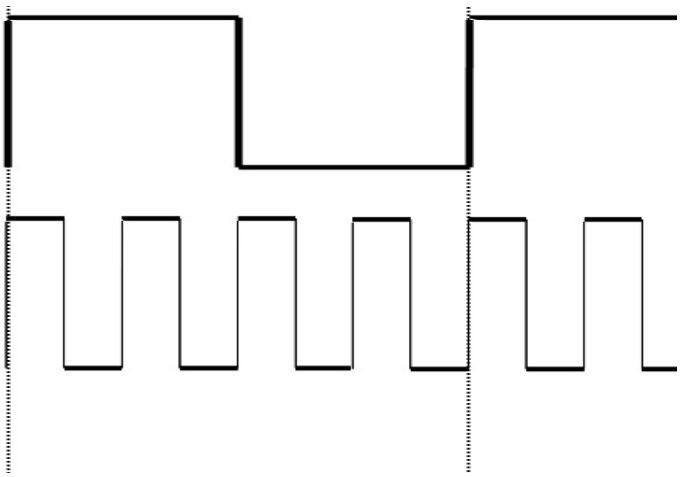

Figure 9: The ideal waveform of the multiplying delay-locked loop. The oscillator's frequency (bottom) should be four times faster than the reference's (top).

Two multiplexers were used on the differential pair of one stage. If only one multiplexer was used to inject the reference into only half of one pair, whenever the input reference experience a sudden frequency jump, only one of the input pair would experience this shift and the other of the pair would have an incorrect phase.

The multiplexer's control signal is driven by a pulse generator. The pulse generator outputs a pulse with every rising edge of the input reference. The length of the pulse determines how long the reference is injected and consequently affects the duty cycle and jitter of the oscillator.

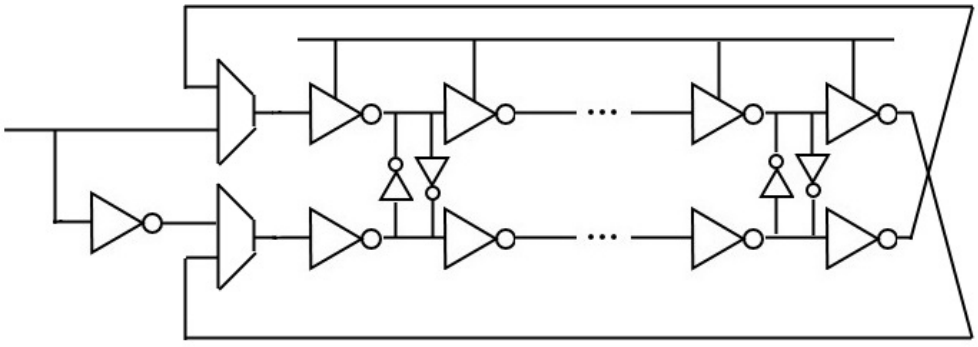

Figure 10: Basic structure of the delay-locked loop.

3.2.4 Multiplying Delay-Locked Loop Controller

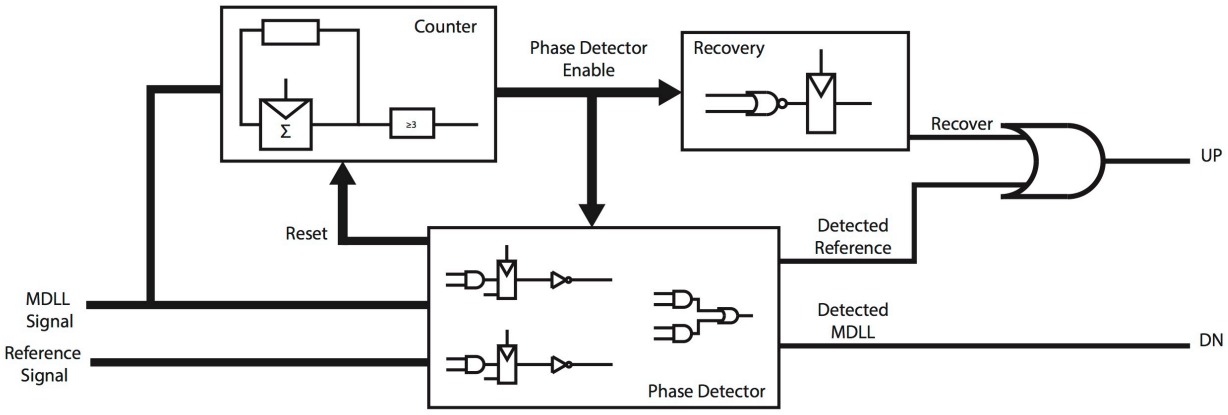

Figure: 11 Overview of the controller circuit.

The controller is responsible for making sure the ring oscillator is at the correct frequency by correctly asserting UP and DN. It uses a counter to decide when to turn the phase detector on and the phase detector determines the UP and DN signal for the charge pump. The recovery circuit also uses the phase detector's enable signal to decide when to assert UP independent of the phase detector.

3.2.4.1 Phase Detector

Since the multiplying delay-locked loop is comparing the phase of the reference signal with the phase of a loop stage that is four times faster, the phase detector needs control to ensure that it detects the phase error between the reference signal and only every fourth edge of the loop.

The phase detector is implemented using a pair of flip-flops and works in conjunction with a counter. The inputs to these flip-flops are gated by an enable signal that only toggles high once the counter has counted the correct number of edges. The specifics of the counter will be discussed in the next section.

The function of the flip-flops is to detect whether the reference signal's rising edge or one of the loop stages' rising edge occurred first. One is triggered by the reference signal rising edge and the other by a loop signal rising edge. The input to each flip-flop is a logical high if the counter asserted that the phase detectors are enabled and that the other flip-flop hasn't detected anything. In other words, when enabled, once a flip-flop's respective clock triggers it, it toggles high and the other flip-flop cannot toggle high until both flip-flops are reset. For this circuit, detecting the reference first is an equivalent of asserting UP on the charge pump and detecting the loop signal first is an equivalent of asserting DN.

Detecting both a reference rising edge and a loop rising edge resets the flip-flops. This ensures that the phase detector detects the full duration of the phase error. The reset signal goes through a pulse generator because the flip-flops need the reset signal to be asserted a certain length of time.

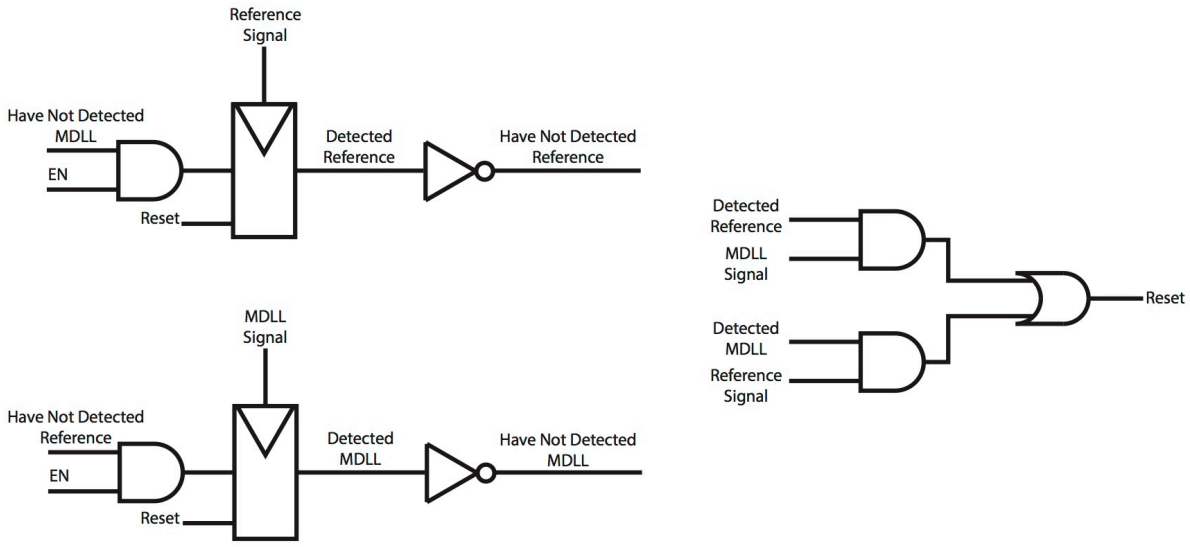

Figure 12: Basic overview of the phase detector. The "Detected Reference" signal is equivalent to a DN signal to the charge pump while the "Detected MDLL" signal is equivalent to a UP.

An example of the phase detector's operation is illustrated in Figure 13.

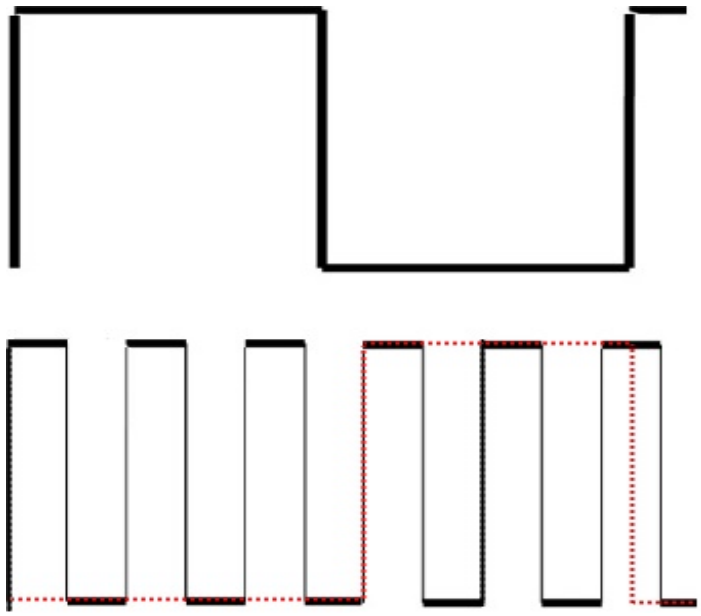

Figure 13: Waveform of when the phase detector is enabled. The top waveform is the reference clock and the bottom is one stage of the ring oscillator. The phase detector being enabled is indicated by the red dotted line.

Once the phase detector is enabled, it waits for the next rising edge from either the reference or loop stage. Then it asserts UP or DN based on whichever it detects first. In Figure 14, this is depicted by the blue waveform.

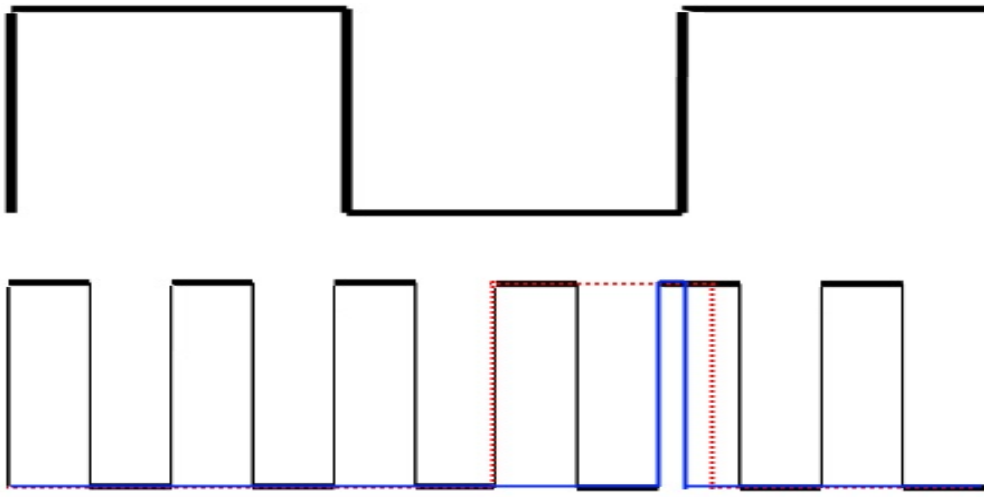

Figure 14: The blue waveform indicates that DN needs to be asserted because the ring oscillator's rising edge came before the reference.

Given that the supply line starts off at 1V, the loops frequency is usually more than four times faster than the reference. This means the DN signal will asserted more

often than the UP signal. The UP signal should rarely need to be asserted using the phase detector once the multiplying delay-locked loop reaches the desired equilibrium state, but is needed to ensure that the loop frequency doesn't keep drifting down.

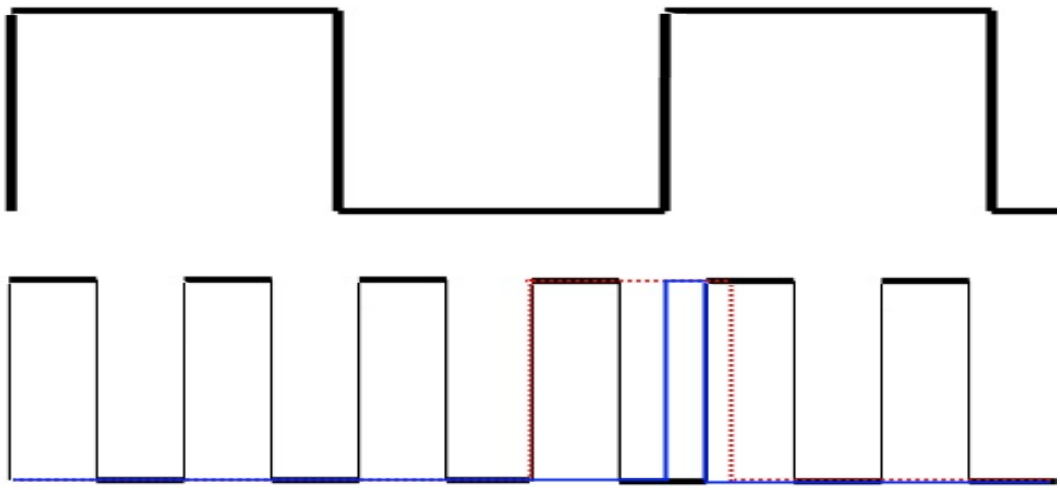

Figure 15: Here the multiplying delay-locked loop's rising edge is a bit late so the UP signal indicated by the blue waveform is asserted.

3.2.4.2 Counter

The controller is responsible for making sure that phase detection occurs only when desired. It accomplishes this by using a counter that increments every time it detects the rising edge of the loop stage and asserting the phase detector's enable only after the counter has counted up to 3. The counter counts up to 3 rather than 4 because the reference edge should ideally count as one of the loop's edges as it is being directly injected into the loop.

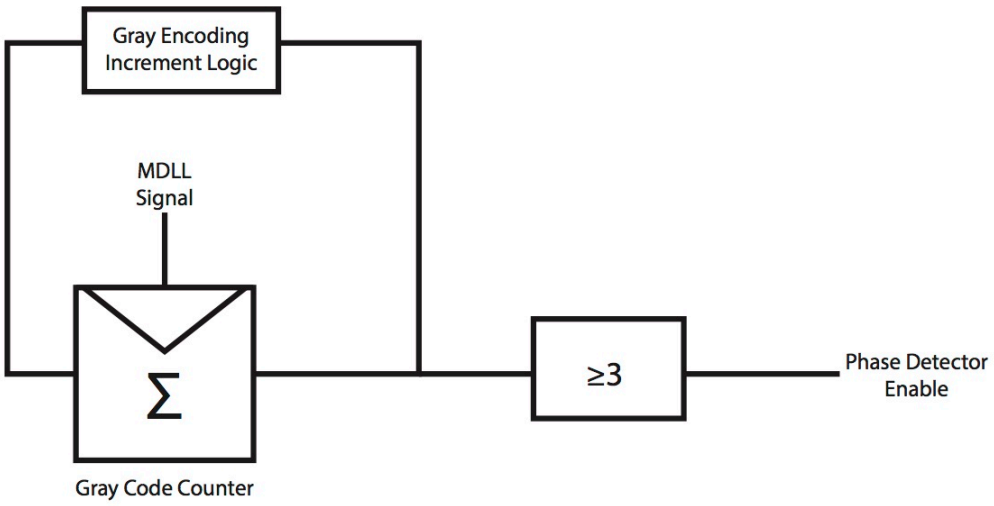

Figure 16: Basic overview of how the phase detector's enable signal is asserted.

Since the greater than or equal to 3 logic is combinational, the controller uses a 4 bit gray encoded counter. Using a normal binary encoding causes a glitch in the phase detector's enable signal. For example, 3 is represented as 011 in binary and 4 is represented as 100 and in either case the enable signal should be high. In the combinational logic for detecting greater than or equal to 3, there would be a brief moment where the enable would toggle low again when switching from 3 to 4 because all three of the lowest bits are all changing. In the worse cause, this can cause the phase detector to miss an edge.

3.2.4.3 Recovery

The controller is also responsible for one last task, which is recovering the signal from false locking. One of the main issues with the controller without this recovery mechanism is that if the rail voltage of the loop drops too low, then the multiplying delay-locked loop will reach a state where the loop will only have 3 rising edges to each reference signal rising edge. Since the combinational logic expects the ratio to be 4 before the phase detector's enable signal toggles high, none of the circuit's feedback will occur and the circuit will be stuck in a state where it can't lock properly.

With the current charge pump and loop filter, this doesn't happen nominally if the supply starts at 1V. The control voltage can't drop fast enough for the loop's frequency to go from 4 times or higher to 3. However, if the we experience sudden low to high frequency shifts as shown in Figure 18, then recovery is needed since the loops frequency starts off lower than 3 times the reference when shifting into a higher reference frequency.

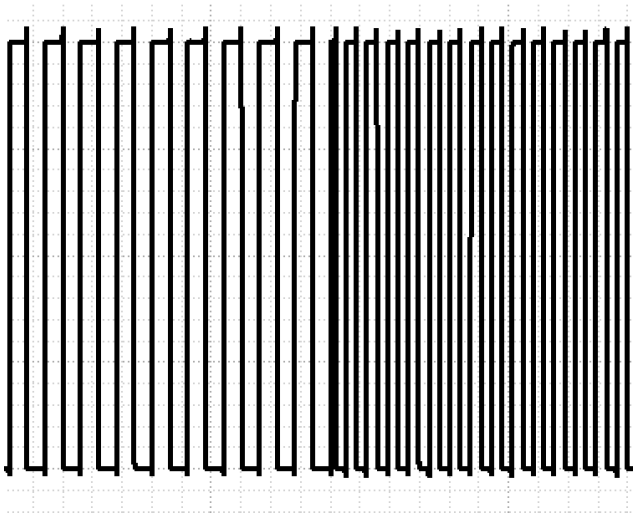

Figure 17: Example of suddenly shifting reference frequency.

To avoid this, a recovery mechanism is implemented. The recovery signal is an UP assertion into the charge pump independent of the phase detector. A simple register controls when the recovery signal occurs. Anytime a reference edge occurs but the phase detector's enable is not on, the register state will assert the recovery signal. The signal is asserted until the next reference edge comes in which case the register is toggled off. This should raise the rail voltage of the loop high enough such that the ratio of the loop frequency to the reference frequency is at least 4 again.

If the rail voltage is still not high enough, during the next rising edge, the recovery register will toggle on again and raise the rail voltage further. This means that it is likely that the rail voltage will overshoot which will increase the time it takes to lock to the correct frequency again. However, it ultimately allows the circuit to recover back to a state where it can adjust its frequency.

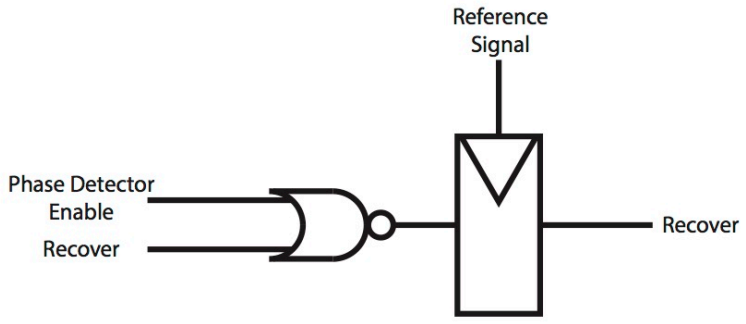

Figure 18: Basic overview of how the recovery signal is asserted.

3.3 Tunable Replica Circuit

The tunable replica circuits are designed to be configurable inverters chains with the configurability coming from the multiplexers. A digital calibration circuit approximates critical path of the processor and controls the multiplexers based on the estimation. The inverters are connected to the processor supply, which, as mentioned earlier, varies as a result of voltage scaling, and causes the invert delays to change appropriately with the supply voltage.

There are two tunable replica circuits; one is responsible for generating a pulse to control the rising edge of the final clock output and the other is responsible for the falling edge. This gives some flexibility in adjusting the duty cycle and period of the final clock output separately.

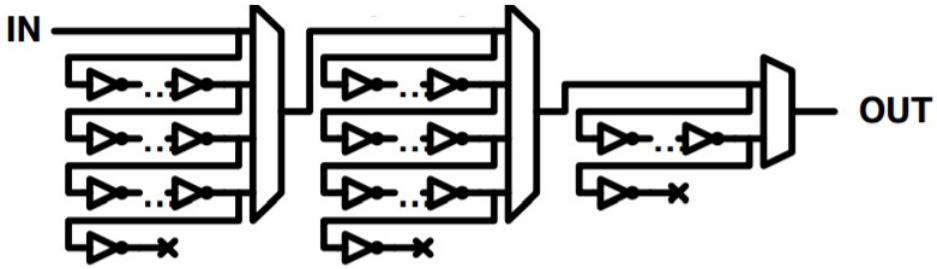

Figure 19: A tunable replica circuit. A signal goes though a delay chain and is input into a multiplexer at different points of the chain to get different delays [3].

3.4 Controller

The controller is responsible for generating the final output clock signal. This output signal is driven directly by the output of a flip-flop whose data input is connected to the 1V supply voltage. One tunable replica circuit is responsible for resetting the flip-flop, which causes the falling edge of the final output clock signal. The second tunable replica circuit determines the rising edge. The controller samples the closest rising delay-locked loop reference and that ultimately triggers the flip-flop to update its output to the 1V input, causing the rising edge of the final output clock signal.

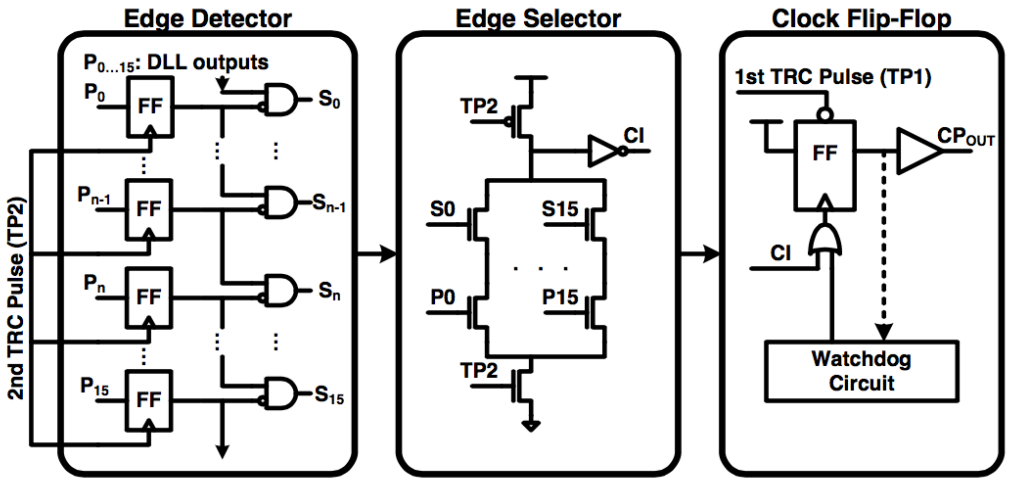

Figure 20: Structure of controller [3].

The delay-locked loop references and tunable replica circuit are asynchronous so the edge detector may experience metastability. To avoid metastability from killing the clock signal, a watchdog circuit monitors the output. It will generate an extra pulse for the clock output flip-flop if the final output clock remains low for too long.

4. Results

The clock generator was implemented using 28nm FDSOI. Most of the main parts such as the multiplying delay-locked loop's controller and tunable replica circuits were done using standard cells with the exception of a few such as the LDO's operational amplifier and charge pump. The functionality of the circuit was verified across three process corners (TT, FF, SS) and three temperatures (0, 27, 100 Celsius).

First, we verified it worked at the nominal 500MHz reference frequency as desired.

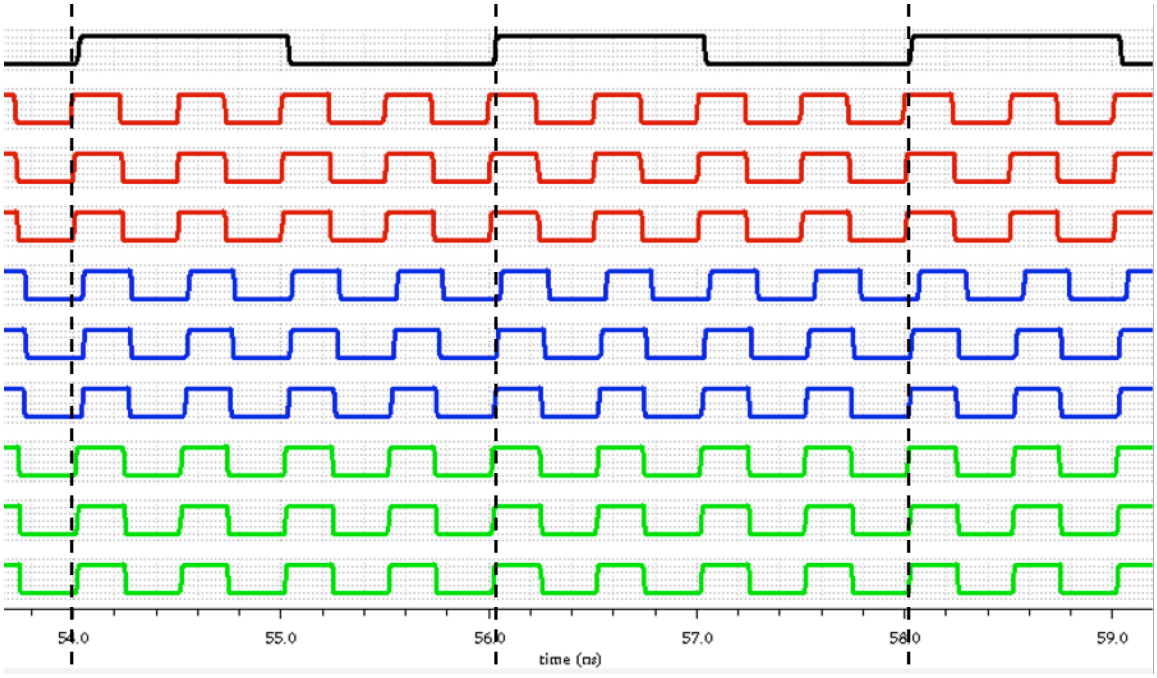

Figure 22: Waveform of the circuit. The top waveform (black) is the reference input clock. The others are one stage of the ring oscillator's waveform that is compared to the reference at different corners (red is the FF corner, blue is the SS corner, and green is the TT corner) at different temperatures (0, 27, 100 Celsius).

Then, we swept the input reference frequency to get the locking range of the circuit. The locking range in this case is considered valid input frequencies under which the multiplying delay-locked loop still oscillates at the desired 4 times input frequency and the control voltage rises and falls appropriately to adapt its frequency (there are cases where the circuit gave a proper output frequency that was 4 times faster than the reference but the control voltage was not behaving properly because the supply voltage hit its upper limit). According to simulation results, the higher end of the input frequency range is around the 556MHz and the lower end is around 286MHz.

As a comparison, the original delay-locked loop's locking range is 1.82GHz to 2.86GHz. Its first and last stages' phases must be initialized to a certain condition, which limits its locking range. Since the multiplying delay-locked loop injects a reference periodically, it does not have to worry about initial conditions.

The multiplying delay-locked loop consumes 1.231mW of power when running at 2GHz while the original consumed 1.478mW of power when running at the same frequency. The multiplying delay-locked loop consumed less power and we expect that using a slower reference also would use approximately 4 times less power on the clock distribution network.

To test the robustness of the controller circuit, we introduced an input that suddenly shifted frequencies from the lowest of the locking range to the highest and vice versa to confirm that the controller can correctly adjust to sudden frequency changes.

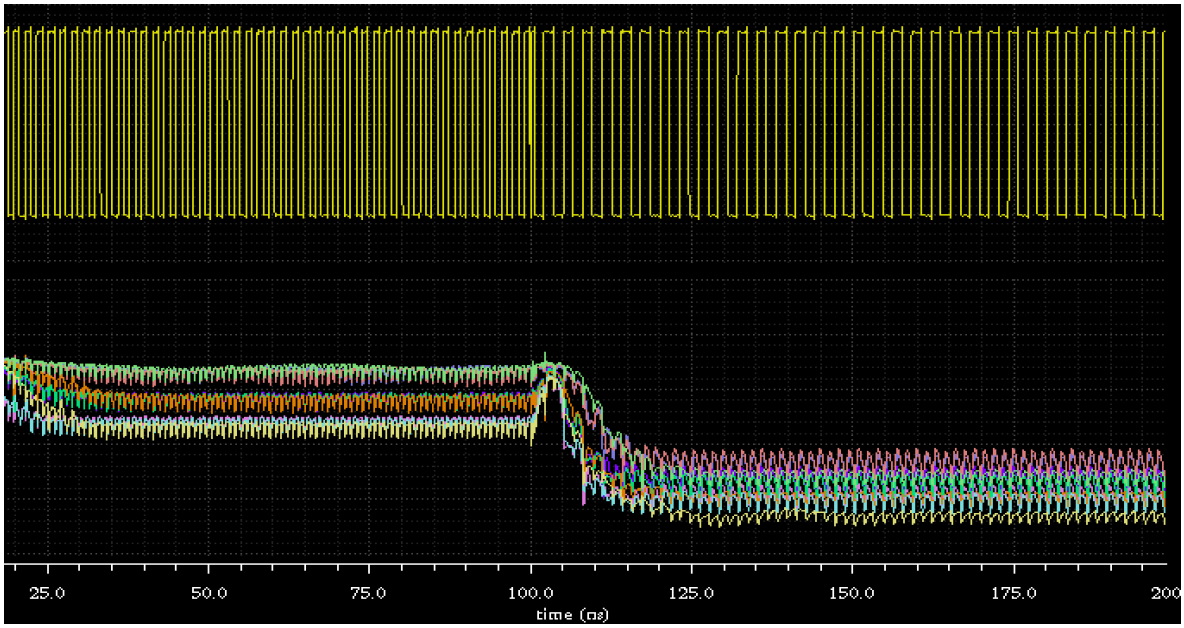

Figure 23: Waveform of the rail voltage (bottom) as the reference voltage (top) suddenly changes from the higher end of the locking range to the lower end across different corners and temperatures.

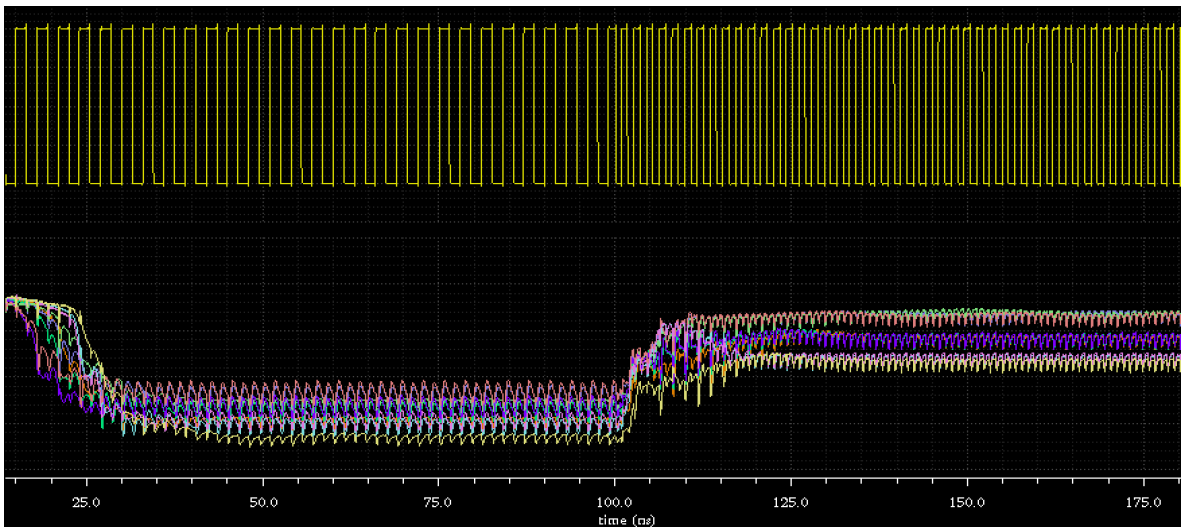

Figure 24: Waveform of the rail voltage (bottom) as the reference voltage (top) suddenly changes from the lower end of the locking range to the higher end across different corners and temperatures.

Another benefit of the circuit is changing the multiplicity by simply changing the comparison logic in the controller. We can also replace the greater than 3 combinational logic with another to increase the multiplication factor such that its output frequency can vary without changing the reference frequency or, conversely, change the reference frequency without changing the output frequency. As an example we used a 400 MHz reference signal and multiplied it by 5.

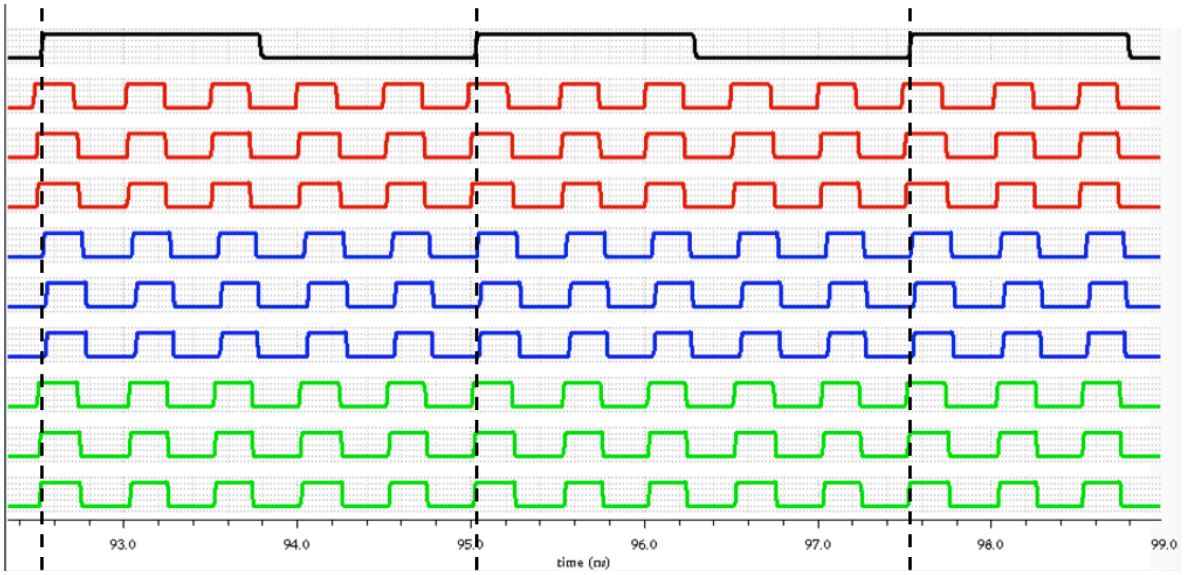

Figure 25: Waveforms of the circuit multiplying the reference frequency by 5 instead of 4 across different corners and temperatures.

5. Conclusion

5.1 Conclusion

The goal of this project was to design a clock generation circuit that used a slower reference clock than the previous version and would lend itself to better approach the issues with synchronization.

Using a slower reference clock uses less power gives the circuit higher flexibility after the frequency is multiplied. An edge combiner was initially considered as a method to multiply a slower frequency, but in the course of this project, we discovered that there were duty cycle issues stemming from uneven rise and fall times of the gates used in the edge combiner. Instead, we decided to speed up the delay loop in the multiplying delay-locked loop architecture. The comparator logic block responsible for comparing greater than or equal to 3 can easily be replaced by another block that can compare with higher values such as 4 or 5, and the controller locks the loops frequency based on the comparator.

The multiplying delay-locked loop currently functions nominally but lacks the fine-tuning that the original delay-locked loop has. More time will be needed to optimize it to avoid having as much jitter as it currently does. However, I believe the injection and controller logic give it more versatility than the original delay-locked loop had. Future efforts, such as reducing the circuit's jitter, is discussed in the following section.

5.2 Future Work

The jitter is currently a main concern of the circuit. Without having done much optimization, the observed jitter from multiplying delay-locked loop is undesirably high. It's exhibiting a behavior where each edge is consistently appearing picoseconds away from the others, causing groups of four to appear as shown in Figure 26.

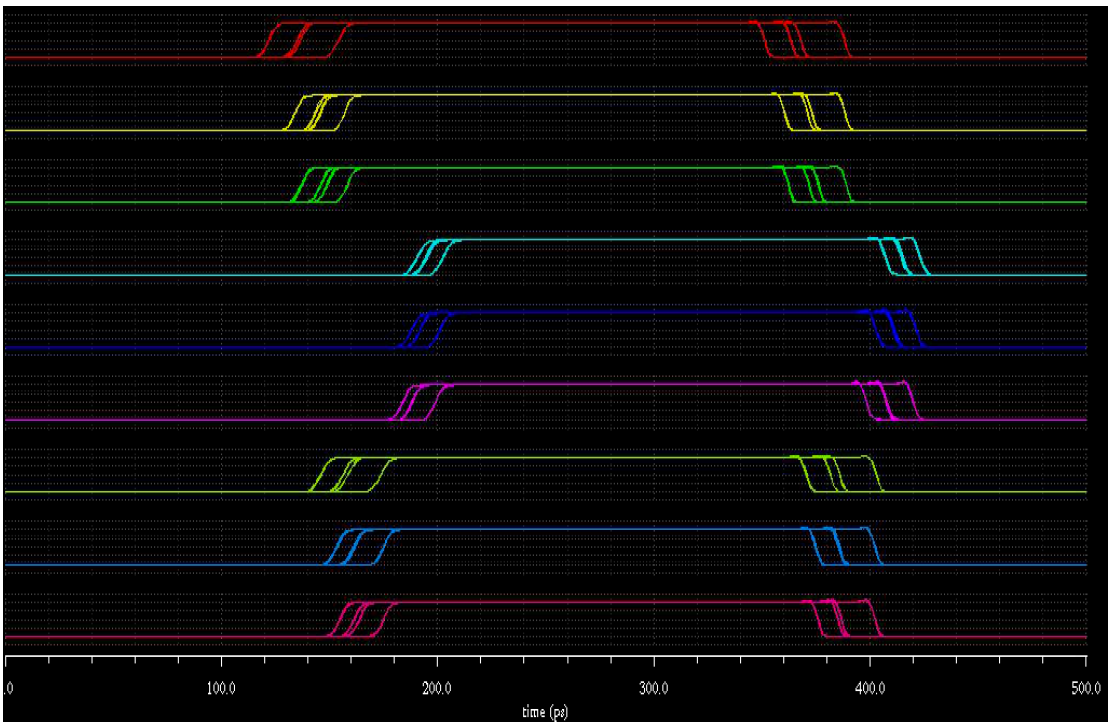

Figure 26: Waveform of the jitter in different corners and temperatures.

Another issue is that in the worse case corners there is an offset between the control voltage and actual rail voltage. While the rail voltage still tracks the control voltage, the offset prevents the circuit from working correctly beyond certain frequencies. Further tuning of the LDO circuit needs to be done to get rid of the offset.

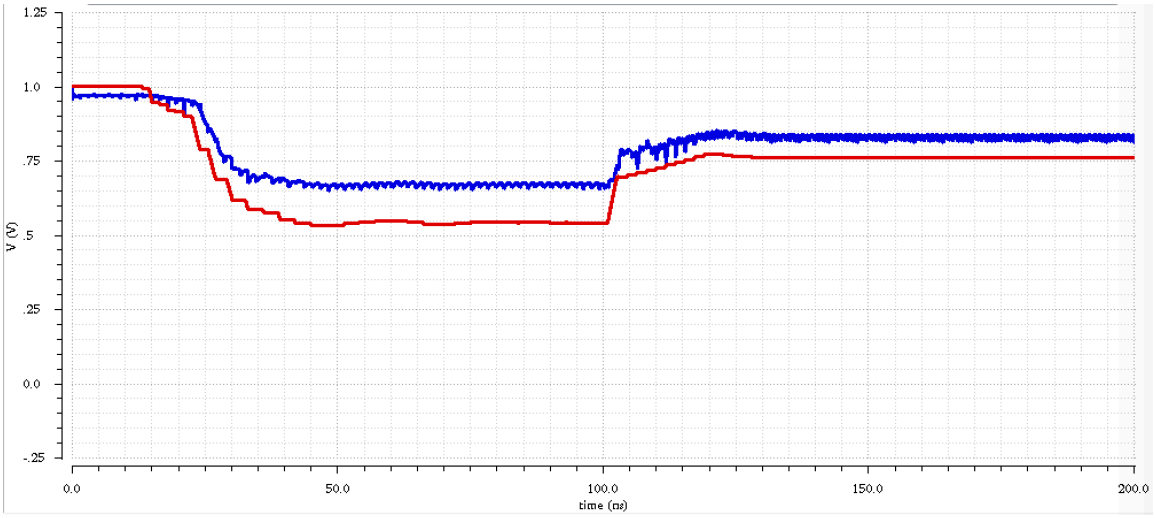

Figure 27: Graph demonstrating the offset problem between the control (red) and actual rail voltage (blue).

Acknowledgements

I would like to thank my professor Borivoje Nikolic and graduate students Benjamin Keller and John Wright for helping me with my project.

References

[1] Moore, Chuck, M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten. "Data Processing in Exascale-Class Computing Systems." Presented at the 2011 Salishan Conference on High-speed Computing.

[2] Restle, Phillip J., Timothy G. McNamara, David A. Webber, Peter J. Camporese, Kwok F. Eng, Keith A. Jenkins, David H. Allen, Michael J. Rohn, Michael P. Quaranta, David W. Boerstler, Charles J. Alpert, Craig A. Carter, Roger N. Bailey, John G. Petrovick, Byron L. Krauter, and Bradley D. McCredie, "A Clock Distribution Network for Microprocessors," in *IEEE Journal of Solid-State Circuits*, vol. 36, no.5, pp. 792-799, May, 2001.

[3] Kwak, Jaehwa and Borivoje Nikolic. "A Self-Adjustable Clock Generator with Wide Dynamic Range in 28nm FDSOI," in *IEEE Journal of Solid-State Circuits*, vol. 51, no. 10, pp. 2368-2379, October 2016.

[4] A. Grenat, S. Pant, R.Rachala, and S. Naffziger, "Adaptive Clocking System for Improved Power Efficiency in a 28nm x86-64 Microprocessor," in *Proc. IEEE International Solid-State Circuits Conference*, February 2014, pp.106-107.

[5] Li, Y. William, Carlos Ornelas, Hyung Seok Kim, Hasnain Lakdawala, Ashoke Ravi, and Krishnamurthy Soumyanath, "A Reconfigurable Distributed All-Digital Clock Generator Core With SSC and Skew Correction in 22nm High-k Tri-Gate LP CMOS," in *Proc. IEEE International Solid-State Circuits Conference*, 2012, pp. 70-71.

[6] Pandey, Jagdish and Brian P. Otis. "A Sub-100 uW MICS/ISM Band Transmitter Based on Injection-Locking and Frequency Multiplication," in *IEEE Journal of Solid-State Circuits*, vol. 46, no. 5, pp. 1049-1058, May 2011.

[7] Hu, Kangmin, Tao Jiang, Jingguang Wang, Frank O'Mahony, and Patrick Yin Chiang, "A 0.6 mW/Gb/s, 6.4-7.2 Gb/s Serial Link Receiver Using Local Injection-Locked Ring Oscillators in 90 nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 899-908, April 2010.

[8] Park, Pyoungwon, Jaejin Park, Hojin Park, and SeongHwan Cho, "An All-Digital Clock Generator Using a Fractionally Injection-Locked Oscillator in 65nm CMOS," in *Proc. IEEE International Solid-State Circuits Conference*, 2012, pp. 336-337.

[9] Deng, Wei, Ahmed Musa, Teerachot Siriburanon, Masaya Miyahara, Kenichi Okada, and Akira Matsuzawa, "A 0.022mm^2 970uW Dual-Loop Injection-Locked PLL with -243dB FOM Using Synthesizable All-Digital PVT Calibration Circuits," in *Proc. IEEE International Solid-State Circuits Conference*, 2013, pp. 248-249.

[10] "PLL Phase Locked Loop Tutorial". http://www.radio-electronics.com/info/rf-technology-design/pll-synthesizers/phase-locked-loop-tutorial.php.

[11] Stinson, J. "Clock Distribution Techniques". Stanford University.

[12] Gayathri. "What Is Metastability in Digital Circuits." November 24, 2012. http://tech.tdzire.com/what-is-metastability-in-digital-circuits/.

[13] "Behavioral modeling of Clock/Data Recovery". https://www.slideshare.net/arrowdevices/behavioral-modeling-ofclockdatarecovery.

[14] Farjad-Rad, Ramin, William Dally, Hiok-Tiaq Ng, Ramesh Senthinathan, M.J. Edward Lee, Rohit Rathi, and John Poulton. "A Low-Power Multiplying DLL for Low-Jitter Multigigahertz Clock Generation in Highly Integrated Digital Chips," in *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1804-1812, December 2002.

[15] Han, Jaeduk and Angie Wang. "Supply-Adaptive Performance Monitoring/Control Employing ILRO Frequency Tuning for Highly Efficient Multicore Processors." 2013.