A PLL Design based on a Standing Wave Resonant Oscillator

Vinay Karkala*, Kalyana C. Bollapalli*, Rajesh Garg‡, Sunil P. Khatri*
* Department of ECE, Texas A&M University, College Station TX 77843
‡ Intel Corporation, Hillsboro, OR 97124

***Abstract*— In this paper, we present a new continuously variable high frequency standing wave oscillator, and demonstrate its use in generating the phase locked clock signal of a digital IC. The ring based standing wave resonant oscillator is implemented with a plurality of wires connected in a mobius configuration, with a cross coupled inverter pair connected across the wires. The oscillation frequency can be modulated by two means. Coarse modification is achieved by altering the number of wires in the ring that participate in the oscillation, by driving a digital word to a set of passgates which are connected to each wire in the ring. Fine tuning of the oscillation frequency is achieved by varying the body bias voltage of both the PMOS transistors in the cross coupled inverter pair which sustains the oscillations in the resonant ring. We have validated our PLL design in a 90nm process technology. 3D parasitic RLCs for our oscillator simulations were extracted, with skin effect accounted for. Our PLL has been implemented to provide a frequency locking range from ∼6 GHz to ∼9 GHz, with a center frequency of 7.5 GHz. The oscillator alone consumes about 25 mW of power, and the complete PLL consumes a power of 28.5 mW. The observed jitter of the PLL is 2.56%.**

I. Introduction

In recent times there has been much interest in mobius ring based resonant oscillators as a means to generate the clock signal for digital ICs. Both traveling wave [1], [2], [3], [4] and standing wave [5] oscillators have been proposed in the literature. The typical configuration of these oscillators is a pair of closely spaced rings, implemented on higher metal layers on an IC. At one end, these rings are connected in a mobius fashion. This structure has RLC parasitics, and for reasonable values of the perimeter of the ring, it can exhibit high frequency oscillations. A single pair (or a multitude of pairs) of cross coupled inverters is connected between the 2 rings, to sustain the oscillation. Since charge is recirculated in these configurations, they exhibit a low power consumption. Resistive losses in the ring, as well as the power consumed by the inverter pair(s) contribute to the power consumption of these structures. By choosing the length of the ring carefully, oscillations of high frequencies can be sustained, as long as the inverter pair(s) can adequately switch at these frequencies. The traveling wave structure has been fabricated and impressive performance was demonstrated [6].

Both the standing and traveling wave configurations provide a means to generate a *free-running clock signal*. In other words, the resonant standing or traveling wave structure oscillates at a fixed frequency, determined by the parasitics of the rings. In practice, however, it is crucial that any oscillator in a digital system has the ability to modify its frequency in a predictable manner, so as to allow it to be integrated into a PLL. Without a PLL, a high frequency on-chip oscillator (such as a resonant standing or traveling wave oscillator) can easily exhibit significant skew compared to the external (board or system) clock. This would make it impossible to design a synchronous system using an IC with a free-running, high frequency resonant oscillator.

Typical oscillators on a digital IC use some form of Voltage Controlled Oscillator (VCO) [7], [8], [9], [10], [11], and implement a PLL with the VCO in a closed-loop configuration. A phase frequency detector (PFD) determines the phase error, and accordingly either speeds up or slows down the oscillation frequency of the VCO. VCOs are typically implemented in one of two ways

- *Analog* VCOs (ACOs) [12], [13], in which a ring oscillator with a small (odd) number of inverters is typically used. This ring oscillator's frequency is modified by means of a voltage or current signal.
- *Digitally Controlled Oscillators* (DCOs) [9], [10], [11], in which a large number of inverters are implemented such that inverter i drives the input to inverter $i+1$. By closing the loop at the k^{th} inverter (where k is odd), the oscillator can be made to oscillate at variable frequencies. The oscillator can be sped up or slowed down by decrementing or incrementing k respectively. In a PLL, this action is performed by a circuit whose input is the output of the PFD.

In this paper, we introduce a PLL which is implemented as a combination of both an AVCO and a DCO, using a resonant standing wave topology. Since the mobius rings are made up of wires on (typically higher) layers of the metal stack of the IC, it is hard to modify the parasitic inductance or capacitance of the resonant rings in practice. The parasitic inductance and capacitance of the rings are fixed once the wire dimensions, layer and spacing are determined. The key observation that enables frequency to be varied is the use of n wires in each ring (i.e $2n$ wires in all). By selecting between various configurations in which a variable number m of these wires are programmed to carry the ring signal (with the other n - m wires connected to a virtual ground terminal), we are able to vary the parasitic inductance and capacitance significantly, allowing the structure to behave like a DCO. This gives us the ability to tune the oscillator frequency in a coarse manner. For finer tuning of the oscillation frequency, we modulate the reverse body bias voltage of the PMOS transistors of the cross coupled inverter pair. We found that this is an effective way to change the ring capacitance in a continuous manner.

We have designed a PLL with the above resonant oscillator as one of its components. The PLL consists of coarse and fine tuning circuits. The coarse tuning circuit, with the help of a threshold detector, a successive one detector and a thermometer converter programs an appropriate number m of wires (out of n) to participate in the oscillation. The fine tuning circuit is a conventional third order PLL consisting of a PFD, charge pump and low pass filter. The fine tuning circuit changes the body bias voltage of the PMOS transistors of the inverter pair in order to modify the oscillation frequency.

The complete PLL has been simulated in HSPICE [14], using a 90 nm PTM [15] technology. Skin-effect adjusted parasitics of the mobius ring were extracted using Raphael [16].

The key contributions of this paper are:

- By providing both a coarse and fine control over the frequency of the resonant oscillator, we are able to demonstrate a continuous frequency response from ∼6 GHz to

978-1-4244-5028-2/09/$25.00 ©2009 IEEE

$\sim$9 GHz (with the reduced power of resonant standing wave oscillators).

- The PLL has been integrated with the proposed variable frequency oscillator, and is able to lock within the frequency range using the fine and coarse control circuits, based on our SPICE simulations.

The remainder of this paper is organized as follows. Previous work is described in Section II, while Section III provides the details of our resonant standing wave oscillator and the PLL. In Section IV we present results from experiments which we conducted to implement our PLL. We conclude in Section V.

II. Previous Work

Recently, a *traveling wave* resonant oscillator circuit (referred to as a *rotary clock*) was described and implemented [1]. The key idea in this approach is to utilize a sufficiently long wiring ring, such that its capacitive and inductive parasitics result in a high frequency oscillatory network. In a resonant clock topology oscillations in the network are sustained by a plurality of inverter pairs spaced along the ring.

[4] implements a 2.5 GHz PLL using a traveling wave oscillator, to sample input data and perform clock recovery using the 24-phase distributed VCO. The design recovers 16 bits within a clock period. Though it is advantageous for Clock Data Recovery (CDR), the varying phase of the traveling wave clock makes traditional synchronous clock based design extremely difficult. Also, the clock signal at every point of the ring is a full-rail signal, resulting in a larger power consumption.

In response to this, a *standing wave* resonant oscillator circuit was proposed [5]. In this approach, a long wiring ring is used, but oscillations are sustained in this resonant ring by just using a single inverter pair (Figure 1). By making a mobius connection at the end of the ring, the clock signal at any point in the ring is sinusoidal, but has the *same phase* at all points along the ring.

By using differential amplifiers at different points in the ring, full rail clock signals are extracted at the locations desired. As a consequence, this approach yields clock signals that have the same phase everywhere along the ring. This is a key improvement over the rotary clock of [1]. In addition, the reduced ring capacitance due to the use of significantly fewer inverters (in particular, just one), increases the operating speed and reduces power consumption as well. Note that there is an AC null (virtual "zero") point in the center of the ring. As a result, the phases of the signals on the right and the left of the null point are 180° apart. Therefore, clock recovery circuits on the left have their connections reversed compared to recovery circuits on the right of the null point. Note that clock recovery is not performed around the null point, since the signal amplitude is very low near the null point.

In [17], another high-frequency standing wave oscillator was proposed. It is based on the use of multiple coupled oscillators (each comprised of an NMOS cross-coupled pair to sustain the oscillation, and a PMOS diode connected load for setting the common mode voltage). Unlike our approach, [17] achieves a very small (6.4%) locking range (we achieve a $\sim$ 33% locking range from $\sim$6 GHz to $\sim$9 GHz). The approach of [17] does not use a mobius connection like our approach does.

As described earlier, the resulting clock for all the above approaches is free-running, and since the inductive and capacitive parasitics of the ring are fixed, the above approaches do not lend themselves to realizing a variable oscillation frequency.

The authors of [18], [19], [20] have implemented a PLL using a combination of an ACO and a DCO, with frequencies around 900 MHz, 1 GHz and 10 GHz respectively, but none of them use a resonant oscillator. None of the previous approaches uses indcutance based coarse tuning.

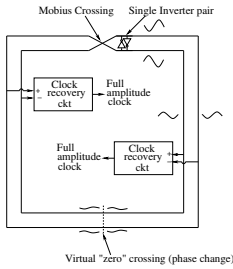


Fig. 1. Standing-wave Resonant Clock [5]

Using our approach, we can effectively control the ring inductance and capacitance (and hence the frequency of oscillation), making the resonant oscillator a good candidate for the oscillator block of a PLL. In general, longer mobius rings can be obtained by interlinking smaller rings in both [1] as well as [5].

III. Our Approach

A. Design Goal

The design goals of our approach are to realize a resonant oscillator with a high center frequency and frequency range, and to demonstrate the working of a PLL which incorporates this resonant oscillator.

The equivalent circuit for our resonant oscillator is shown in Figure 2. In this figure, L_w and C_w refer to parasitic inductance and parasitic capacitance respectively. The capacitance due to the cross-coupled inverter pair (i.e. twice the sum of the diffusion and gate capacitances of any inverter in the pair) is C. Since C and C_w are in parallel, we obtain the equivalent circuit shown.

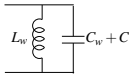


Fig. 2. Equivalent Circuit for Our Resonant Oscillator

The oscillation frequency of the equivalent circuit is given by

$$f = \frac{1}{2\Pi\sqrt{L_w(C+C_w)}} \quad (1)$$

B. Oscillator Design

1) Coarse Frequency Control of the Standing Wave Oscillator: In order to realize a variable frequency oscillator, we modify the base design of [5]. The major modification is to use a large number n of wires in place of the single wire that is used to implement each ring in [5]. By using a subset of the n wires for oscillation, we are able to modify the parasitic inductance and capacitance of the ring on the fly, allowing us to realize a variable frequency standing wave oscillator. The wires are used symmetrically about the midpoint of the $2n$ wire bundle for oscillation. Each subset of the n wires that we use for oscillation is referred to as a *coarse configuration*.

w_1	w_2	w_3	y_3	y_2	y_1	
1	1	1	1	1	1	← Coarse config 1
1	1	0	0	1	1	← Coarse config 2
1	0	0	0	0	1	← Coarse config 3
Wires of outer ring			Wires of inner ring			

Fig. 3. Coarse Frequency Configuration

To simplify the digital control logic, we select a significantly reduced subset of the $2^n - 1$ possible coarse configurations. Figure 3 illustrates the configurations we use, with

n = 3. In this figure, w_1 refers to the outermost wire of the outer ring, and y_1 is the outermost wire of the inner ring. In this figure, coarse configuration 1 (2) uses 6 (4) wires for oscillation (indicated by a '1' in each of the positions). Note that all coarse configurations are symmetric around the midpoint of the bundle of $2n$ = 6 wires. We simulated our oscillator with n = 30.

In practice, the wire locations that are labeled as '0' in Figure 3 are actually left floating by the control logic. Assuming that the supply voltage of the inverter pair is VDD, both rings oscillate around a DC value $VDD/2$, with sinusoidal waveforms which are always in phase, but whose amplitude varies as we traverse the ring. Since the null oscillation point (labeled as *virtual "zero" crossing* in Figure 1) applies for *all* configurations, we short all $2n$ wires at this virtual ground location. As a result, all wires that are left floating by the control logic actually have a $VDD/2$ voltage on them due to the short at the virtual ground location, and therefore act as ground wires in an AC sense.

From Figure 3, suppose we have two configurations with numerical indices P and Q respectively. Let $P < Q$. Then there are more oscillating wires in the inner and outer rings for P (as compared to Q). Also, the distance between oscillating wires in the two rings is lower for P. This has two effects.

- The capacitance of the oscillating wires is larger for P as compared to Q, since P has more oscillating wires.
- The inductance of P is lower than that of Q, since the current return loop is smaller in P compared to Q, due to proximity effect.

The ratio of the increase in capacitance of P over Q is less than the ratio of the increase in inductance of Q over P. As a result, based on the frequency of oscillation of the ring (Equation 1), P oscillates at a higher frequency than Q.

Our resonant standing wave oscillator is controlled by varying the values of the n-bit vectors **w** and **y**.

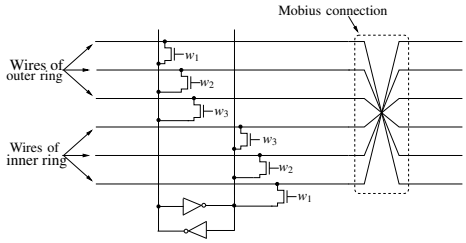


Fig. 4. Coarse Configuration Selection and Mobius Connection of $2n$ Wires

The circuitry for coarse frequency control is illustrated in Figure 4. This circuit takes as input the vectors **w** and **y**. Based on the values of these vectors, the appropriate wires among the $2n$ wires of the oscillators are made to oscillate. If coarse configuration 2 is chosen, for example, only the top 2 and the bottom 2 wires in Figure 4 oscillate. Note that this circuit resides at the mobius point of the resonant oscillator, and the cross-coupled inverter pair is shown in the figure as well. The mobius connection of the $2n$ wires is illustrated to the right of Figure 4. Note that in practice $w_i = y_i$.

In practice, the switches in Figure 4 are NMOS passgates. We tried complementary passgates, but the diffusion capacitance of complementary passgates caused a noticeable drop in oscillation frequency. In order to decrease the body effect, we connect the source and bulk terminals of these NMOS passgates. The coarse tuning approach achieves a maximum (minimum) frequency of a 9.2 GHz (6.2 GHz) in our design.

2) Fine Frequency Control of the Standing Wave Oscillator: For fine frequency control, we take advantage of the fact that the capacitance of the cross coupled inverter pair contributes significantly towards the total capacitance of the resonant ring. One of the major contributors of cross coupled inverter capacitance is the drain to bulk depletion capacitance of both the PMOS and NMOS transistors. This depletion capacitance is varied by changing the bulk voltage. An increase (decrease) in the body bias voltage (i.e an application of reverse body bias) of a PMOS (NMOS) transistor would result in an increase in depletion width, reducing the depletion capacitance and resulting in a decrease of the overall ring capacitance. This results in an increase in the frequency of oscillation of the ring.

In our implementation, we varied the body bias voltage of both the PMOS transistors from 1.2V to 2.4V to achieve fine tuning of the oscillator frequency. Fine frequency control could also be achieved by using varactors, but this would require additional components in the design.

3) Integrated Coarse and Fine Tuning: In this section, we describe the approach by which we designed an oscillator with a continuous frequency range from 6 GHz to 9 GHz, by combining coarse and fine tuning as shown in the Figure 5. In the figure, f_0 is the operating frequency under current conditions and f is the final desired frequency. The dots in the figure are the coarse configuration points. In Figure 5, in order to speed up from f_0 to f fine tuning is done till a coarse point is reached. Because the oscillator has to speed up further after reaching this coarse point, a coarse jump takes place. Another coarse jump takes place since the oscillator has to speed up even further. After this, the oscillator has to slow down, and since the desired frequency is within the coarse range, fine tuning is done to reach f.

One important aspect that lets us achieve integration of coarse and fine tuning is the fact that the frequency range spanned by reverse body biasing the PMOS transistors of the inverters at every coarse configuration is greater than the difference in frequency between the two adjacent coarse frequency points. This is important as it allows us to achieve a continuously adjustable frequency in the range of 6 GHz to 9GHz without any "holes" in frequency coverage.

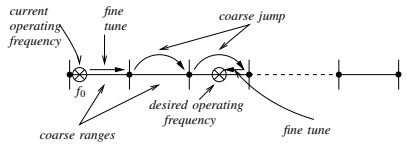


Fig. 5. An Example to Show Coarse and Fine Tuning Integration

C. Proposed PLL Design

1) Overview: Figure 6 shows the block diagram of our PLL with the resonant oscillator incorporated. All the components above the dotted line in the figure constitute the fine control circuit and the components below the dotted line constitute the coarse control circuit. The fine tuning circuit of the PLL is a conventional PLL consisting of a PFD, charge pump, low pass filter, VCO and a divider. The charge pump output is used to control the bulk voltage of the PMOS transistors of the cross coupled inverter pair, thereby achieving fine frequency control. The coarse tuning circuit consists of two threshold detectors, two successive one detectors and a thermometer converter. The threshold detector output goes high if the control voltage, which is the charge pump output (*bulk*) voltage, reaches its highest (lowest) value. The output of the successive one detector goes high if the output of the corresponding threshold detector stays high for more than a predetermined number of clock cycles. A rising transition in the output of either of the successive one detectors results in a change in the state of thermometer converter, which drives the digital word

that is used to program the oscillator to a particular coarse configuration.

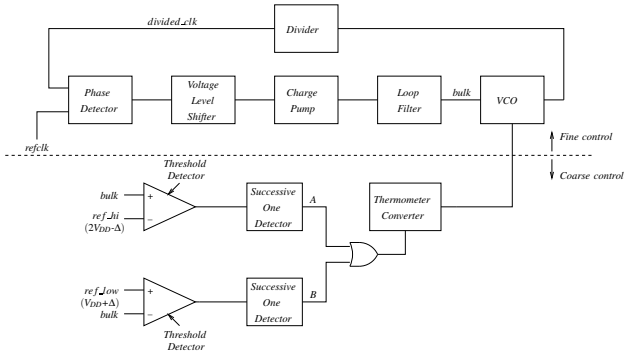


Fig. 6. Block Diagram of the Proposed PLL Design

2) Coarse Control: A brief description of each of the components of the coarse control is given below.

Threshold Detector:

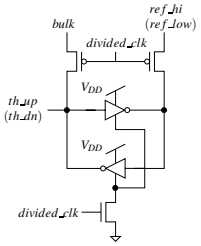


Fig. 7. Threshold Detector Circuit

If the control voltage exceeds (or goes below) a predetermined threshold value, the threshold detector output goes high. Figure 7 shows the circuit of the threshold detector that we used. Two threshold detectors have been used in the PLL circuit to detect when the control voltage (*bulk*) exceeds the threshold voltage *ref_hi* or when it goes below the threshold voltage *ref_lo*.

The threshold detector circuit has a pair of PMOS transistors whose gates are driven by *divided_clk*, and source terminals of the transistors are connected to *bulk* and *ref_hi* (*ref_lo*). The drain terminals of the PMOS transistors are connected across a cross coupled inverter pair whose ground terminal is gated with a NMOS transistor whose gate is driven by *divided_clk*.

The working of the circuit is as follows. When *divided_clk* goes low, the voltage on *bulk* and *ref_hi* (*ref_lo*) terminals is sensed by the nodes of the cross coupled inverters (whose ground terminal is floating as the NMOS transistor is off). When *divided_clk* goes high both the PMOS transistors are off and NMOS transistor is on, and hence the ground terminal is not floating anymore. The intermediate nodes are pulled to either *VDD* or *GND* based on which of these nodes was at a higher voltage when *divided_clk* went high.

In our simulations the value of the voltage of reference nodes *ref_hi*, *ref_lo* was 2.3V and 1.3V respectively, we assume that these voltages are provided externally since the range of voltages of the *bulk* node is 1.2 V to 2.4 V.

Successive One Detector (SOD): The successive one detector detects if the control voltage (*bulk*) exceeds (or goes below) the threshold voltage for more than a particular number of cycles of *divided_clk*. It is a shift register as shown in the Figure 8. Its output *A* (*B*) goes high when the output of all the flip-flops in the shift register go high.

The outputs of successive one detectors drive the clock signal of the thermometer converter. All the flip-flops of the detector have to be reset after a particular number of cycles of *divided_clk*. This is because if there is a coarse jump in order to speed up (slow down) the clock, and if the oscillator needs to speed up (slow down) yet further then *A* needs to be reset before it rises again, since the rising edge of *A* triggers a change to the next faster coarse configuration. This is achieved by having additional flip-flops in the shift register of the SOD. In Figure 8, the *A* (*B*) signal rises if *th_up* (*th_dn*) is high for three consecutive cycles of *divided_clk*. This triggers a transition to the next higher (lower) frequency coarse configuration. If another shift is required to the next higher (lower) frequency configuration, *th_up* (*th_dn*) continues to stay high, and after seven more consecutive cycles of *divided_clk*, the *reset* signal resets the flip-flops of the SOD. Now if *th_up* (*th_dn*) stays high for three more cycles, *A* (*B*) rises again, causing a transition to the next higher (lower) coarse frequency configuration.

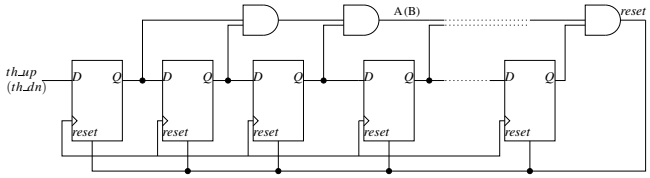


Fig. 8. Successive One Detector Circuit

Thermometer Converter: The next coarse configuration can be arrived at by an arithmetic right shift of the current state (to speed up) or an arithmetic left shift of the current state (to slow down). The thermometer converter shown in Figure 9 implements this logic. The thermometer converter consists of a series of D flip-flops connected through MUXes. The output of a particular flip-flop stage is the input to the MUX, and it drives the MUX output when the control signal *A* (output of the successive one detector which goes high if there has to be a change in coarse configuration in order to speed up) is high. The input to the first stage of the MUX that is selected when *A* goes high is *VDD*. Hence when *A* goes high (indicating that we need to shift to the next faster coarse configuration), the 1's in the thermometer shift to the right. The other input to the MUX is the output of the D flip-flop two stages away from the current flip-flop being considered. The input to the last stage of the MUX that is selected when *A* goes low is *GND*. Therefore if *A*=0 and *B* goes high (indicating that we need to change to the next slower coarse config), the 1's of the thermometer shift to the left with a 0 injected into the last stage. The clock signal of all the flip-flops is the output of an *OR* gate whose inputs are *A* and *B* of the two successive one detectors.

As we used *n*=30 wires in our resonant ring, we used 30 flip-flops and 30 MUXes in order to configure the oscillator.

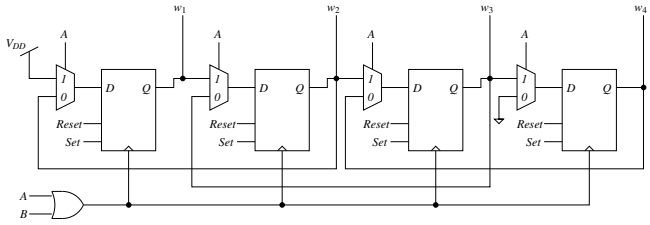
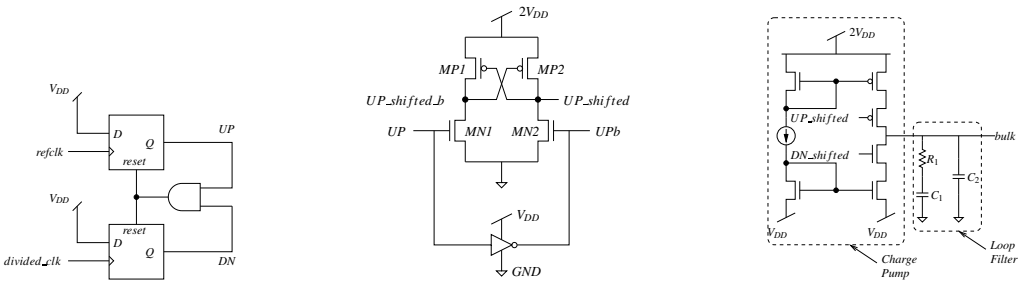


Fig. 9. Thermometer Converter Circuit

3) Fine Control: A brief description of each of the components of the fine control is shown below.

Phase Frequency Detector (PFD): The PFD consists of two flip-flops and an *AND* gate connected as shown in the Figure 10 a). When *ref_clk* leads (lags), the *divided_clk* pulse width of the *UP* (*DN*) signal is the phase error between the two signals. Under a lock condition, short pulses will be generated on both the *UP* and *DN* outputs.



(a) PFD (b) Level Shifter (c) CP and LPF

Fig. 10. Fine Tuning Circuits

Voltage Level Shifter: In the fine tuning circuit that we used, we require a voltage level shifter in order to drive a charge pump which is connected between $2VDD$ and VDD. The charge pump drives the bulk of the PMOS transistors of the cross coupled inverters with a voltage within this range, to reverse body bias the PMOS transistors.

The outputs of the PFD are the inputs to the two voltage level shifters. The circuit for the voltage level shifter used for the UP signal is shown in the Figure 10 b). An identical voltage level shifter for the DN signal was used. The voltage level shifter requires two power supplies, the input domain voltage supply (VDD) and the output domain voltage supply ($2VDD$). When the input signal UP (DN) is at VDD, MN1 turns on and MN2 is off and thus pulls the $UP\_shifted\_b$ ($DN\_shifted\_b$) signal to GND. This transition in $UP\_shifted\_b$ turns on MP2, which pulls the $UP\_shifted$ ($DN\_shifted$) signal to $2VDD$.

Charge Pump and Low Pass Filter: The level shifted pulses $UP\_shifted$ and $DN\_shifted$ must be converted into an analog voltage that controls the voltage of the bulk node of PMOS transistors of the cross coupled inverters. This has been implemented by making use of a simple charge pump and a second order filter as shown in the Figure 10 c). The charge pump implemented is a pair of current sources being switched by using the $UP\_shifted$ and $DN\_shifted$ signals. A pulse on the $UP\_shifted$ ($DN\_shifted$) signal adds (removes) charge to the capacitors at the output, proportional to its pulse width. If the width of $UP\_shifted$ pulse is larger than $DN\_shifted$ pulse there is an effective increase in the output ($bulk$) voltage, which increases the reverse body bias and hence the frequency of the oscillator.

The loop filter consists of a resistor R_1 and capacitors C_1 and C_2, and hence is a second order filter, making the system third order. Having only a capacitor at the output of the charge pump would result in a open loop transfer function of second order, with both poles located at the origin. This would render the system unstable as each of the poles contributes a constant phase shift of 90° resulting in a 180° phase shift before the unity gain crossover frequency, causing the system to oscillate. Hence, in order to stabilize the system, phase characteristics have been modified by introducing a zero in the loop gain by adding a resistor (R_1) in series with the loop filter capacitance (C_1). Even though the system is stable, the series combination of R_1 and C_1 could result in a large control voltage ripple that severely disturbs the oscillator. In order to suppress these ripples an additional capacitance (C_2) was added.

Divider: In order to achieve an oscillator running with a desired frequency in the range of 6 GHz to 9 GHz, the oscillator output frequency has to be divided by a constant factor of 128 in order to compare the phase and frequency with the 50 MHz reference clock. The divider that we used was a 7-bit ripple counter. A dynamic D flip-flop with carefully sized gates has to used at the first stage, as the value of $max(t_{setup} + t_{clktoQ})$ must not exceed the minimum desired period of the oscillator. Starting from the second stage we can use either static or dynamic D flip-flops. In our PLL, we used dynamic D flip-flops for the second and higher stages as well.

Index	**w** value
1	1111111111111111111111111000000
2	1111111111111111111111110000000
3	1111111111111111111111100000000
4	1111111111111111111111000000000
5	1111111111111111111110000000000
6	1111111111111111111100000000000
7	1111111111111111110000000000000
8	1111111111111111100000000000000
9	1111111111111111000000000000000
10	1111111111111110000000000000000
11	1111111111111100000000000000000
12	1111111111111000000000000000000
13	1111111111110000000000000000000
14	1111111111100000000000000000000
15	1111111111000000000000000000000
16	1111111110000000000000000000000

TABLE I

COARSE CONFIGURATIONS USED IN OUR EXPERIMENTS

IV. EXPERIMENTS

We implemented the PLL described in this paper, using a 90nm BSIM3 PTM [15] process technology. The power supply voltage was 1.2V, and all simulations were conducted in HSPICE [14].

A. Oscillator Design

The inner and outer wiring rings of the oscillator consist of n = 30 wires each. All the 60 wires were implemented on METAL8, and the total length of the ring was 1500 μm. Each of the 60 wires were 1 μm wide, with an inter-wire spacing of 1 μm. In all the results presented in this paper, we ensured that at least 80% of the ring (except for the region near the virtual ground point) could be used to recover a rail-to-rail clock using a clock recovery circuit. All large MOSFETs were implemented using multiple *fingers* which shared diffusions, thereby reducing the parasitic capacitances in our design.

The 16 coarse configurations that we used are shown in Table I. Note that for each configuration, we report the 30 values of **w**.

We next swept the size of the cross-coupled inverters in HSPICE. We observed that for larger inverter sizes, the inverters' diffusion and gate capacitance resulted in slower oscillations. For smaller inverter sizes, we found that not all of our coarse configurations could sustain oscillations in the ring. Sometimes, even if a smaller inverter size could sustain the oscillation, it allowed a rail-to-rail clock to be recovered from less than 80% of the ring. We eliminated these inverter sizes from consideration since they did not meet the 80% recovery requirement. We found that the optimal width of the cross-coupled inverters was 160μm (66μm) for the PMOS (NMOS) device. This allowed all configurations to sustain oscillation, and also passed the 80% recovery requirement.

To size the NMOS passgates of Figure 4, we conducted a SPICE sweep starting with a minimum sized passgate.

As we increased the passgate size, we found that more configurations were able to sustain oscillation. For an NMOS passgate of size 20μm, oscillations were sustained by every configuration of Table I. Also, the 80% recovery rule was met in each case.

We have simulated the oscillator in HSPICE [14], with skin-effect adjusted parasitics extracted using Raphael [16]. Our oscillator provides a continuous and substantially linear frequency response from $\sim$6 GHz to $\sim$9 GHz, with a center frequency of 7.5 GHz.

The power consumption of our oscillator varies between configurations, but the range of power consumption values stayed between 23.5mW and 25.5mW. As expected, more power was typically consumed at higher frequencies.

The two rings of our oscillator sustain a sinusoidal oscillation. To recover a rail-to-rail clock from any point on the ring, a clock recovery circuit (as shown in Figure 1) is required. This circuit is essentially a differential amplifier with a buffered output. A plot of several recovered signals from around the ring is shown in Figure 11. From this figure, the rising skew is 3.44ps, while the falling skew is 3.99ps (for a clock of period 142ps).

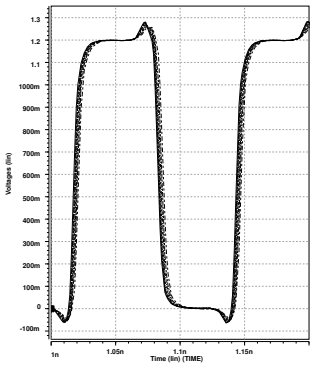


Fig. 11. Recovered Waveforms

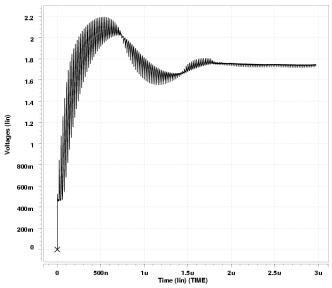


Fig. 12. Control Voltage

B. PLL Design

The PLL designed consists of coarse and fine tuning circuits. The fine tuning circuit is a conventional third order PLL design. The main parameters that determine the stability of the system are R_1, C_1 and C_2 which are the components of the loop filter as shown in the Figure 10 c). These values have to be chosen with care in order to ensure that the system is stable. The value of K_0 varies from 125 MHz/V to 375 MHz/V across the coarse configurations. Using the set of design equations in [21] with the highest value of K_0 the values of R_1, C_1 and C_2 turned out to be 9242 Ω, 16.067 pF and 1.24 pF respectively. The stability of the system was verified for various values of K_0 with the above calculated values of R_1, C_1 and C_2 in MATLAB.

Figure 12 shows the control voltage waveform in a particular coarse range. The lock time from boot within the coarse range based on the control voltage profile is 2 μs for the given reference frequency (50 MHz). Note that the time to switch between adjacent coarse ranges is 1280 cycles. For an operating frequency of 7.5 GHz this corresponds to 0.13 ns. Hence the time to switch from the slowest to the fastest coarse range is 15$\times$0.13 = 1.95 ns. Hence the total lock time is dominated by lock time of the fine tuning circuit. We also verified that our PLL locked correctly to the reference when this required several coarse ranges to be traversed in order to lock.

The total power dissipated in the PLL at lock was found to be 28.5 mW and the total jitter observed was 2.56% and the jitter with a perfect clock recovery circuit was 0.36%.

V. Conclusions

In this paper, we present a PLL, based on a standing wave oscillator, for use in generating the clock signal of a digital IC. The resonant ring is implemented with a plurality of wires connected in a mobius configuration. The oscillation frequency is modulated by coarse and fine tuning. Our oscillator was implemented to provide a continuous frequency response from $\sim$6 GHz to $\sim$9 GHz The length of the ring considered was 1500 μm. Parasitic RLC extractions have been performed for these simulations, with skin effect accounted for. The PLL with the resonant oscillator incorporated was validated using a 90nm process technology. The total power dissipated in the PLL at lock is about 28.5 mW of which the oscillator alone dissiptes 25 mW and the jitter observed at lock is 2.56%.

References

[1] J. Wood, T. Edwards, and S. Lipa, "Rotary traveling-wave oscillator arrays: a new clock technology," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1654–1665, Nov 2001.

[2] S. Chan, P. Restle, K. Shepard, N. James, and R. Franch, "A 4.6GHz resonant global clock distribution network," *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, pp. 342–343 Vol.1, Feb. 2004.

[3] MultiGig http://www.multigig.com.

[4] T. Nedovic *et al.*, "A 40-44 gb/s 3 x oversampling cmos cdr/1:16 demux," *IEEE Journal of Solid-State Circuits*, pp. 2726–2735, 2007.

[5] V. Cordero and S. Khatri, "Clock distribution scheme using coplanar transmission lines," in *DATE*, pp. 985–990, 2008.

[6] J. Wood, T. Edwards, and C. Ziesler, "A 3.5GHz rotary-traveling-wave-oscillator clocked dynamic logic family in 0.25 μm CMOS," *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pp. 1550–1557, Feb. 2006.

[7] A. Tasic, W. Serdijn, and J. Long, "Adaptivity of voltage-controlled oscillators - theory and design," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 52, pp. 894–901, May 2005.

[8] A. Megej, K. Beilenhoff, and H. Hartnagel, "Fully monolithically integrated feedback voltage controlled oscillator [using PHEMTs]," *Microwave and Guided Wave Letters, IEEE*, vol. 10, pp. 239–241, Jun 2000.

[9] J. Y. H. K. J. Yamada, Takaaki (Kanagawa, "Digital voltage controlled oscillator having a ring oscillator with selectable output taps," August 1993.

[10] M. Thamsirianunt and T. Kwasniewski, "CMOS VCO's for PLL frequency synthesis in GHz digital mobile radio communications," *Solid-State Circuits, IEEE Journal of*, vol. 32, pp. 1511–1524, Oct 1997.

[11] A. Hajimiri, S. Limotyrakis, and T. Lee, "Jitter and phase noise in ring oscillators," *Solid-State Circuits, IEEE Journal of*, vol. 34, pp. 790–804, Jun 1999.

[12] G. Jovanovic and M. K. Stojcev, "Current starved delay element with symmetric load," *International Journal of Electronics*, vol. 93, pp. 167–175, March 2006.

[13] O.-C. Chen and R.-B. Sheen, "A power-efficient wide-range phase-locked loop," *Solid-State Circuits, IEEE Journal of*, vol. 37, pp. 51–62, Jan 2002.

[14] HSPICE http://www.synopsys.com/products/mixedsignal/hspice/hspice.html.

[15] PTM http://www.eas.asu.edu/ ptm.

[16] "Raphael Interconnect Analysis Tool: User's Guide." Paul Kostek, `http://www.todaysengineer.org/2008/Jun/semiconductor.asp`.

[17] F. O'Mahony, P. Yue, M. Horowitz, and S. Wong, "Design of a 10GHz clock distribution network using coupled standing-wave oscillators," in *DAC '03: Proceedings of the 40th conference on Design automation*, pp. 682–687, ACM, 2003.

[18] T.-H. Lin and W. J. Kaiser, "A 900-mhz 2.5-ma cmos frequency synthesizer with an automatic sc tuning loop," *IEEE Journal of Solid-State Circuits*, pp. 424–431, 2001.

[19] K. L. William B. Wilson, Un-Ku Moon and L. Dai, "A cmos self-calibrating frequency synthesizer," *IEEE Journal of Solid-State Circuits*, pp. 1437–1444, 2000.

[20] T.-H. Lin and Y.-J. Lai, "An agile vco frequency calibration technique for a 10-ghz cmos pll," *IEEE Journal of Solid-State Circuits*, pp. 340–349, 2007.

[21] T. H. LEE, *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge university press.