

A Comparative Approach for the Design of Delay Locked Loop with Low Power Consumption

¹P.Latha, ²Dr. S.Ramasamy and ³K.Vigneshraja

¹Associate Professor, ²Professor, ³Student

Department of ECE, RMK Engineering College, Chennai, India

Abstract-- For the common applications such as frequency generation, clock recovery, clock synchronisation, a Phase locked loop architecture will be generally used. To get a more stable operation Delay locked loops can be preferably used. This paper presents a comparative approach for the design of delay locked loops with application specific performance and low power consumption. Various phase detectors are compared here and a modified delay cell is proposed and the same is being used with various PD. A justification is brought about for using various PD for the respective applications. A comparison of DLLs using the normal and the modified delay cell are also presented here. A basic problem in DLL such as dead-zone problem is also addressed in this work. The figure of merit (FOM) is also calculated here and this work achieves a good FOM of 73. A Multiplexing architecture is also proposed with two different application specific DLLs combined together. The DLL architectures are implemented in Cadence Virtuoso schematic editor using UMC180nm. In addition, a frequency multiplying scheme using DLL is also presented here.

Keywords-- Dynamic PFD * PFD - GDI * Static and Dynamic Delay cell * Figure of merit, Multiplexing Architecture

I. INTRODUCTION

Delay locked loops (DLLs) have been widely used as frequency synthesizers and clock de-skewing circuits in radio frequency (RF) transceivers inter-chip communication interfaces and clock distribution networks. In addition to frequency multiplication, using DLL for signal synchronization would be the better choice than using a PLL [1]. Because DLL is a first order control system, it is more stable and easier to design. Though these functions can also be performed with phase locked loops (PLLs), DLLs are often preferred due to their ease of design, better immunity to on-chip noise, and stability. To achieve low power operation of the DLL we will be using a modified delay cell in which a static current mirror will be converted into a dynamic current mirror. Due to this conversion there will be increase in jitter and the increased jitter can be reduced by selecting the proper phase detector from the compared ones in this paper. A Multiplexing DLL architecture which allows to switch between two different application specific DLLs is also presented here.

DLL will be using VCDL which does not accumulate jitter over many clock cycles. Therefore, DLL shows better jitter performance [2], [3]. In addition, DLL have smaller area and quicker locking time. The faster locking time enables the charge pump to stabilize, thereby reducing the power consumption further. Low power, short locking time, and low jitter are main focuses of the DLL design, which are obtained by combining the best of the components together. The paper is organized as follows: Section 2 describes the basic structure of DLL and gives overview of the delay cells architectures. Section 3 describes the circuit level implementations and simulation results. Section 4 gives some techniques for power

reduction and Section 5 gives some conclusions and future enhancements.

II. DLL BASICS

A DLL is basically a nonlinear negative feedback system. However, it is a common approach to illustrate a DLL by linear analysis. Although linear analysis is not able to produce a very precise result, it can still serve as a sensible first-order approximation [4] and can lead to some useful insights into a DLLs operation.

In a DLL, the input clock signal propagates through the VCDL and develops phase shift (or time delay) at every delay stage of the VCDL is shown in fig 1. The voltage of a loop filter controls the phase shift of each delay stage. The output is taken from one of the delay steps. A comparison is made between the phase of the reference clock and phase of output signal in the PD. The phase difference information generated by the PD (usually in the form of a voltage or a current) is then transferred to the CP [5]. The CP uses the phase error information to fine-tune the voltage of the loop filter and thus to change the delay of the delay line. Due to such a negative feedback mechanism, the phase error is steadily reduced until it finally becomes zero. At that moment, the delay of the whole VCDL line becomes equal to one clock cycle, and the voltage of the loop filter is steadied [6], which indicates that a locked state has been established.

There are essentially two types of DLL designs: analog and digital. The design choice is determined by several factors, including design complexity [8], layout size, noise levels of the system, process portability, and essential accuracy. A digital delay-locked loop uses digital devices to implement the variable delay-line. This means that the minimum change in the delay is some quantized step.

A digital delay cell makes up the minimum delay, and a sequence of delay cells are used to create the delay line. This minimum delay, or delay per stage [1], [10], [11], will be limited by the CMOS fabrication process of the device. However, due to the consistent nature of digital circuit performance across varying processes, a digital design may still be necessary.

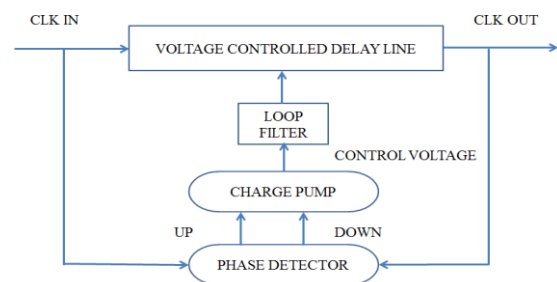


Figure 1: Block diagram of DLL

To reduce the static phase error incurred by using the quantized timing generated by a digital delay line, an analog delay line may be used. Due to the ability of an analog delay element to vary continuously, the jitter in the output is reduced. The overall topology remains the same; however, a charge-pump/loop filter combination is frequently used to control the analog delay line. The delay elements used to create the delay line are commonly found in voltage controlled oscillators (VCO's) typical for PLL design. They may range from a simple, current-starved inverter to a complex amplifier-based designs. Again, the selection depends on the environment in which the design will be used, the process tolerances, and the jitter requirements. Besides the advantage of reduced jitter in the output, an analog design may exhibit a better power-supply rejection ratio (PSRR), can occupy a reduced area in the layout [10], and consume less current than a digital DLL. However, to achieve these results, the design [8] is usually of higher complexity and requires a process-specific implementation, making it less portable to other processes.

III. DLL DESIGN

The proposed DLL-based frequency synthesizer is implemented in a 180nm CMOS technology. Our model has 4 sub modules which are Phase Detector (PD), Charge Pump (CP), Loop Filter (LP), and Voltage Controlled Delay Line (VCDL). An Edge Combiner can be used along with it for frequency multiplication, hence called as the Frequency Multiplier. The tenacity of this edge combiner is to accept low frequency signals and combine them to get a higher frequency signal.

A common mode feedback circuitry is used to control the common mode voltage of the capacitors because the current introduced by the charge pump is pumped to the capacitor. The speed of the charge pump is half of the common mode feedback speed [2], [13]. The design of the proposed synthesizer is based on the equations which administer the dynamics of the loop. This section describes the circuit level realization of different blocks.

A. Phase Detector

The main objective of a phase detector is to detect the phase difference between the reference signal and the feedback signal. The PD will measure the phase difference and will produce a corresponding phase offset pulse whose pulse width will be equal to that of the phase difference between reference and feedback signals. The PD will be producing UP and DOWN pulses, based on whether the reference clock leads or lags the feedback signal. When both the pulses are equal then both the UP and DOWN pulses will be zero.

Paper [15] presents the most commonly used linear type D-FF based phase detector. This Phase detector is simpler in design and has a stable operation. The drawbacks of this phase detector are its high power consumption, higher transistor count and when there is a change from phase lead to phase lag, vice versa, the D-FF will receive a reset pulse. This reset pulse will be a longer one and it will be having higher glitches that will produce a longer delay before the locking process begins. Also due to this reset delay it has a higher static phase error that characterizes the dead-zone problem. So this PD can be used in cases where stability and wider range of operation is preferred.

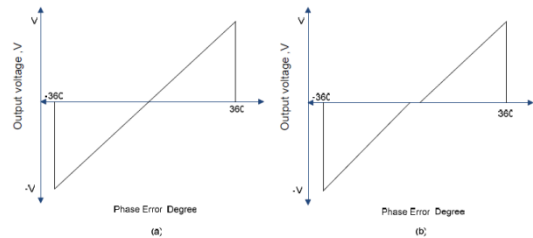


Figure 2: Dead-zone free vs Dead-zone presence in DLL

The dead zone problem arises when the PD fails to detect the minimum phase difference ie, the static phase error. The main reason for this problem is the reset delay of PD. Fig2 shows the cases of dead-zone free PD and a PD affected by dead-zone problem. This problem can be countered by reducing the overall delay of the PD or by reducing the delay in the reset path.

The next phase detector presented in paper [15] is a dynamic phase detector which mainly focuses on the glitch time and dead-zone problem in the previous PD. This PD does not use any gates in the feedback paths that are used for resetting. This reduces the delay in the reset path and thereby reduces the glitch time in the UP\DOWN signals. This PD also has the advantage of reduced transistor count and reduced power consumption. This PD also suffers from a dead-zone problem but the dead-zone is comparatively lower than the D-FF type PD.

The third Phase detector [16] compared here is GDI based PD. In this PD we will be using Gate diffused input(GDI) technique instead of the usual CMOS and PTL techniques. This GDI techniques enables faster and low power operation of the PFD. In this PFD we will be replacing a CMOS NOR gate with a GDI NOR gate that reduced the power consumption of the PFD. This PFD also suffers from a small dead-zone problem and it has to be noted that twin tub or SOI fabrication technique have to be used while implementing GDI technique. The final PD used is a High-speed PFD [17] which mainly focuses on speed of operation. In this PD the reset delay is totally eliminated by removing the feedback path. This almost eliminates the dead-zone problem providing only a phase error of 2ps which is negligible. Though this PD provides promising results, due to the lack of the feedback path, the stability of the PD becomes a bigger concern.

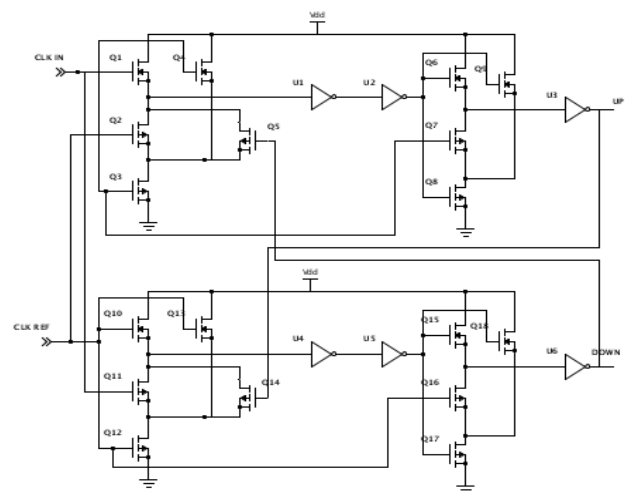


Figure 3: Dynamic PD

As a whole the second and third PD are most suitable for common low power applications whereas in cases where speed is the major concern, the fourth PD can be used.

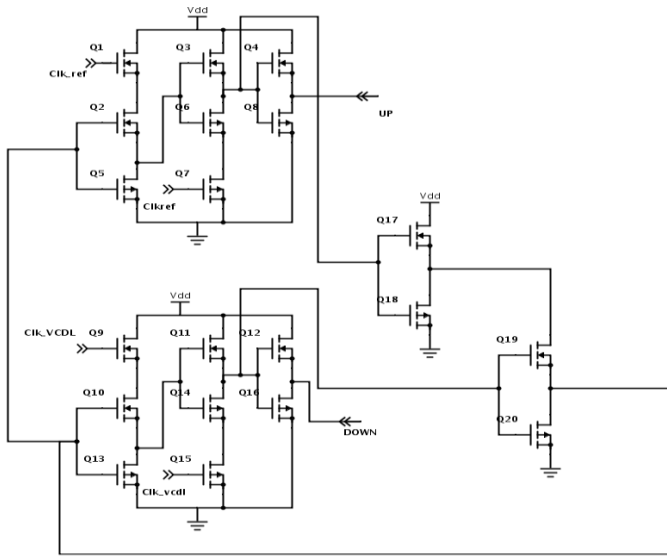


Figure 4: PD-GDI

The comparison table for the various PD are as follows

Table 1: Comparison of various PD

Phase Detector	Avg. Power Consumption	Glitch Period	Transistor Count	Static phase error	Stability
D-FF PD [15]	40.96uW	0.4ns	38	40ps	Stable
Dynami c PD[15]	32.5uW	0.2ns	30	20ps	Stable
PD-GDI[16]	19.3uW	0.2ns	20	10ps	Stable
High Speed PD[17]	14.68uW	0.04ns	18	2ps (negligible)	Lower Stability

B. Charge Pump

CP design is one of the most complicated parts of the DLL structure. The CP controls the charging/discharging current by UP/DN signal from PFD, and uses the phase difference between the up and down signals from the PFD to convert the phase error into current. Then, loop filter converts the current into the control voltage, by charging or discharging the capacitor and sending it to Vctrl to set the VCDL delay is shown in fig 5. Because of the small drop in voltage from input to output, the charge pump circuit supports a extensive range of supply voltages [13] and is ideal for powering high power circuitry. The basic purpose of Charge Pump is to convert the digital input signal to analog output signal. The charge Pump takes the digital output from the Phase Detector (PD) in the form of variable width pulses up and down signal and converts it to analog signal that is known as control voltage.

The combination of PFD-CP provides charging current error of less than 2 % over the entire range of output voltage swing under entire range of phase errors. It is shown later that 100 μA is appropriate value for I_{CP}. That means if the difference

between [12] rising edges of PFD’s input is 1 ns, the output voltage of CP will varied in a case that the time delay in VCDL to be 0.1ns. The general current equation for charge pump will be

$$I = I_{\text{pump}} \Delta\theta / 2\pi \dots\dots\dots (1)$$

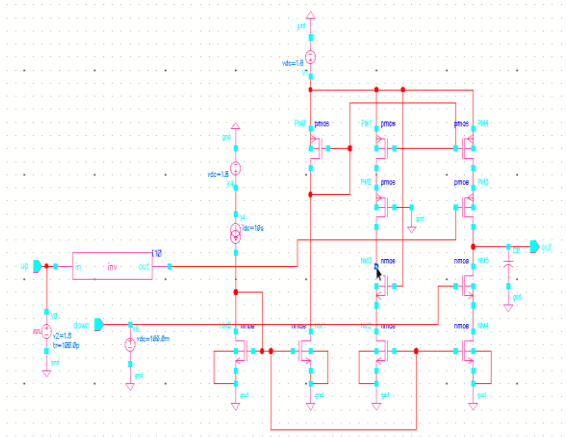


Figure 5: Charge pump

C. Voltage Controlled Delay Line

The biasing transistors have Vctrl as input from the charge pump and loop filter. This bias voltage wheels the current that passes through the inverter and controls the delay according to the control voltage [14]. As we had only one input from the loop filter so we have to design a method to invert the voltage to control the biasing transistors. A circuit to control the bias voltage with an added inverter is shown in fig.6. The two transistors copy the value of Vctrl to the PMOS of biasing transistors.

The most critical part of designing delay line is transistor sizing. At First we have to design a single inverter, so that it can provide 50% duty cycle for the positive edge and negative edge of the output. We have to adjust the transistor sizing to get the desired duty cycle and connected these transistors such that they are constantly in operation region.

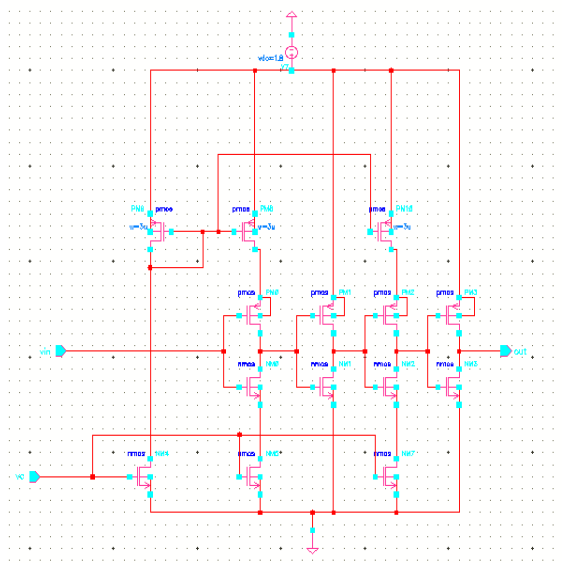


Figure 6: Delay cell

The delay produced by each stage of the delay element can be determined by using the formula

$$T_{\text{delay}} = V_{\text{sw}} (C / I_{\text{CP}}) \dots\dots\dots(2)$$

Where V_{sw} is the swing voltage of the inverter, C is the output load capacitance and I_{CP} is the charging/discharging current of the capacitor. We can realize I_{CP} as function of V_{ctrl} ,

$$I_{CP} = k * V_{ctrl} \dots\dots\dots(3)$$

Where k is a constant corresponding to the bias current. Given that the above condition is true, then T_{delay} can be realized as a non linear function of V_{ctrl}

$$T_{delay} = f * V_{ctrl} \dots\dots\dots(4)$$

As we were not able to get the required delay of about 0.25ns with a single inverter, we have to add two inverters. In the beginning one inverter stage is added and sized to get the 50% duty cycle. After correct sizing of the 2nd inverter stage, the next inverter stage is added to get inverted output of the reference signal.

Let the input frequency for the DLL be f_{in} and the initial phase of the DLL be ϕ_{in} . When the signal passes through first delay element then it will get phase shifted due to the delay produced. Let the phase change produced in the signal be $\Delta\phi_1$. So after crossing the first delay element the frequency will remain same as f_{in} whereas the phase gets shifted into $\phi_{in} + \Delta\phi_1$ and will get changed to $\phi_{in} + \Delta\phi_1 + \Delta\phi_2$ correspondingly.

This process will be continued until the output phase $\phi_{in} + \sum \Delta\phi_n$ will be equal to that of the input phase ϕ_{in} .

$$\phi_{in} + \sum \Delta\phi_n = \phi_{in} \dots\dots\dots(5)$$

In the case where there is static phase error, it will be included as σ . This σ contributes for the dead-zone problem of the DLL

$$\phi_{in} + \sum \Delta\phi_n + \sigma = \phi_{in} \dots\dots\dots(6)$$

Combining all the blocks together we get the waveforms for DLL, shown in fig 7

The Figure of merit of the DLL can be obtained by using the equation 7

$$FOM = 10 \log ((L * P) / (f_0^2 * N^2)) \dots\dots(7)$$

Where L is the Phase Noise, P is the average power consumption, f_0 is the output frequency and N is the number of delay stages.

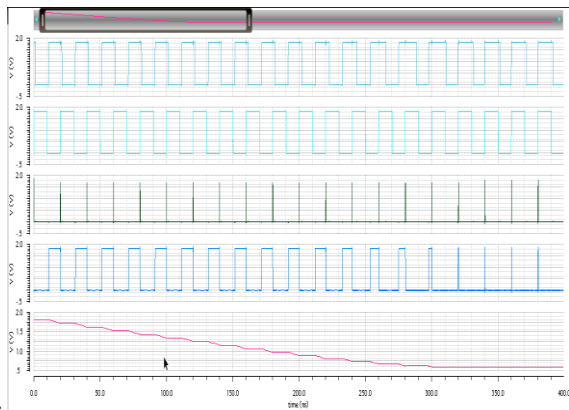


Figure 7: DLL waveform

D. Power Reduction Using Modified Delay Cell

The delay cell is the most repeating component in a delay cell. So reducing the power usage in each delay cell will considerably reduce the overall power consumption of the DLL. Generally the normal delay cell will be having a current mirror that will be having the branch B1. There will be a constant flow of current through this branch throughout the

operation of the circuit. This leads to power loss. To reduce this power loss we will be using a transistor Q13 in the branch B1. The gate terminal of the transistor Q13 will be connected to the input signal to the delay cell. So whenever the input is available, only at that time the current in branch B1 will be established. This reduces the power consumption of the circuit since it can save current upto 100uA.

But there are some shortcomings in this type of delay cell. The major one is the jitter. When the switch transistor is introduced in the branch B1, it will make the static current mirror into a dynamic one. This leads to the constant switching in voltage at the source of the biasing PMOS of the current starved delay element. This switching process leads to the production of jitter in the output signal. This jitter can be reduced by using the PD which provides immunity against the jitter. Dynamic PD and PD-GDI are well suited for this purpose.

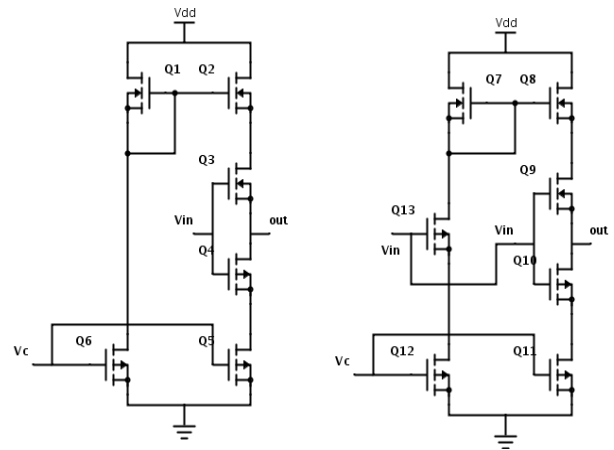


Figure 8: Static vs Dynamic Delay cell

Design corner	Modified Static delay cell	Modified Dynamic delay cell
SS	3.96mW	0.11mw
SF	5.4mW	0.17mW
FS	4.24mW	0.12mW
FF	5.96mW	0.18mW

Table 2 Average Power consumption of static and dynamic delay cell at various design corners

E. Edge Combiner

The edge combiner can be used to achieve frequency multiplication. To achieve the multiplication factor of four, we want eight different phases from the delay line. The frequency multiplier is needed to combine these phases in such a way that we get the frequency four times the input frequency i.e. the reference frequency is 100 MHz and the output frequency is 400 MHz. The delay of every individual phase is 0.5ns. In order to design a frequency multiplier we will be using six XOR and a single OR gate. There should be a 0.5 ns delay between each clock phase. An XOR circuit is fed with phase 1 and phase 2, similarly phase 3 with 4, phase 5 with 6, phase 7 with 8 were fed into next XOR circuits respectively.

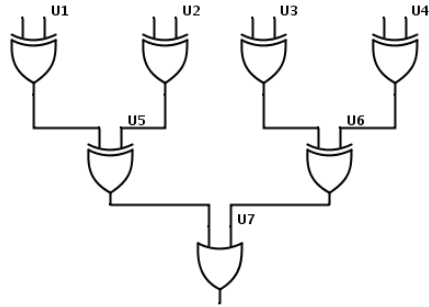


Figure 9: Edge combiner

In the second stage the XOR output of the first and second gates will be feed into first XOR gate and third and fourth into XOR of the second stage correspondingly. Lastly in order to get the output the second stage outputs are fed to OR gate to get the four times frequency of the input signal. When DLL is in locked condition, the edge combiner combines the edges of the delay stages outputs to provide an integer multiple of the input frequency.

F. Multiplexing Architecture For Dll

Two different DLLs are designed here. One by using Dynamic PD along with Static delay cell. This combination provides a low jitter performance while having a moderate power consumption. On the other hand the second DLL is formed by using PD-GDI along with dynamic delay cell. This combination provides a ultra low power operation while having a trade-off for jitter performance. The block diagram of both these DLLs combined together is shown in Fig 10.

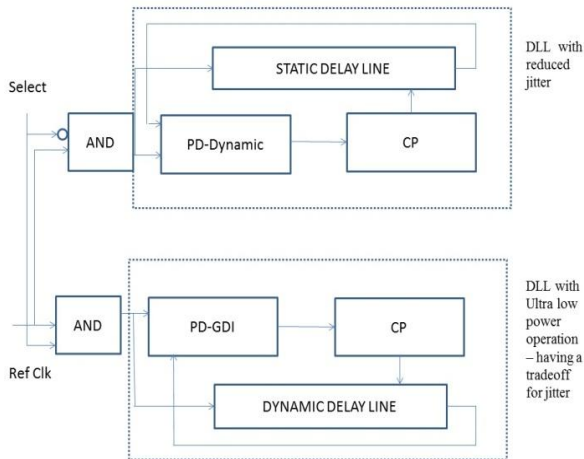


Figure 10: DLL – Multiplexing Architecture

In this architecture the select signal is used to switch between the two different DLLs , Switching the select signal will make the reference clock available only to one of two available DLLs. So as per the user requirements, the DLLs can be switched to achieve the desired performance.

The DLL using the static and Dynamic delay cell are analysed here and their characteristics are tabulated and compared.

Table 3: Comparison of DLL – Using Static and Dynamic Delay cell

Parameters	DLL[18]	DLL – Static delay cell	DLL – Dynamic delay cell
Supply Voltage	1.8V	1.8 V	1.8 V

Locking Time	400ns	350ns	420ns
Frequency	84-800MHz	40-333 MHz	40-333 MHz
Average Power Consumption	5mW	2.82 mW	0.78 mW
Phase noise	-	-155.2 db/Hz	-157.9 db/Hz
Noise figure	-	70.46 db	69.9 db
Periodic jitter	238fs at 800MHz	101.4 fs at 100 MHz	663.9 fs at 100 MHz
Figure of Merit	-	127	73
Compatible Phase Detector	Modified D-FF type PFD	Dynamic PFD and PFD - GDI	Dynamic PFD and PFD - GDI

CONCLUSION

A DLL with Low power Consumption and good figure of merit is presented here. A comparative approach for the design of DLL with options of various phase detectors , each one having their own advantages is also presented here. The PD can be selected as per our design Considerations. First the design of a Static delay cell is given and then the static delay cell is modified into a Dynamic delay cell to achieve reduced power consumption while having a trade-off of increased Jitter. The Locking time is also kept low to reduce power loss since it switches off CP faster. Then a multiplexing architecture is presented here, enabling to switch between two different application specific DLLs.

ACKNOWLEDGEMENT

The authors are grateful to the lab facility at Electronics and Communication Engineering department of RMK Engineering college, Chennai which provided excellent support to finish the work successfully.

References

- [1] Mohammad Gholami “A Novel Low Power Architecture for DLL-Based Frequency Synthesizers”, Circuits System Signal Process (2013) 32:781–801
- [2] M. Gholami, Gh. Ardeshir, H. Ghonoodi, “A novel architecture for low voltage-low power DLL-based frequency multipliers” IEICE Electron. Express 8(11), 859–865 (2011)
- [3] C.-H. Kuo, H.-J. Lai, M.-F. Lin “A multi-band fast-locking delay-locked loop with jitter-bounded feature” IEEE Trans. Ultrason. Ferroelectr. Freq. Control 58(1), 51 59 (2011)
- [4] C.N. Chuang, S.I. Liu “A 0.5–5 GHz wide-range multi-phase DLL with a calibrated charge pump” IEEE Trans. Circuits Syst. II, Express Briefs 54(11), 939 943 (2007)
- [5] B. Razavi “Design of Analog CMOS Integrated Circuits” (McGraw-Hill, New York)

- [6] Kuo-Hsing Cheng, Shu-Ming Chang, Yu-Lung Lo, and Shu-Yu Jiang, "A 2.2 GHz Programmable DLL-Based Frequency Multiplier for SOC Applications," in Proc. of the 2004 Fourth IEEE AP-ASIC, Aug. 4-5, 2004, Fukuoka, Japan.
- [7] Sebastian Hoyos, Cheongyuen W. Tsang, Johan Vanderhaegen, Yun Chiu, Yasutoshi Aibara, Haideh Khorramabadi, and Borivoje Nikolic "A 15 MHz to 600 MHz, 20 mW, 0.38 mm Split-Control, Fast Coarse Locking Digital DLL in 0.13 m CMOS" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 20, NO. 3, MARCH 2012
- [8] Tai-Cheng Lee, Member, IEEE, and Keng-Jan Hsiao "The Design and Analysis of a DLL-Based Frequency Synthesizer for UWB Application" IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41, NO. 6, JUNE 2006
- [9] H.-H. Chang, J.-W. Lin, and S.-I. Liu, "A Fast Locking and Low Jitter Delay- Locked Loop Using DHDL" IEEE J. Solid-State Circuits, vol. 38, pp. 343-346, Feb.2003.
- [10] M. Mansuri and C.-K. K. Yang, "Jitter Optimization Based on Phase-Locked Loop Design Parameters" IEEE J. Solid-State Circuits, vol. 37, pp. 1375-1382, Nov.2002.
- [11] Cicero S. Vaucher, Senior Member, IEEE, and Bram Nauta, Member, IEEE "Low-Jitter Clock Multiplication: A Comparison Between PLLs and DLLs" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING, VOL. 49, NO. 8, AUGUST 2002
- [12] G. Chien, "Low-noise local oscillator design techniques using a dll based frequency multiplier for wireless applications," Ph.D. dissertation, Univ. California, Berkeley, 2000.
- [13] W. B. Wilson, U.Moon, K. R. Lakshmikummar, and L. Dai, "A CMOS self-calibrating frequency synthesizer," IEEE J. Solid-State Circuits, vol. 35, pp. 1437-1444, Oct. 2000.
- [14] G. Chien, P. Gray, "A 900-MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications". IEEE J. Solid-State Circuits 35, 1996-1999 (2000)
- [15] Shirish Tripathi " Design of Low glitch dynamic phase detector for Delay locked loops". International journal of emerging science and engineering (IJESE), ISSN:2319-6378, Volume-1, Issue-8, June 2013.
- [16] Raj Nandini, Himadri singh raghav, B.P Singh " Design of low power phase frequency detectors for Delay locked loops". International journal of emerging science and engineering (IJESE), ISSN:2319-6378, Volume-1, Issue-5, March 2013.
- [17] Raj Nandini, Himadri singh raghav " Comparison of low power phase frequency detectors for Delay locked loops". International journal of Engineering research and technology (IJERT) , ISSN:2278-0181, Vol-2 Issue1, January-2013
- [18] Himadri singh raghav, B.P singh , " Design of Low power Low jitter DLL tested at all 5 design corners to avoid false locking " IEEE- ICSE2012, Kulalampur, Malaysia.