A mixed PLL/DLL architecture for low jitter clock generation

Conference Paper · January 2004

DOI: 10.1109/ISCAS.2004.1329122 · Source: DBLP

Gu-Yeon Wei
Harvard University

A MIXED PLL/DLL ARCHITECTURE FOR LOW JITTER CLOCK GENERATION

Yong-Cheol Bae and Gu-Yeon Wei

Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138

ABSTRACT

This paper presents a mixed PLL/DLL architecture for low-jitter clock generation that merges phase-locked loop (PLL) and delay-locked loop (DLL) characteristics. It relies on an interpolator to configure the loop to operate more like a PLL or more like a DLL depending on the interpolator's settings. The ability to vary interpolator settings enables wide range control of the clock generator's loop bandwidth. Therefore, the loop bandwidth can readily be adjusted to accommodate different noise conditions. A discrete-time Z-domain analysis is provided to illustrate the noise filtering characteristics of the loop in the presence of various noise sources and highlights the potential advantages of the mixed PLL/DLL architecture. Simulation results verify stable operation of the loop designed for a 0.18μm CMOS process.

1. INTRODUCTION

Low-jitter clock generation is required for reliable operation of high-speed synchronous designs such as microprocessors and data communication links. A clock signal sets the timing reference with respect to which computation and propagation of synchronous data occurs. Variations in this timing reference (i.e., clock jitter) requires designs to incorporate additional margins that degrade performance and can cause bit errors in communications systems. In this paper, we focus on two types of noise sources that can cause clock jitter in integrated clock generators – noise on the input reference clock and on-chip noise sources (e.g., thermal, device, power supply, and substrate noise). Depending on the particular application, the relative magnitudes of these noise sources can vary. For example, complex digital systems can inject appreciable amounts of switching noise into the power supply and substrate. Therefore, it is desirable to build a clock generator that can accommodate a wide variety of noise conditions in efforts to minimize clock jitter.

Of course, designers of modern clock generators strive to minimize the effects of all noise sources on the final output clock through careful circuit design [1] and by appropriately tuning the clock generators' loop bandwidths [2]. We build upon these circuit-level advances with a modified architecture to further mitigate the effects of noise on on-chip clock jitter. Clock frequency synthesis (clock multiplication) used to be reserved for phase-locked loops (PLLs), but several researchers have shown that clock multiplication can also be achieved using delay-locked loops (DLLs) [1, 3, 4, 5]. As a result, there has been some question as to whether PLLs or DLLs are better suited for low-jitter clock generation. One can argue that PLLs are better suited for rejecting noise in the input reference clock and DLLs are better suited for rejecting on-chip noise.

In order to accommodate a wide range of noise conditions, this paper presents a mixed PLL/DLL (MP/DLL) architecture that merges the characteristics of the two loops into a single clock generator and has the ability to operate either as a PLL, a DLL, or a combination of the two. The paper proceeds with a section that describes the overall architecture of the MP/DLL and highlights key components that make this merging possible. In order to understand the filtering characteristics of the loop under various noise conditions, a Z-domain analysis of the loop is provided in Section 3. Section 4 then presents a detailed circuit description of the MP/DLL with HSPICE simulation results that verify stable operation.

2. MP/DLL ARCHITECTURE

A block diagram of the MP/DLL architecture is presented in Fig. 1. It resembles the architecture presented in [5], which is a 500MHz clock frequency synthesizer that can be configured to operate either as a multiplying PLL or a multiplying DLL. The ability to switch between the two modes of operation is determined by whether the delay-elements are configured as a ring oscillator or a delay line via a multiplexer. The main difference between that design and the proposed MP/DLL is that the multiplexer has been replaced with an interpolator. Depending on the interpolation weight setting, the clock generator loop can operate with merged characteristics of both a PLL and a DLL.

The operation of this MP/DLL is like any conventional PLL or DLL. Tracing through the loop in Fig. 1, a phase-

0-7803-8251-X/04/$17.00 ©2004 IEEE

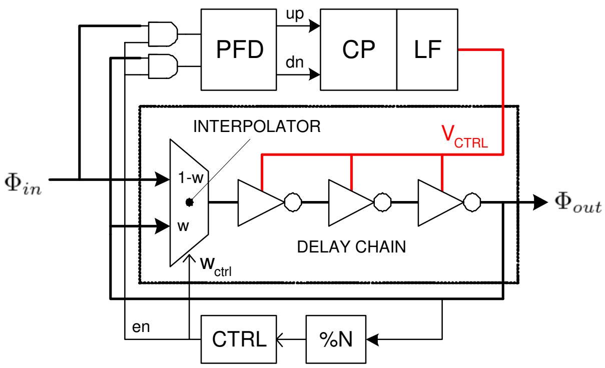


Fig. 1. Mixed PLL/DLL block diagram.

frequency detector (PFD) compares the edges of the input reference clock, Φ_{in}, and the output clock, Φ_{out}. The non-zero up and down pulses out of the PFD (to avoid deadband) feed into a charge pump (CP) and loop filter (LF) to integrate the detected phase error. The loop filter consists of a resistor and capacitor, which add a zero for stable PLL operation. The resulting control voltage, V_{ctrl}, out of the loop filter sets the delay for each of the delay elements and locks Φ_{out}'s frequency and phase to Φ_{in}. In order to accommodate clock multiplication, a divider circuit works in concert with control circuitry to pass appropriate edges to the PFD and control the interpolator.

Configuration of the delay elements determines whether the loop acts like a PLL, a DLL, or a combination of the two, which are all possible through different settings of the interpolator. When $w = 1$, the clock edge, Φ_{out}, from the end of the delay chain feeds back around to the beginning. The input reference clock edge sees a weighting of 0 and, therefore, does not contribute any energy to the input of the chain. Hence, the delay elements are configured as a ring oscillator and the overall loop essentially operates like a PLL. When $w = 0$, the opposite case occurs. The signal at the end of the delay chain does not contribute any energy to the beginning. Instead, only the input reference clock edge, Φ_{in}, passes through. Hence, the delay elements implement a delay line and the loop is a DLL. When $0 < w < 1$, energy from both the input reference clock edge, Φ_{in}, and Φ_{out} feed into the input of the delay elements. Therefore, the resulting loop is a mixture of a PLL and DLL.

It is important to point out one subtle aspect of the loop to clarify how it can operate either as a PLL or a DLL. In DLL mode, a rising transition at Φ_{in} exits the delay chain at Φ_{out} as a falling edge transition, but two passes through the delay chain are required to complete a full clock cycle. So, in order to share the PFD (that compares rising clock edges) for both PLL and DLL modes of operation, the interpolator must fully switch (i.e., $w = 1$) to pass the falling edge every half cycle when there is no clock multiplication. For clock multiplication, multiple consecutive edges recirculate through the delay line. The divider ($\%N$ and control circuity ($CTRL$) coordinate this process. The $CTRL$ block sends an enable signal to selection gates to pass the appropriate rising edges to the PFD and phase error detection is only done at the rising edge of every reference cycle.

3. Z-DOMAIN ANALYSIS

In order to understand and elucidate the noise filtering characteristics of the MP/DLL, this section presents a simple 2^{nd}-order Z-domain analysis of the loop.[1] From this analysis, we can see how the MP/DLL enables us to easily adjust the loop bandwidth of the clock generator by adjusting interpolator weighting. For this analysis, it is important to use a Z-domain analysis since a linearized S-domain analysis does not properly capture the input-to-output phase transfer function (PTF) nor the noise-to-output transfer function (NTF) of a DLL [6].

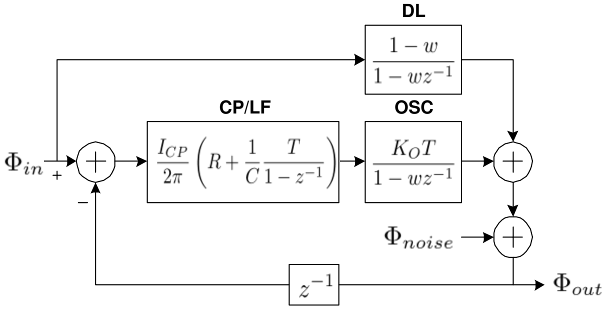

DL
$\frac{1-w}{1-wz^{-1}}$
CP/LF
OSC
$\frac{I_{CP}}{2\pi}\left(R+\frac{1}{C}\frac{T}{1-z^{-1}}\right)$
$\frac{K_O T}{1-wz^{-1}}$
Φ_{in}
+
−
Φ_{noise}
z^{-1}
Φ_{out}

Fig. 2. Z-domain block diagram of MP/DLL.

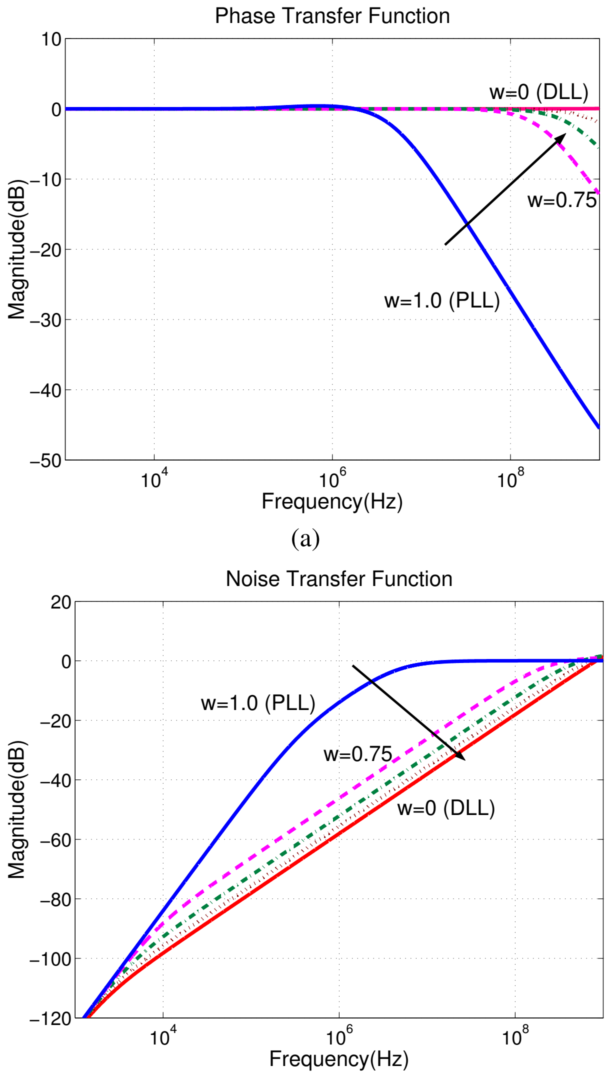
Fig. 2 illustrates a block diagram of the Z-domain model for the MP/DLL. The model again resembles that for a PLL, but with an additional feed-forward block that adds the contribution of the reference clock phase through the delay line to the output phase. Note that the interpolation weighting factor, w, scales the relative phase contributions of the delay line and oscillator. When $w = 1$, the model simplifies to one for a PLL and it simplifies to a DLL model with $w = 0$. The resulting PTF (ϕ_{out}/ϕ_{in}) and NTF (ϕ_{out}/ϕ_{in}) are as follows:

$$\frac{\phi_{out}(z)}{\phi_{in}(z)} = \frac{\left[K'\left(1+\frac{T}{RC}\right)+1-w\right]z - K' - 1 + w}{z^2 + \left[K'\left(1+\frac{T}{RC}\right)-w-1\right]z + w - K'} \quad (1)$$

$$\frac{\phi_{out}(z)}{\phi_{noise}(z)} = \frac{z^2 - 2z + 1}{z^2 + \left[K'\left(1+\frac{T}{RC}\right)-w-1\right]z + w - K'} \quad (2)$$

where $K' = K_{CP}K_OTR$. K_{CP} encompasses the CP current and PFD gain, T is the cycle time, R is the loop-filter resistor, C is the loop-filter capacitor, and K_O is the oscillator gain.

[1] A 3^{rd}-order Z-domain analysis has also been done, but omitted due to space constraints. Moreover, the general characteristics of the loop are sufficiently captured by the 2^{nd}-order analysis.



(a)

(b)

Fig. 3. Phase (a) and Noise (b) transfer functions

Fig. 3 plots the resulting PTF and NTF for different interpolation weighting values ($w = 0, 0.25, 0.5, 0.75, 1$). When $w = 1$, the PTF exhibits low-pass filter characteristics and the NTF exhibits high-pass filter characteristics expected of a PLL, where the corner frequencies for both occur at the same frequency. As w decreases, the PTF bandwidth increases and becomes an all-pass for a DLL. Similary, the high-pass corner frequency of the NTF increases and the loop rejects more (high frequency) noise. These plots are consistent with the common understanding of the relative advantages and disadvantages of PLLs and DLLs in the presence of different noise sources. One advantage of the MP/DLL is that it provides an easy way to change the bandwidth of the loop over a wide range (multiple orders of magnitude), which would be difficult to achieve by changing loop parameters in a conventional PLL [2].

We have also investigated the MP/DLL's noise filtering properties with a discrete time-sampled model of the loop that has been coded in Matlab. Fig. 4 presents a summary of several simulation runs and plots the resulting output clock's RMS and peak-to-peak jitter versus interpolation weighting. The simulations include white and colored noise spectra added to both Φ_{in} and Φ_{noise}. The input reference clock's RMS and peak-to-peak jitter are also shown on the plot. For this particular noise condition, setting the interpolation weight close to $w = 0.6$ results in the minimum jitter, since the loop is able to reject both input and on-chip noise sources. When $w = 0$, the output jitter is dominated by input noise. When $w = 1$, the output jitter is dominated by the on-chip noise accumulated by the oscillator.

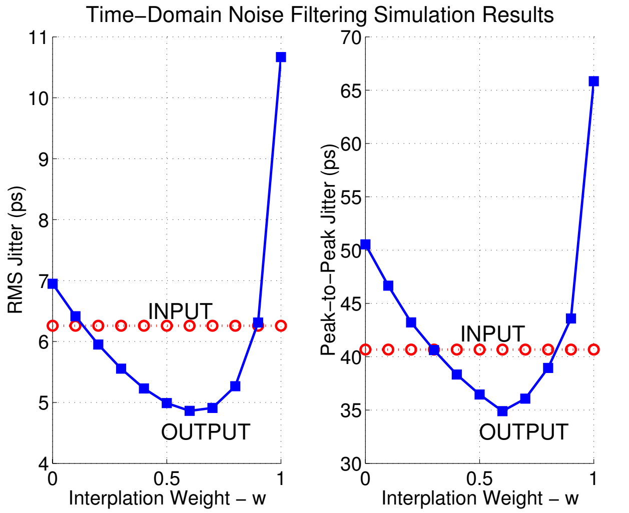


Fig. 4. RMS and peak-to-peak clock jitter versus interpolation weighting.

4. CIRCUITS

The MP/DLL has been designed in a 0.18-μm CMOS process, operating off of a 1.8-V supply. Extensive HSPICE simulations verify stable operation of the loop in all three modes - PLL, DLL, and mixed ($w = 0.5$). The design utilizes a glitch latch based PFD, current-controlled delay elements, and a voltage-to-current converter (VIC) similar to those found in [2]. A differential CP and LF are used in efforts to create symmetric paths for the up and down pulses out of the PFD. As a result, the differential output of the loop filter must be converted into a single-ended signal. To do this, a self-biased scheme similar to the approach in [5] is used. Fig. 5 presents a detailed schematic of the differential CP, LF, and VIC. With the aid of an operational transimpedance amplifier (OTA), the differential output of the LF is integrated onto a capacitor to set a coarse voltage,

V_{main}, which sets the nominal control voltage level that sets the current to the delay elements, $I_{ctrl} \propto K_{scale} V_{main}$. While the time constant to set V_{main} is large, minor (higher speed) adjustments are made through the differential output of the loop filter, which proportionally skews I_{main}. In lock, the low-bandwidth loop through the OTA ideally drives V_{main} to a level such that the differential output of the CP and LF is driven to zero. This has the benefit of minimizing the finite output impedance effects of the CP. While not shown in the figure, an additional low-bandwidth loop is included to compensate for potential device mismatches in the CP that can exacerbate jitter [5].

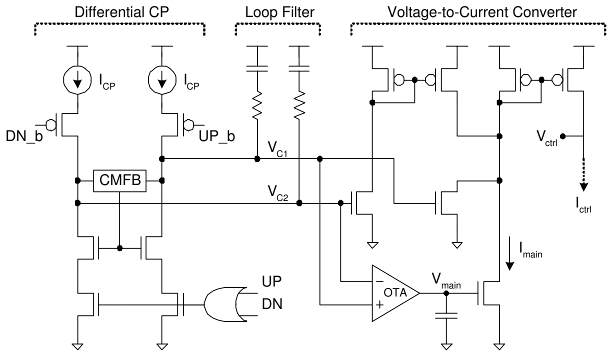


Fig. 5. Detailed circuit schematics.

Fig. 6 presents transient simulation results of the proposed MP/DLL iterating through three modes of operation (by changing w) locking to a 900-MHz input reference. The figure illustrates locking transients of V_{ctrl} and V_{main}, and the phase error between Φ_{out} and Φ_{in}.

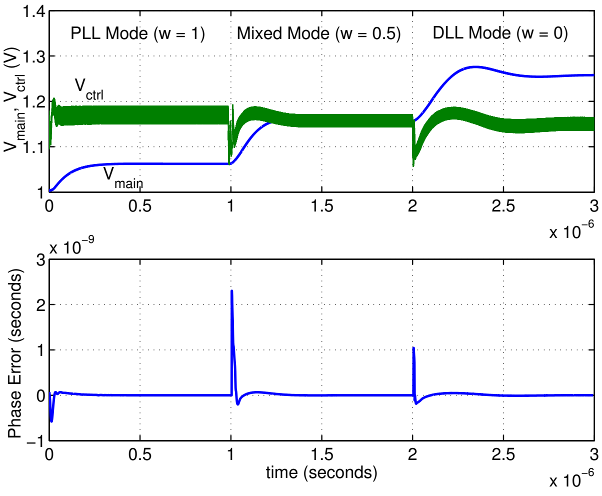


Fig. 6. Transient locking simulations (HSPICE).

5. CONCLUSIONS

This paper has presented a mixed PLL/DLL architecture to build a low-jitter clock generator that has the ability to operate as a PLL, a DLL, or a combination of the two. The ability to gradually shift from one mode of operation to another has the effect of changing the bandwidth of the loop over a wide range. Depending on the noise conditions of each particular application, the interpolation weighting (i.e., loop bandwidth) can be set to minimize jitter. The physical layout of the MP/DLL design is currently being completed and will be fabricated and tested in order to verify the Z-domain analysis and simulations with experimentally measured results. The next step will be to design a block that can monitor the on-chip jitter magnitude and automatically adapt the interpolation weighting to the lowest jitter setting.

6. REFERENCES

[1] J.G. Maneatis, “Low-jitter process-independent dll and pll based on self-biased techniques,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1723–1732, November 1996.

[2] M. Mansuri and C.-K.K. Yang, “Jitter optimization based on phase-locked loop design parameters,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1375–1382, November 2002.

[3] R. Farjad-Rad, W. Dally, H.-T. Ng, R. Senthinathan, M.-J.E. Lee, R. Rathi, and J. Poulton, “A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1804–1812, December 2002.

[4] A. Waizman, “A delay line loop for frequency synthesis of de-skewed clock,” in *IEEE Solid-State Circuits Conference Digest of Technical Papers*, February 1994, pp. 298–299.

[5] G.-Y. Wei, J. Stonick, D. Weinlader, J. Sonntag, and S. Searles, “A 500-MHz MP/DLLclock generator for a 5Gb/s backplane transceiver in 0.25μm CMOS,” in *IEEE Solid-State Circuits Conference Digest of Technical Papers*, February 2003, pp. 464–465.

[6] M.-J.E. Lee, W. Dally, T. Greer, H.-T. Ng, R. Farjad-Rad, J. Poulton, and R. Senthinanthan, “Jitter transfer characteristics of delay-locked loops – theories and design techniques,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 4, pp. 614–621, April 2003.