Advantages of Shift Registers Over DLLs for Flexible Low Jitter Multiphase Clock Generation

Xiang Gao, *Student Member, IEEE*, Eric A. M. Klumperink, *Senior Member, IEEE*, and Bram Nauta, *Fellow, IEEE*

***Abstract*—In this paper, we compare a shift register (SR) to a delay-locked loop (DLL) for flexible multiphase clock generation, and motivate why a SR is not only more flexible but often also better. For a given power budget, we show that a SR almost always generates less jitter than a DLL, assuming both are realized with current-mode logic. This is due to differences in jitter accumulation and the possibility to choose latch delays in a SR much smaller than the delays of DLL elements. For N-phase clock generation, a SR also functions as a divide-by-N and requires a voltage-controlled oscillator with N times higher frequency. However, this does not necessary lead to more power consumption and can even have advantages like higher Q and less area for the inductors.**

***Index Terms*—Clock generation, multiphase clocks, current-mode logic (CML), delay-locked loop (DLL), divider, jitter, timing jitter, phase noise, shift register (SR).**

I. Introduction

MULTIPHASE clocks are useful in many applications, e.g., in high-speed serial links [1] to process data streams at a bit rate higher than the clock frequency, and in time-interleaved analog–digital converters (ADCs) [2]. In wide-band wireless communication systems, harmonic rejection mixers and multipath polyphase circuits need multiphase clocking to reject unwanted harmonics and sidebands [3]. Aiming for multifunctionality (e.g., software defined radio), we would like a flexible multiphase clock generator (MPCG) to adapt to largely different data rates, sampling rates or radio frequencies.

To implement a MPCG, both delay-locked loops (DLLs) and shift registers (SRs) are used. A SR MPCG also functions as a divide-by-N divider for N-phase clock generation. Although a SR MPCG seems more attractive due to its wide working frequency range (flexibility), it requires an N times higher clock frequency and at first glance seems to consume more power. However, a SR MPCG doesn't have jitter accumulation from one clock phase to the other as in a DLL equivalent, which should be taken into account for a fair comparison. This paper aims to make a solid comparison between these two MPCGs, primarily based on their power and absolute output jitter performance. A part of this work, related to thermal noise jitter was presented earlier in [4]. This paper also analyses deterministic jitter due to mismatch, and jitter transfer characteristics. Furthermore, flexibility aspects relevant for multifunctionality will be discussed.

This paper is arranged as follows. Section II describes the architecture of a DLL MPCG and analyses its jitter performance, while Section III addresses the SR. Section IV makes a comparison and Section V verifies the analysis via simulation results, while Section VI presents conclusions.

Manuscript received July 10, 2007; revised November 8, 2007. This paper was recommended by Guest Editor A. Tasic.

The authors are with the IC-Design Group, CTIT, University of Twente, 7500 AE Enschede, The Netherlands (e-mail: X.Gao@utwente.nl).

Digital Object Identifier 10.1109/TCSII.2008.918972

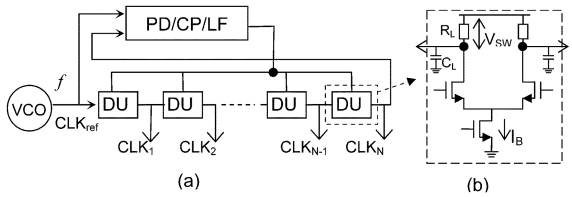

Fig. 1. (a) DLL MPCG architecture. (b) CML delay unit schematic.

II. DLL MPCG Jitter

A. DLL MPCG Architecture

The architecture of a DLL MPCG is shown in Fig. 1(a). It consists of a voltage-controlled delay line (VCDL) which has N identical delay units (DUs) and a control loop consisting of a phase detector (PD), a charge pump (CP) and a loop filter (LF). In the DLL, a reference clock $\mathrm{CLK_{ref}}$, generated by a voltage-controlled oscillator (VCO) with a frequency of f, is propagated through the VCDL. The loop compares the phase of the last output of the VCDL with $\mathrm{CLK_{ref}}$ and controls the VCDL so that its total delay time is one reference clock period. Once locking is achieved, the N outputs $\mathrm{CLK}_1 \sim \mathrm{CLK}_N$ are multiphase clocks with $2\,\pi/N$ phase spacing.

B. DLL MPCG Output Jitter

The DLL MPCG output jitter can be divided into three parts: 1) jitter transferred from the reference clock; 2) jitter generated by the VCDL and 3) jitter from the control loop. The jitter of the reference clock is transferred to the DLL outputs with some jitter peaking [5], [6]. The DLL cannot decrease reference clock jitter, but jitter peaking can be made very small by choosing a low DLL loop bandwidth [5], [6]. For an optimal DLL design, the jitter contribution of the control loop is negligible [5] and hence ignored hereafter. Thus, VCDL jitter is our main worry.

In a DLL MPCG, the VCDL generates two types of jitter: random noise jitter caused by *thermal noise* and deterministic mismatch jitter due to *mismatch* of the delay units. The DLL renders no improvement of VCDL noise jitter. Again, the VCDL noise jitter is lowest for low values of the loop bandwidth, in which case it would be almost equal to that of a free-running VCDL [5]. The jitter will thus accumulate from one delay unit to the other. If the noise jitter variance of one delay unit is $\sigma^2_{t,\mathrm{DU,noise}}$, and we assume uncorrelated white noise, the noise jitter variance on the output of the nth delay unit will be n times bigger. For multiphase clock applications like the software defined radio transmitter in [3], the jitter of every clock phase is equally relevant. To quantify the jitter of a set of N-phase

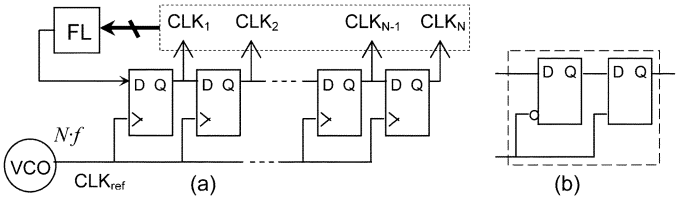

Fig. 2. (a) SR MPCG architecture. (b) DFF block schematic.

clocks, the averaged jitter variance of the N clocks is a meaningful quantity. The average noise jitter variance generated by the DLL can be calculated as

$$\left(\sigma^2_{t,\mathrm{DLL,noise}}\right)_{\mathrm{avg}N} = \frac{1}{N}\cdot\sum_{n=1}^{N} n\cdot\sigma^2_{t,\mathrm{DU,noise}} = \frac{N+1}{2}\sigma^2_{t,\mathrm{DU,noise}}. \quad (1)$$

Different from noise jitter, the DLL loop *can* improve the deterministic mismatch jitter. The start and end of the VCDL are both aligned to the reference clock and thus have zero deterministic time error. The maximum mismatch jitter appears at the middle of the VCDL. If we define the mismatch jitter variance of one delay unit as $\sigma^2_{t,\mathrm{DU,mis}}$, the jitter variance on the output of the nth delay unit can be calculated as [5]

$$\sigma^2_{t,\mathrm{DU}_n,\mathrm{mis}} = \frac{n(N-n)}{N}\sigma^2_{t,\mathrm{DU,mis}}. \quad (2)$$

The average mismatch jitter variance generated is then

$$\left(\sigma^2_{t,\mathrm{DLL,mis}}\right)_{\mathrm{avg}N} = \frac{N^2-1}{6N}\sigma^2_{t,\mathrm{DU,mis}} \overset{N^2\gg 1}{\approx} \frac{N}{6}\sigma^2_{t,\mathrm{DU,mis}}. \quad (3)$$

III. SR MPCG Jitter

A. SR MPCG Architecture

The architecture of a SR MPCG is shown in Fig. 2(a). It consists of a D flip-flop (DFF) chain with N identical DFFs. A reference clock $\mathrm{CLK_{ref}}$, generated by a VCO with a frequency $N\cdot f$, is fed into the DFF chain. A flip logic (FL) circuit monitors the N outputs of the DFF chain and flips the logic value at the D input of the first DFF twice every N reference clock cycles. In other words, the outputs of the DFF chain run at a frequency of f and the SR based MPCG also functions as a divide-by-N divider. Since a DFF is sensitive to rising edges, the Q output of each DFF is delayed from the previous DFF's output by one reference clock period, which is equivalently a $2\ \pi/N$ phase delay. In this way, N-phase clocks $\mathrm{CLK}_1 \sim \mathrm{CLK}_N$ are generated. Depending on different implementations of the flip logic, the duty cycle of the N -phase clocks can theoretically vary from $1/N$ to $(N-1)/N$. For example, if 18-phase clocks with a 1/3 duty cycle are wanted, the flip logic can simply be a NOR-gate with CLK_6 and CLK_{12} as its inputs [3]. This gives the SR based MPCG extra flexibility.

B. SR MPCG Output Jitter

The SR MPCG output jitter can be divided into two parts: jitter transferred from the reference clock and jitter generated by the DFF chain. The flip logic is simply a logical "enabler" for the first DFF and will not contribute to jitter.

For the jitter transferred from the reference clock, the SR MPCG renders no improvement. Any timing error at the reference clock will be transferred to the DFF chain outputs.

Similar to the VCDL, the DFF chain also generates two types of jitter: noise jitter and mismatch jitter. However, there is *no jitter accumulation* from one DFF to the other, since each DFF output only acts as an "enabler" for the next DFF, while the VCO defines the timing. A DFF can be designed with two master/slave latches as shown in Fig. 2(b). For a proper design, only the second latch contributes to jitter since the first is just an "enabler." If we define the rms noise and mismatch jitter variance of one latch as $\sigma^2_{t,\mathrm{Latch,noise}}$ and $\sigma^2_{t,\mathrm{Latch,mis}}$ respectively, the average jitter variance for the set of N-phase clocks generated by the SR can be easily calculated as

$$\left(\sigma^2_{t,\mathrm{SR,noise}}\right)_{\mathrm{avg}N} = \frac{1}{N}\cdot\sum_{n=1}^{N}\sigma^2_{t,\mathrm{Latch,noise}} = \sigma^2_{t,\mathrm{Latch,noise}} \quad (4)$$

$$\left(\sigma^2_{t,\mathrm{SR,mis}}\right)_{\mathrm{avg}N} = \frac{1}{N}\cdot\sum_{n=1}^{N}\sigma^2_{t,\mathrm{Latch,mis}} = \sigma^2_{t,\mathrm{Latch,mis}}. \quad (5)$$

IV. Comparison Between DLL and SR Jitter

A. Comparing Jitter Transferred From the Reference Clock

From the analysis above, we see that both the DLL and SR MPCGs render no improvement on the reference clock jitter. However, the SR MPCG needs a reference clock with N times higher frequency than the DLL. If both clocks are generated by a VCO,[1] the VCO for the SR should work at N times higher frequency, raising the question how this impacts power consumption. Assuming the VCO has an f^{-2} power spectrum and its quality of design is adequately assessed via the often used figure of merit FOM [7], the single sideband phase noise to carrier ratio at an offset frequency f_m can be expressed as

$$L(f_m) = \frac{10^{\mathrm{FOM}/10}}{P_{\mathrm{VCO}}}\cdot\frac{f^2_{\mathrm{VCO}}}{f^2_m} \quad (6)$$

where f_{VCO} is the frequency and P_{VCO} is the power dissipation in milliwatts. It is well known that the variance for stationary absolute jitter is related to the total area of its power spectrum, i.e., the reference clock jitter variance $\sigma^2_{t,\mathrm{ref}}$ becomes

$$\sigma^2_{t,\mathrm{ref}} = \frac{2\times\int_{f_l}^{f_h} L(f_m)df_m}{(2\pi f_{\mathrm{VCO}})^2} = \frac{10^{\mathrm{FOM}/10}}{2\pi^2\cdot P_{\mathrm{VCO}}}\cdot\left(\frac{1}{f_l}-\frac{1}{f_h}\right) \quad (7)$$

where $[f_l, f_h]$ is the specified integration region. Equation (7) indicates that although the VCO in the SR MPCG runs at N times higher frequency, it outputs the same jitter, given the same power and the same quality of design. If an *LC* VCO is used, higher working frequency may even be preferred, since the quality factor of an inductor $(\omega L/R)$ increases with frequency and smaller inductors are needed (less chip area). On

[1]The VCO can be part of a synthesizer, e.g., a PLL. We didn't discuss the effect of the PLL loop on the reference clock phase noise since it's the same for the SR and DLL. The PLL for the SR does not require an extra divide-by-N since the SR itself functions as a divide-by-N and can be re-used.

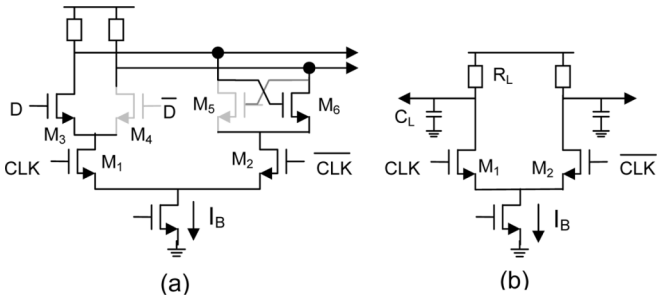

Fig. 3. (a) Schematic of a CML latch at the switching instant. (b) Simplified schematic for jitter analysis.

the other hand, there are limits to increasing the frequency, and also clock buffer power consumption can become an issue.

B. Comparing Jitter Generated due to Thermal Noise

Because of better supply noise rejection, current-mode logic (CML) circuits are often used in low jitter designs. To compare the jitter generated by the two MPCGs, we assume that they both use CML circuits. The simplified schematic of a CML delay unit is shown in Fig. 1(b). It is based on an nMOS source coupled differential pair driving the resistive load R_L and biased by a current source I_B. As the loads are *RC* circuits, the propagation delay t_d can be approximated as

$$t_d = \ln 2 \cdot R_L C_L = \ln 2 \cdot (V_{\rm SW}/I_B) \cdot C_L \tag{8}$$

where $V_{\rm SW}$ is the differential output swing and is determined by R_L and I_B due to the full switching of the tail current.

The CML implementation of a latch is shown in Fig. 3(a). For a proper operation, the D inputs of the latch should be already stable before the CLK starts to switch. For example, D is high and $\bar{D}$ is low and therefore, at the switching moment, transistors M4 and M5 are off. M3 and M6 are in their saturation region and work as cascode transistors on top of the differential pair. The noise contribution of M3-M6 can thus be neglected. The schematic of the latch can be simplified to Fig. 3(b) which is exactly the same as the schematic of the CML delay unit in Fig. 1(b). Therefore, we can apply the same noise jitter analysis for the delay unit and the latch.

The noise jitter variance of a CML delay unit can be predicted using the analysis presented in [8] as

$$\sigma^2_{t,\rm noise} = \left(1 + \gamma + \gamma_T \cdot \frac{2I_B}{V_{{\rm OV},T}} \cdot \frac{R_L}{2}\right) \cdot \frac{2kTC_L}{I_B^2} \tag{9}$$

where γ and γ_T are, respectively, the noise factor of the differential pair transistors and the tail bias transistor, $V_{{\rm OV},T}$ is overdrive voltage of the tail bias transistor and $2I_B/V_{{\rm OV},T}$ represents its transconductance assuming a square-law model.

In most of the clock generator designs, jitter and power are two important parameters. Via admittance level scaling [9], both noise and mismatch jitter can always be reduced at the cost of increasing the power consumption P. In order to take this tradeoff into account and make a fair comparison, jitter variance is normalized to power, with 1 mW as reference

$$\left(\sigma_t^2\right)_{\rm NorP} = \sigma_t^2 \cdot (P/1\ {\rm mW}). \tag{10}$$

For a given circuit, applying admittance level scaling will not change the value of $(\sigma_t^2)_{\rm NorP}$. Smaller $(\sigma_t^2)_{\rm NorP}$ means generating less jitter for a given amount of power. For a CML circuit, the power consumption is dominated by the static power $I_B \cdot V_{\rm DD}$. With (9) and (10), we find for both a CML delay unit and latch

$$\left(\sigma^2_{t,\rm noise}\right)_{\rm NorP} = \left(1 + \gamma + \gamma_T \cdot \frac{I_B R_L}{V_{{\rm OV},T}}\right) \frac{2kT \cdot V_{\rm DD}}{1\ {\rm mW}} \cdot \frac{C_L}{I_B}. \tag{11}$$

Substituting (8) into (11) yields

$$\left(\sigma^2_{t,\rm noise}\right)_{\rm NorP} = \left\{\left(1 + \gamma + \gamma_T \cdot \frac{V_{\rm SW}}{V_{{\rm OV},T}}\right) \frac{2kT \cdot V_{\rm DD}}{\ln 2 V_{\rm SW} \cdot 1\ {\rm mW}}\right\} t_d. \tag{12}$$

Equation (12) indicates that the *normalized noise jitter variance is proportional to* t_d for a given power budget.

In a DLL, if t_d is tuned by tuning R_L while keep $V_{\rm SW}$ constant, I_B and thus $V_{{\rm OV},T}$ in (12) will vary with t_d. Here to simplify the comparison, we ignore this second order effect and assume the delay unit and the latch have the same $V_{\rm SW}$ and $V_{{\rm OV},T}$. We will see the effect of this simplification in Section V. A DLL has N delay units contributing to jitter and power while a SR has N latches contributing to jitter and 2 N latches dissipating power. The average noise jitter variance generated by the DLL and the SR MPCGs can then be compared using (1), (4) and (12), as

$$\begin{aligned}\frac{\left(\sigma^2_{t,\rm SR,noise}\right)_{{\rm avg}N,\rm NorP}}{\left(\sigma^2_{t,\rm DLL,noise}\right)_{{\rm avg}N,\rm NorP}} &= \frac{\left(\sigma^2_{t,\rm Latch,noise}\right)_{\rm NorP} \times 2N}{\frac{N+1}{2} \times \left(\sigma^2_{t,\rm DU,noise}\right)_{\rm NorP} \times N} \\ &= \frac{4}{N+1} \cdot \frac{t_{d,\rm Latch}}{t_{d,\rm DU}}.\end{aligned} \tag{13}$$

The comparison result thus depends on the amount of delay of the delay unit $t_{d,\rm DU}$ and that of the latch $t_{d,\rm Latch}$. In a DLL MPCG, the VCO defines the frequency and the VCDL defines the delay in between the N output clocks. Both the VCO and the delay line need to be tuned for the DLL MPCG to work at a frequency f, where the delay of each delay unit should satisfy

$$t_{d,\rm DU} = \frac{T}{N} = \frac{1}{N \cdot f}. \tag{14}$$

In contrast, the SR MPCG is more flexible. For different f, only the VCO needs to be tuned since both the frequency and the delay in between the N output clocks are defined by the clock period of the VCO. The only concern is that the DFFs should operate correctly, which requires [10]

$$t_{d,\rm Latch} + t_{\rm su} \le \frac{1}{N \cdot f} \tag{15}$$

where $t_{\rm su}$ is the setup time required by the DFF. Defining the maximum working frequency of a SR MPCG for N-phase clock generation in a certain technology as $f_{\rm max,SR}$, the latch delay will have its minimum value $t_{d,\rm Latch,min}$ at $f_{\rm max,SR}$ given by

$$t_{d,\rm Latch,min} = \frac{1}{1 + \alpha_{\rm su}} \cdot \frac{1}{N \cdot f_{\rm max,SR}} \tag{16}$$

with $\alpha_{\rm su}$ the ratio between $t_{\rm su}$ and $t_{d,\rm Latch,min}$. As a small delay is preferred for a small $(\sigma^2_{t,\rm noise})_{\rm NorP}$, the latch delay can be equal to its minimum in (16). For a delay unit, the delay is limited by (14). Taking this factor into account, (13) can be re-written as

$$\frac{\left(\sigma^2_{t,\rm noise,SR}\right)_{\rm avg,NorP}}{\left(\sigma^2_{t,\rm noise,DLL}\right)_{\rm avg,NorP}} = \frac{1}{1+\alpha_{\rm su}} \cdot \frac{f}{f_{\rm max,SR}} \cdot \frac{4}{N+1}. \tag{17}$$

As soon as the wanted number of clock phases is larger than three ($N > 3$), (17) is smaller than one since the DFF needs a finite setup time ($\alpha_{\rm su} > 0$) and the working frequency of the SR can't surpass the technology limit ($f \le f_{\rm max,SR}$). This means that the SR based MPCG generates less noise jitter than the DLL counterpart for a given power budget. Equation (17) also indicates that the advantage of the SR based MPCG will be larger if more advanced technologies are used and in applications where clocks with a larger number of phases at lower frequencies are needed.

C. Comparing Jitter Generated due to Mismatch

Based on similar reasoning as for the noise jitter analysis, the latch can be simplified as shown in Fig. 3(b) for mismatch jitter analysis and we can apply a similar analysis. In a CML delay unit, there are two mismatch jitter sources: one is the RC load which contributes to RC delay mismatch $\sigma^2_{t,RC,\rm mis}$ and the other is the differential pair input referred offset voltage $\sigma^2_{V\rm off}$ which makes the switching moment deviate from the actual crossing point of the input clocks. The tail bias transistor mismatch does not lead to jitter since it's a common mode error and we are interested in the crossing points.

Using (8), the jitter due to the RC load mismatch becomes

$$\left(\frac{\sigma_{t,RC,\rm mis}}{t_d}\right)^2 = \sigma^2_{\Delta R_L/R_L} + \sigma^2_{\Delta C_L/C_L} \tag{18}$$

with ΔR_L and ΔC_L the absolute error in the value of R_L and C_L.

In a DLL, the RC delay must be tunable. For simplicity, we assume that C_L is tuned by putting less or more capacitors in parallel and R_L is tuned by putting less or more resistors in parallel.[2] Since the matching improves with area [9], (18) can be rewritten as

$$\sigma^2_{t,\rm RC,mis} = [(A_R \cdot \sqrt{R_L})^2 + (A_C/\sqrt{C_L})^2] \times t_d^2 \tag{19}$$

where A_R and A_C are IC process constants for the matching property of the load resistance and capacitance, respectively.

The input referred offset voltage of a differential pair can be calculated using the method presented in [11] as

$$\sigma^2_{V\rm off} = \sigma^2_{\Delta Vt} + \frac{I_B}{4K} \times \sigma^2_{\Delta R'_L/R_L} + \frac{I_B}{4K} \times \sigma^2_{\Delta K/K} \tag{20}$$

where $\sigma^2_{\Delta Vt}$ is the differential pair threshold voltage mismatch variance, $\Delta R\prime_L$ is the relative error between the two R_L loads, K is the transconductance parameter of the differential pair with $\sigma^2_{\Delta K/K}$ describing its mismatch.

The total mismatch jitter variance $\sigma^2_{t,\rm mis}$ can be found by adding $\sigma^2_{t,\rm RC,mis}$ and the jitter variance caused by $\sigma^2_{V\rm off}$ which is $\sigma^2_{V\rm off}$ divided by $(I_B/C_L)^2$, the square of the slope of the differential switching voltage at the zero crossing.

$$\sigma^2_{t,\rm mis} = A_R^2 \cdot R_L \cdot t_d^2 + \frac{A_C^2 \cdot t_d^2}{C_L} + \frac{\sigma^2_{\Delta Vt} + \frac{I_B}{4K} \times A_R^2 \cdot R_L + \frac{I_B}{4K} \times \sigma^2_{\Delta\beta/\beta}}{(I_B/C_L)^2}. \tag{21}$$

The power normalized mismatch jitter variance can be derived with (10) and (21) as

$$\left(\sigma^2_{t,\rm mis}\right)_{\rm NorP} = \frac{V_{\rm DD}}{1\ \rm mW} \cdot \left\{ V_{\rm SW} \cdot A_R^2 \times t_d^2 + \ln 2 \cdot V_{\rm SW} \cdot A_C^2 \times t_d + \frac{\sigma^2_{\Delta Vt}}{\ln 2 \cdot V_{\rm SW}} \times C_L \cdot t_d + \frac{A_R^2}{\ln 2 \times 4K} \times C_L \cdot t_d + \frac{\sigma^2_{\Delta K/K}}{4K} \times C_L^2 \right\}. \tag{22}$$

Equation (22) shows that the delay unit and latch generates less mismatch jitter for a smaller delay, with a given power. It also suggests that with a constant $V_{\rm SW}$, it's better for a DLL to tune up R_L instead of C_L when larger delay is needed.

Assuming the terms with t_d proportionality in (22) which include the threshold voltage mismatch are the dominating mismatch jitter sources and setting the other initial conditions the same for a fair comparison, the mismatch jitter generated by the DLL and SR can be compared with (3), (5), and (22) as

$$\frac{\left(\sigma^2_{t,\rm SR,mis}\right)_{{\rm avg}N,\rm NorP}}{\left(\sigma^2_{t,\rm DLL,mis}\right)_{{\rm avg}N,\rm NorP}} \approx \frac{12}{N} \cdot \frac{t_{d,\rm Latch}}{t_{d,\rm DU}}. \tag{23}$$

Substituting (14) and (16) into (23) yields

$$\frac{\left(\sigma^2_{t,\rm SR,mis}\right)_{{\rm avg}N,\rm NorP}}{\left(\sigma^2_{t,\rm DLL,mis}\right)_{{\rm avg}N,\rm NorP}} = \frac{1}{1+\alpha_{\rm su}} \cdot \frac{f}{f_{\rm max,SR}} \cdot \frac{12}{N}. \tag{24}$$

The situation where (24) is larger than one only occurs when the wanted number of clock phases N is smaller than 12 together with a high frequency f close to $f_{\rm max,SR}$. In other cases, (24) is smaller than one, which means that the SR MPCG generates less mismatch jitter than the DLL counterpart for a given power budget. Equation (24) also indicates that the advantage of the SR based MPCG will be larger if more advanced technologies are used and a larger number of clock phases at lower frequencies are needed.

D. Discussion

The analysis above shows that a SR MPCG transfers the same jitter from the reference clock and almost always generates less jitter[3] than a DLL MPCG for a given power consumption. For

[2]If R_L is realized with a MOS transistor in linear region and R_L is tuned by tuning the gate voltage, it can be shown that the matching property of R_L in a DLL delay unit is even worse.

[3]In case phase noise is important, the SR is also better as both the SR and DLL generate white phase noise, while the reference clock has the same spectrum shape for both cases.

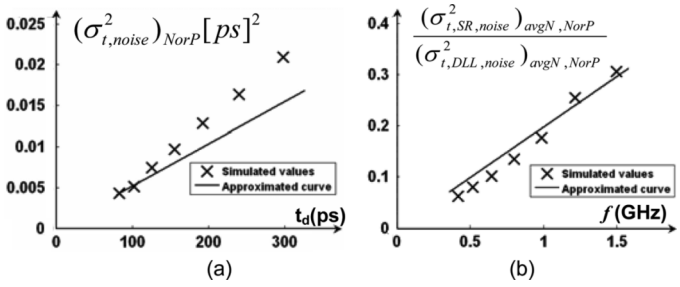

Fig. 4. Noise jitter simulation results in 0.13-μm CMOS with $N = 8$ for: (a) a CML delay unit, and (b) DLL and SR comparison.

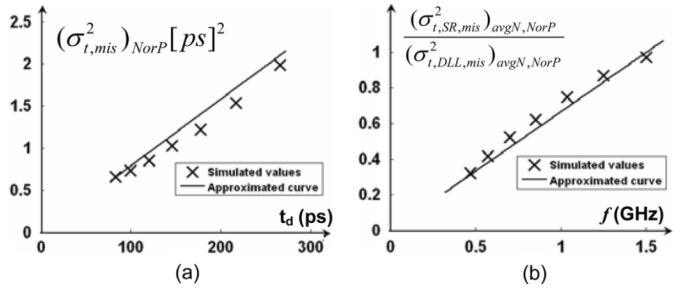

Fig. 5. Mismatch jitter simulation results in 0.13-μm CMOS with $= 8$ for: (a) a CML delay unit, and (b) DLL and SR comparison.

mismatch jitter, the DLL MPCG may have a slight advantage in some high-frequency cases.[4]

From an implementation point of view, the SR MPCG has a simpler architecture since it does not require analog tuning and no feedback is needed. However, it can be more difficult to implement in applications where N is large and f is high since it works at $N \cdot f$, but this improves as technology advances. Another concern is that the loading of the VCO is more severe in the SR MPCG, since it needs to drive N DFFs. This problem can be alleviated by downscaling the DFFs by admittance scaling [9], which is acceptable because they generate less jitter than the dely units, thus saving power and chip area.

From a multifunctionality point of view, the SR MPCG is clearly more attractive: it is basically a digital circuit which can operate from arbitrarily low frequency up to $f_{\max,\mathrm{SR}}$, while a DLL required tuning of an "analog" delay. Also, a SR can basically instantaneously change its output frequency, while a DLL settles slowly, due to the preferred low loop bandwidth. Finally, a SR MPCG has the flexibility to generate clocks with different duty cycle.

V. Simulation Results

In order to verify the calculations, simulations were done for a DLL and a SR for $N = 8$ in 0.13-μm CMOS. The reference clocks are voltage sources with 1-kΩ source resistance. The VCDL delay is tuned up by tuning up the load resistance as suggested by (22) while keep V_{SW} to be 0.6 V. For the DFFs, α_{su} is about 0.5. The load capacitance is 100 fF, which is comparable to the parasitic capacitances. In this implementation, $f_{\max,\mathrm{SR}}$ is about 1.5 GHz for 8-phase clock generation. Fig. 4 shows the strobed PNoise analysis results for noise jitter. The simulated values coarsely fit the estimated curve. The larger deviation when t_d is larger relates to the simplification we made below (12). We see this simplification is in favor of the DLL which normally has a larger t_d. Therefore, it does not affect the conclusion. Fig. 5 shows the Monte Carlo analysis results for mismatch jitter. The bent shape of the simulated values when t_d is tuned from low to high is predicted by (22). The simulated values fit the estimated curve well which means the threshold voltage mismatch dominates in this design.

[4]If 50% reference clock duty cycle is guaranteed, both edges can be used. The N DFFs in the SR can be replaced with N latches as in [3]. The previous analysis then overestimates the SR MPCG power consumption by two times.

VI. Conclusion

This paper motivates why a SR MPCG is more attractive for flexible multifunctional circuits than a DLL MPCG as it is easier to change its frequency and duty cycle. Furthermore, analysis shows that a SR MPCG almost always generates less jitter than a DLL equivalent when both are realized with CML circuits, at a given power budget. This is partly because a SR MPCG has no jitter accumulation from one clock phase to the other as in a DLL counterpart. In addition, a SR MPCG can use latches with very small delay time, while jitter generation of a CML circuit is proportional to its (functionally required) delay time. A SR MPCG requires a reference clock with higher frequency, which can be realized in a power neutral way provided that the VCO core determines power consumption. The advantages of a SR MPCG will be larger as technology advances.

References

[1] C. K. Yang and M. A. Horowitz, "A 0.8-/spl mu/m CMOS 2.5-Gb/s oversampling receiver and transmitter for serial links," *IEEE J. Solid-State Circuits*, vol. 31, no. 12, pp. 2015–2023, Dec. 1996.

[2] W. C. Black and D. A. Hodges, "Time-interleaved converter arrays," *IEEE J. Solid-State Circuits*, vol. SC-15, no. 6, pp. 1022–1029, Dec. 1980.

[3] E. A. M. Klumperink, R. Shresta, E. Mensink, V. J. Arkesteijn, and B. Nauta, "Cognitive radios for dynamic spectrum access—Polyphase multipath radio circuits for dynamic spectrum access," *IEEE Commun. Mag.*, vol. 45, no. 5, pp. 104–112, May 2007.

[4] X. Gao, E. Klumperink, and B. Nauta, "Low-jitter multiphase clock generation: A comparison between DLLs and SRs," in *Proc. Int. Symp. Circuits Syst.*, May 2007, pp. 2854–2857.

[5] R. van de Beek, E. Klumperink, C. Vaucher, and B. Nauta, "Low-jitter clock multiplication: A comparison between PLLs and DLLs," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 49, no. 8, pp. 555–566, Aug. 2002.

[6] M.-J. Edward Lee *et al.*, "Jitter transfer characteristics of delay-locked loops-theories and design techniques," *IEEE J. Solid-State Circuits*, vol. 38, no. 4, pp. 614–621, Apr. 2003.

[7] P. Kinget, , W. Sansen, J. Huijsing, and R. van de Plassche, Eds., "Integrated GHz voltage controlled oscillators," in *Analog Circuit Design: (X)DSL and Other Communication Systems; RF MOST Models; Integrated Filters and Oscillators*. Boston, MA: Kluwer, 1999, pp. 353–381.

[8] S. Levantino *et al.*, "Phase noise in digital frequency dividers," *IEEE J. Solid-State Circuits*, vol. 39, no. 5, pp. 775–784, May 2004.

[9] E. Klumperink and B. Nauta, "Systematic comparison of HF CMOS transconductors," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 10, pp. 728–741, Oct. 2003.

[10] J. M. Rabaey, *Digital Integrated Circuits, A Design Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[11] P. Gray, P. Hurst, S. Lewis, and R. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th ed. New York: Wiley, 2001, pp. 236–237.