Antonov, Yury; Stadius, Kari; Ryynänen, Jussi

A Charge Limiting and Redistribution Method for Delay Line Locking in Multi-Output Clock Generation

Published in:
2017 IEEE International Symposium on Circuits and Systems (ISCAS) Proceedings

DOI:
10.1109/ISCAS.2017.8050740

Published: 28/09/2017

Please cite the original version:
A Charge Limiting and Redistribution Method for Delay Line Locking in Multi-Output Clock Generation

Yury Antonov, Kari Stadius and Jussi Ryynänen
Department of Micro- and Nanosciences, School of Electrical Engineering
Aalto University, Espoo, Finland, P.O. Box 13000 Aalto
Email: yury.antonov@aalto.fi

Abstract—This paper proposes a new type of delay line locking mechanism with digitally controlled charge transfer. Delay-locked loop (DLL) based on the presented method features multi-phase outputs and is stepwise driven towards lock by a co-action of 1-bit Time-to-Digital Converters and revised charge-pump. On-chip pulse “slicing” arrangement provides high-rate clock for the Digital Signal Processing algorithm, enabling fine-tuning of the proposed DLL. Locking mechanism is implemented with standard digital cells and complete mixed-signal design is simulated in 28nm ST CMOS with full physical device models to prove functionality. When locked to reference frequency of 1.25GHz, this design consumes 1.1mW from 1V supply and produces 64+64 12ps-spaced output phases with <176fs phase error ripple and 50dB SFDR.

I. INTRODUCTION

Various modern applications (including polar transmitters, mixer-first receivers, many-core processors, phased-arrays, etc.) require multi-phase low-jitter clock generator as one of the key building blocks in the System-on-Chip. Among several ways to generate multiple phases, Delay-Locked Loops (DLLs) offer unconditional stability and faster design process in contrast to PLL-based architectures. Fed from low-frequency reference, DLLs consume smaller dynamic power and restrict phase noise accumulation [1].

It is attractive to use digital cells in locking mechanism implementation, which would leverage Digital Signal Processing (DSP), enable automatic placing and routing, and make implementation robust against process variations and portable across process nodes. However, shown in Fig. 1a basic phase detector (PD), charge pump (CP) and loop filter (C_{LF}) combinations [2, 3] as well as advanced analog designs [4, 5] are not directly transferable to digital assistance paradigm. First, techniques to stabilize voltage on integration capacitor C_{LF} rely on analog matching of source and sink currents. Second, width of the pumped current pulse is directly proportional to the phase difference between PD inputs, leaving no mechanism for stepwise fine-tuning.

In the paper we propose a new type of delay line locking mechanism, where voltage on the integration capacitor C_{LF} becomes defined not by the amount of time source/sink current is active, but the number of times integration capacitor is switched to precharged/discharged branches C_{UP}/C_{DN}. This yields several benefits. First, pulsed control of CP branches enables stepwise tuning of the DLL. Second, limiting charge at the fine-tuning stage permits reduction of tuning voltage ripple (equivalently, phase error ripple). Third, combination of the two techniques allows on-chip algorithm to pin-point locked condition. Finally, DSP clock generator “pulse-slices” reference period by leveraging delay line outputs with no need for additional oscillating structures.

Section II of this paper describes overall DLL architecture, with subsections detailing various building blocks. Simulation results are given in section III, and section IV concludes the paper.

II. PROPOSED DLL ARCHITECTURE

DLL with multi-phase pulsed outputs is outlined in Fig. 1b. Delay line includes 68 identical elements where individual delays T_{UD} are controlled by tuning voltage V_{TUNE} with the aid of digital processing. In the following subsections, we introduce a circuitry for stepwise adjustment of V_{TUNE} so that total line delay aligns with the reference frequency cycle. This architecture produces 2 × 64 T_{UD}-spaced clock phases.
A. Proposed locking algorithm

Locking algorithm flowchart in Fig. 2 illustrates coarse and fine tuning steps. Proposed mechanism exploits observation that if resolution of 1-bit Time-to-Digital Conversion (TDC) is too coarse to further drive charge pump towards lock, then duty-cycle of recorded TDC output starts to oscillate around proper but unreachable value.

At the coarse tuning stage, DLL is directed towards lock by co-action of 1-bit TDC and 2-bit Pulse DAC, enabling stepwise adjustments. Unlike typical approach, the charging current is “stepped” with intermediate capacitors so that integration capacitor is never connected directly to supply rails (except for the first moments of chip power-up).

Fine tuning stage begins when calculated duty-cycle of a TDC output is close enough to a programmed threshold value $DC_{TH}$. In this case, power supplies disconnect from capacitance group and therefore net charge of the group is not replenished while DLL settles to locked condition.

When capacitance group starves charge, it effectively lengthens duration to flip the total line delay over the equilibrium with reference cycle. When incremental steps of $V_{TUNE}$ last long enough, onchip algorithm is able to pick up locked condition.

B. Proposed charge pump

Charge pump revised to enable stepwise tuning is drawn in Fig. 3. In the proposed approach, two groups of switches separate integration capacitor from the supplies and two extra capacitors are added.

Switches $up/dnx$ turn additional capacitors into charge reservoirs, precharging $C_{UP}$ and discharging $C_{DN}$ respectively. At the coarse-tuning stage these capacitors accumulate limited extra charge for further refinement of $V_{TUNE}$. In the fine-tuning process, switches $upx/dn$ redistribute this extra charge within the capacitance group, while switches $up/dnx$ stay halted. Being proportional to the unitary delays, switch control pulses shrink with $T_{UD}$ as DLL proceeds to locked condition, ensuring smaller steps in the fine-tuning. All switches $up/upx/dn/dnx$ differ in size so that target duty-cycle of $DC_{TH}$ is recorded by on-chip tracking when DLL is near lock.

C. Unitary delay element

Detailed schematic of variable unitary delay is provided in Fig. 4. Differential delay elements are preferred in this design to double the amount of output phases. Strong CMOS inverters are arranged in pseudo-differential manner by cross-coupling outputs through weaker inverters. Since the inverter delay shrinks as circuits move to smaller process nodes, 2-bit band select allows lower $f_{ref}$ frequencies for the DLL. Voltage controlled tunable delay is implemented by loading the outputs with a NMOS varactor. Capacitor loading is favored to provide close to linear delay tuning characteristics.
D. Period slicing logic

Shown in Fig. 4 is reference period “slicing” arrangement to obtain pulsed outputs and enable further on-chip frequency multiplication. Taking advantage of multi-output layout, standard AND cells are connected to various antiphase outputs of the delay line to create the desired pulse width. A duration of 4 unitary delays is chosen for this design to meet the minimum pulse width requirement of standard digital cells.

E. Local clock generator

Local generator (Fig. 5) establishes high-frequency clock domain $clk$ for DSP. On-chip frequency multiplication proceeds by assembling small periods out of 4 correspondingly shifted pulses via four-input OR gate. With this approach no self-oscillating or start-up circuitry is required and $clk$ is produced whenever $fref$ is present at the DLL input. For resampling purpose it is enough to have $clk = 2 \times fref$, since there are at most 2 TDC output changes per $fref$ period. An on-board frequency divider provides clock and timely reset signals for duty-cycle tracking mechanism. Divider is constructed with combinational logic feedback on top of a shift register and, having no glue logic in-between the stages, is well suited for gigahertz $clk$ feed.

F. Pulse Digital-to-Analog Converter

Pulse DAC (Fig. 5) converts lead-lag detection stream to corresponding quantities of short pulses, outputted at $8 \times fref$ frequency each $clk$ period. Generation of $pdacdn$ is identical to shown $pdacup$ circuitry, except that $tdc1q$ and $tdc2q0$ signals are replaced with $tdc2qnx$, $tdc1qn$, and all pulse feeds come from the lower half of DVCDL (suffix “x” in Fig. 4).

G. Duty-cycle tracking

Shown in Fig. 6 is an on-chip duty-cycle estimation arrangement, monitoring whether delay-line has coarsely settled. At the first stage of estimation, accumulator measures instantaneous pulse-width of TDC output in $clk$-cycles. At the second stage, simple Infinite Impulse Response filter (IIR) estimates duty-cycle of the TDC output and averages instantaneous anomalies. For compact implementation, filter coefficients are realized as programmable right-shifts by $\lambda \in \{1, 2, 3\}$ bits.

Unlike traditional scheme in Fig. 1a, presented TDC features D-type no-reset flip-flops, allowing highest operating frequency. Moreover, such TDC enables delay line to sample itself, directly benefiting from D-flop resolution improvement in successive process nodes. For a faster acquisition rising edges of both $ref$ and $refx$ waveforms are used in TDC: second conversion corrects the previous result through LSBs of respective PDACs (TDC outputs in Fig. 6 are interleaved and differently delayed during resampling in Fig. 5).

As total line delay shrinks under limited charge fine-tuning, TDC output becomes more elaborate and $dnctl$ starts to include full 00 $\Rightarrow$ 10 $\Rightarrow$ 11 $\Rightarrow$ 01 $\Rightarrow$ 00 sequence. This indicates that total line delay has been equal to reference cycle 2 times. After the above sequence, when $upctl$ transition 01 $\Rightarrow$ 00 occurs, a single CP pulse suffices to go past the locked state, which in this design identifies lock vicinity.

H. Controlling Finite State Machine

Controlling FSM logic (Fig. 6) tests whether duty-cycle of a lead-lag waveform is constrained in the $\Delta$-vicinity of programmable $DC_{TH}$ setting. As rising edges of $p(x)0$ and $p(x)63$ are close enough to each other, measured duty-cycle values start oscillating at target $DC_{TH}$. Controlling FSM asserts $frz$, thus freezing $up/dnx$ switches and enabling fine-tuning mode. The latter is monitored to detect joint occurrence of the specified $dnctl$ and $upctl$ sequences and the state of DLL is then locked by halting CP updates with $lck$ control.

III. SIMULATION RESULTS

DLL with multi-phase pulsed outputs is designed in 28nm ST CMOS and verified in Eldo with mixed-signal simulations of full physical device models. Simulated design operates at 1V power supply and DLL is locked to clean $fref=1.25GHz$ ($T_{REF}=800ps$), thus exhibiting 12ps delay per delay element.
Power consumption of digital blocks is 1.7mW (deactivated when lock is achieved), while total power consumption (adding delay line and capacitance group) amounts to 2.8mW.

Essential signals when locking develops in time are plotted in Fig. 7a. Phase error reduction between opposite ends of the delay line is shown in Fig. 7b. Output power and phase noise spectra are exhibited in Figs. 7c,d respectively.

IV. CONCLUSION

We have shown a new type of delay line locking mechanism and an exemplary DLL design. Utilizing revised charge pump and digital assistance to attain lock, this design produces $2 \times 64$ 12ps-spaced output phases with $<176$fspp ripple and 50dB SFDR from clean 1.25GHz reference frequency.

Consuming 1.1mW from a supply of 1V, presented high-frequency low-power architecture can serve as an ad-hoc tuning engine for delay-line-based applications or be reused in other multi-phase and pulsed circuits.

ACKNOWLEDGMENT

The authors would like to thank Tapio Rapinoja for supplying reference design of conventional DLL from [6].

This research has received funding from Academy of Finland.

REFERENCES