

Programmable Delay Controller Allowing Frequency Synthesis and Arbitrary Binary Waveform Generation

Marek Peca*, Michael Vacek, Vojtěch Michálek

Výzkumný a zkušební letecký ústav, a. s., Beranových 130, Praha – Letňany, Czech Republic

České vysoké učení technické v Praze, Břehová 7, 115 19 Praha 1, Czech Republic

*e-mail: pec@vzlu.cz

Abstract—Design of a programmable delay controller (PDC) within an aerospace-compatible field-programmable gate array (FPGA) fabric is presented. Although PDC is a common digital block nowadays, the possibility to use it for low-jitter arbitrary frequency generation constrained only by minimum edge-to-edge time still seems to be uncovered. The novel idea of the developed PDC is seamless line delay switching at sampling frequencies corresponding to the generated output frequency, unleashing a possibility of arbitrary binary waveform (frequency) generation. The maximum frequency is constrained only by the FPGA fabric performance, and by idle delay of available multiplexers. Glitch-free operation with no unintentional edges is employed for proper PDC control signal switching. The overall output signal jitter is composed solely of the jitter of input signal and propagation jitter of the delay elements ($\sigma_{max} = 4.1$ ps RMS). Measured resolution of the PDC is $\pm\Delta\tau_{max}/2 = \pm 7.5$ ps. Measured temperature drift of the PDC is ~ 30 ps K⁻¹. An ability of PDC to generate fractional frequency from the input has been demonstrated on a simplified, low-resolution variant, delivering 33.3 MHz out of 50 MHz input.

I. INTRODUCTION

Design of a programmable delay controller (PDC) within an aerospace-compatible, FLASH-based field-programmable gate array (FPGA) fabric will be presented. Although PDC is a common digital block nowadays, the possibility to use it for low-jitter arbitrary frequency generation still seems to be uncovered. The objective is to provide a versatile PDC block capable of generating arbitrary binary waveform constrained only by minimum achievable edge-to-edge switching time. The discussed PDC may be able to replace current building blocks such as direct digital synthesizer (DDS), phase-locked loop (PLL), micro phase stepper, or various modulators; all within purely digital circuit (FPGA, custom CMOS).

Let us call the time interval between control signal changes (updates) the sampling period T_s , and let the time interval between the consecutive edges of output signal be called the output interval T_o . Current PDC designs [1], [2], [3], [4], [5], [6], [8] operate at $T_s \gg T_o$, Fig. 1a. The key idea of our proposal is seamless PDC operation at $T_s \rightarrow T_o$, Fig. 1b. Such a development unleashes following advantages:

- generation of arbitrary binary waveform constrained only by $T_o \geq T_s$,

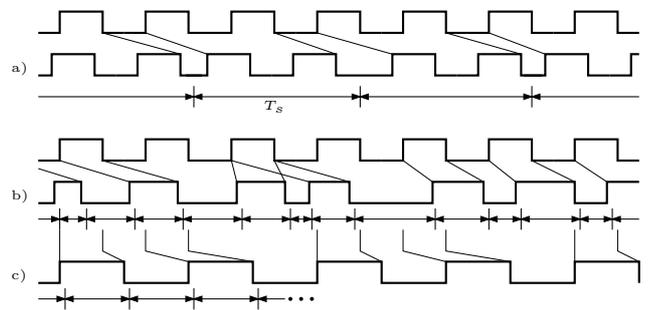


Fig. 1. Input and output waveforms for (a) ordinary PDC, (b, c) PDC allowing rapid switching

- generation of arbitrary frequency, constrained by $f \leq \frac{1}{2 \cdot T_s}$,
- under proper treatment, no edge will be created by the switching elements themselves; therefore, the overall jitter will be composed solely of the jitter of input signal and propagation jitter of the delay elements.

Design and implementation details of the PDC circuit are described. Performance figures, such as pass-through random jitter, achieved delay granularity, and temperature drift follow.

II. CIRCUIT DESIGN

A. Delay line design

This section describes design of the essential PDC component: a delay line. Typical PDC based on a digital circuitry consists of fixed-length delay elements (transmission lines, gates) composed into signal path by electronic switches (multiplexers). An input signal passes through the delay elements and multiplexers to the output. The multiplexers are switched by means of another digital control signal.

The most common approach is to form the delay line as a cascade of binary elements, Fig. 2a. A favorable property of this design is that n elements may cover 2^n equidistant delay steps, provided element delays follow $K2^{-k}$ distribution, i. e. $d_{k+1} = d_k/2$. Such a delay line acts similarly to a Digital-to-Analog Converter (DAC), but working in time domain. Moreover, the conversion is monotonic by design. Since there is a lack of short delay elements within FPGA fabric, a differential

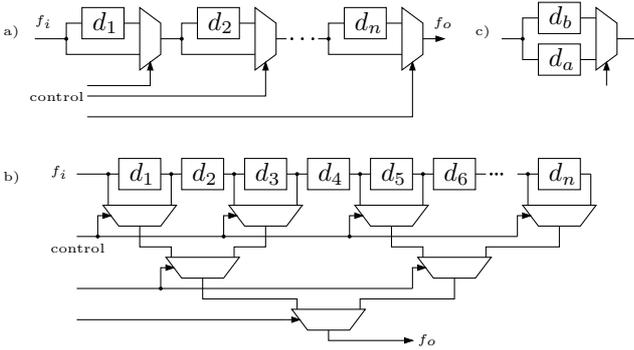


Fig. 2. PDC delay line structure: (a) cascaded design (b) tree-like design (c) fine differential delay

approach to achieve very fine resolution is employed, Fig. 2c. Only $d_b - d_a$ effectively applies, allowing to push resolution down to $\sim 10^{-12}$ s level.

Another approach to delay line uses continuous, tapped line and a binary tree of multiplexers, Fig. 2b. Despite its large area consumption ($\sim 2^{b+1}$ cells for b bits), there is an advantage of uninterrupted delay line, always filled with valid, running signal. See more about its impact in Sec. II-B.

B. Control logic design

In order to achieve versatile signal generation as outlined in Sec. I., we have concluded that the multiplexer control signals should be switched at instants derived from output signal edges (of both polarity) thus operating PDC in a self-clocked manner. It is the most straightforward way to assure proper switching instants. Together with standard asynchronous FIFO, the PDC forms a circuit converting integer numeric data-stream consisting of desired output edge instants t_1, t_2, \dots into actual binary waveform, Fig. 4.

Block diagram of the PDC circuit is shown in Fig. 3. There are two identical delay lines, in order to mitigate glitches which may occur at the output of the delay line during control input reconfiguration. Just after a rising edge passes through the upper delay line, output multiplexer is switched down to the other delay line. Until next (falling) edge passes, the upper delay line may be safely reconfigured. After passage of the falling edge, the multiplexer is switched back to the upper line, and the lower delay line may be rearranged. Then, the process repeats. Both delay lines are controlled by binary delay words, one for the rising edges, other for the falling edges. The words are stored in respective registers. Loading of the registers and switching of the output multiplexer is controlled by the key part of the design, asynchronous state machine. Most importantly, it generates a non-uniform clock signal derived from the delayed edges. This clock also governs readout of the delay words from data source, i.e. a sequencer (in case of fixed sequence) or FIFO (if delays come from a complex computing system).

The most severe difficulty of the design lies in an unwanted (idle) delay between delay line reconfiguration instant and settling of output signal. In case of cascaded structure (Fig. 2a),

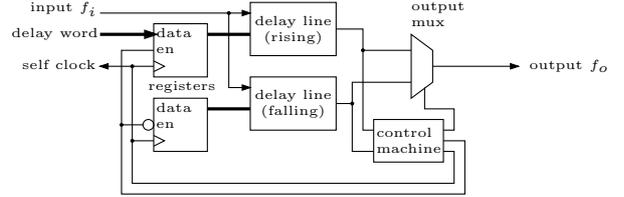


Fig. 3. Complete PDC with dual delay lines and control logic

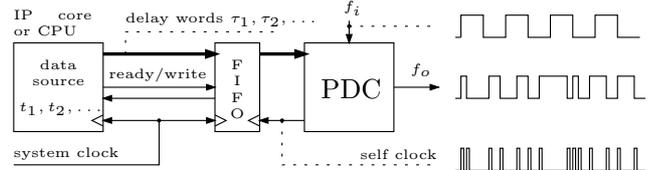


Fig. 4. Self-clocked PDC data-flow

only d_1 has always up-to-date signal at its input and inside. But $d_2 \dots d_n$ have to absorb new input condition upon re-configuration, what may take up to $\sum_{k=2}^n d_k$ amount of time. Pass-through delay of all multiplexers adds up. The tree-like structure (Fig. 2b) does not suffer from such transient problem, so its settling delay belongs solely to the multiplexers.

Suppose a PDC with resolution h and maximum delay T is requested. The cascaded structure would settle in $\sim T/2$ plus delays of multiplexers. Where differential elements (Fig. 2c) are used (FPGA fine part), the common mode delay adds up. The tree-like structure exhibits unwanted delay $\log_2 T/h$ times a multiplexer delay; unfortunately, it requires $\mathcal{O}(T/h)$ macros. We see that the proposed kind of rapidly switching PDC is hard to fit into ordinary FPGA fabric.

III. IMPLEMENTATION

A. Delay line implementation

The delay line was implemented inside Actel/Microsemi ProASIC3E FPGA. The line is composed of two distinct parts: coarse, and fine. Two versions of delay lines, with 5 and 13 control inputs (bits), were implemented. The coarse line (5 bits) is made up of 1, 2, 4, \dots combinatorial macro cells (“gates”) of the same type in chain. The fine line follows differential structure, with two specially selected macros, one per a branch.

In order to obtain reasonable resolution, all the cells should be well selected as to provide fine granularity over all 2^n bit words. To our knowledge, delay models of FPGA macros are not available in open format. Therefore, we have ran place-and-route tool chain in cycle, iterating over whole macro library, and extracted delay values from SDF back-annotated files [7]. Figures were far from accurate, and lacked an important kind of information: in multiple-input cells, the delay from one input to output may depend on a state of other inputs – this has not been reflected in SDF files (although it is possible by specification [7]). Nevertheless, the obtained timing data served us well as an initial guess. In following automated analysis, every possible cell pair was coupled

with appropriately (non-)inverting multiplexer (yielding non-inverting line element). All these combinations were examined for differential delay, sorted, and shortest candidates for given differential delay were selected. Interval from zero to ~ 0.5 ns has been continuously populated with 1 ps step. From this, candidates were picked up by hand, and subcircuit netlist has been generated. Candidate delay line, implemented into FPGA, has been measured (Sec. IV-A), and unsatisfactory cells were reassigned by hand in next iteration. This process was repeated three times.

First experimental delay line, used for performance analysis, was of 13 bit cascaded structure, 45 cells in total including multiplexers. For a frequency synthesis scenario (Sec. IV-B), 5 bit tree-like structure has been used in order to mitigate idle delay; also, elements counts were doubled to cover 50 MHz period.

B. Control logic implementation

PDC core controller has been designed by hand as an asynchronous state machine. An intuitive match of the machine to a set of 3-LUT macros of the ProASIC3E FPGA architecture was checked for equivalence in state encoding and transition maps to eliminate hazards. For demonstration purposes, only a fixed delay word sequence has been wired into shift-register based sequencer: approximations of $(0, T/4, T/2, 3T/4)$ for nominal input signal frequency $f_i = 50$ MHz (Fig. 1c). The aim was to demonstrate generation of $f_o = (2/3)f_i$ frequency by the PDC.

IV. RESULTS

A. Delay line characteristics measurements

There have been performed two fundamental experiments testing the performance of the delay line. The block diagram of measurement setup in which the PDC line resolution and linearity were measured using time interval counter (SR620 [10]) is shown in Fig. 5. The SR620 synchronous reference signal (1kpps) was connected into the FPGA fabric. Inside the FPGA, the signal is split into the PDC input and directly to the output of the FPGA chip and consequently into the A channel of the SR620. The output of the PDC was fetched into the B channel of the SR620. The time difference between the channel A and B was measured for all PDC delay line settings.

Measurement took 54 hours, pseudo-random delay words were intermixed with zero delay settings. Individual element delays have been computed from the overdetermined set of bit word to delay correspondences by least mean square fit, results are plotted in Fig. 6. Total operational line delay is $T = 8.72$ ns. Maximum step size (resolution) between adjacent delay words is $\Delta\tau_{max} = 14.9$ ps.

Temperature in a vicinity of the FPGA chip has been measured as well. Fig. 7 shows temperature variation during experiment together with offset and PDC scale variations. In linear approximation, the offset drift is 23 ps K^{-1} , the scale drift is 8.4×10^{-4} K^{-1} , yielding 30 ps K^{-1} in the worst case.

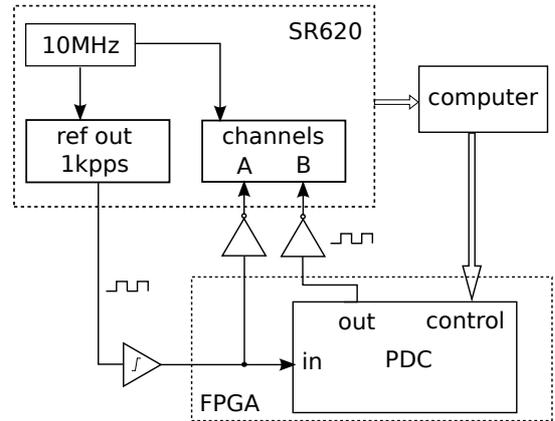


Fig. 5. PDC line resolution and linearity measurement setup

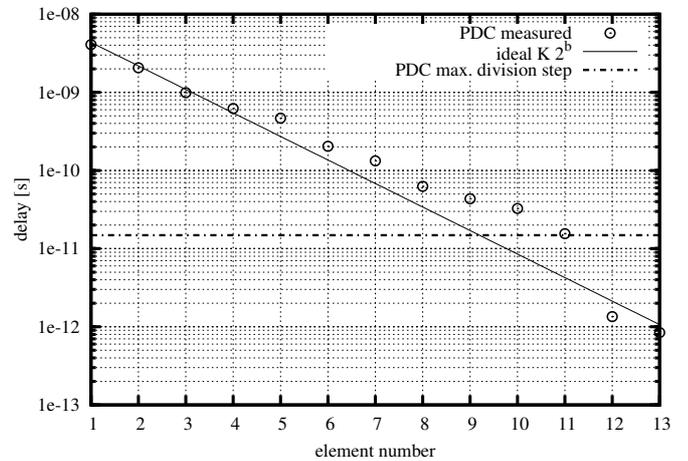


Fig. 6. PDC line resolution and linearity

The propagation jitter of the PDC was determined with the measurement setup depicted in Fig. 8. For precise jitter measurements, NPET device is employed having single shot precision of $\sigma_{NPET} = 1.0$ ps RMS [9]. The NPET generates 100pps reference signal ($\sigma_{ref} = 1.2$ ps RMS) being synchronous with its internal clock; the reference is fed into the PDC input. The NPET measures the jitter of the reference signal propagated through the PDC for different delay settings; each single delay element is enabled at a time.

Propagation jitter of the delay line ranged from 3.9 ps RMS to 5.0 ps RMS (all delays on) for different line configurations. A jitter of signal mirrored at the output of the FPGA, but not passing through PDC, was measured 2.9 ps RMS. Under assumption of independent, normal noises, jitter $\sigma_{max} = 4.1$ ps RMS may be attributed to delay line in its longest configuration.

B. Frequency synthesis

Generation of $f_o = (2/3)f_i$ frequency has been performed in order to prove the concept of control logic. Input and output waveforms acquired are shown in oscilloscope screen shot Fig. 9. Waveform distortion is caused by coarse line length

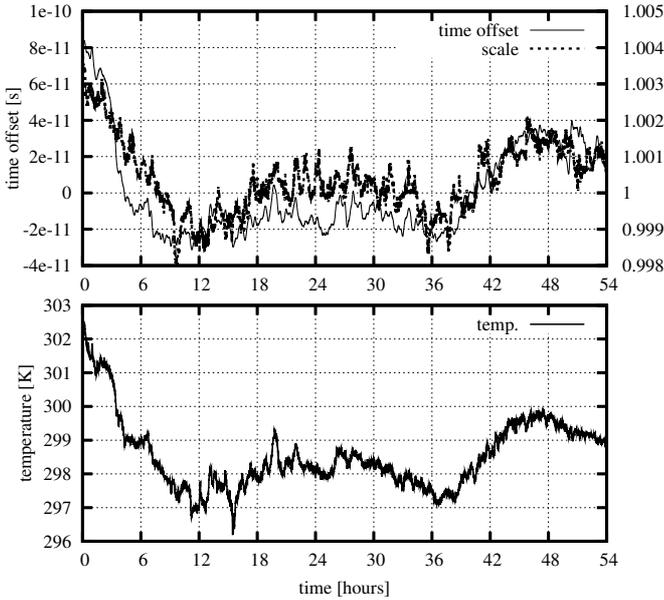


Fig. 7. Temperature and drift of PDC offset and scale

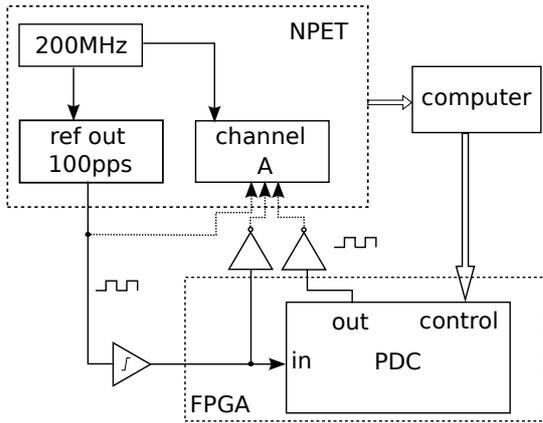


Fig. 8. PDC delay line elements jitter measurement setup

(5 bits) and unoptimized delay words.

V. CONCLUSION

A presumably novel type of PDC has been presented. The delay line cascaded of 13 elements exhibits a deterministic resolution $\pm\Delta\tau_{max}/2 = \pm 7.5$ ps, random jitter $\sigma_{max} = 4.1$ ps RMS, temperature drift $8.4 \times 10^{-4} \text{ K}^{-1}$, drift including offset up to 30 ps K^{-1} . The performance is comparable to recent works [2], as well as to dedicated ECL circuits [5]. Further improvement of resolution is possible by adding more delay elements. Temperature compensation seems plausible by a measurement of reference delay line element, e. g. using ring oscillator, or Time-to-Digital Converter [11].

An ability to generate waveform differing in frequency and phase from the input signal has been demonstrated as a proof-of-concept. The effect of idle delay is severely limiting factor here, especially in an FPGA. We would like to examine pos-

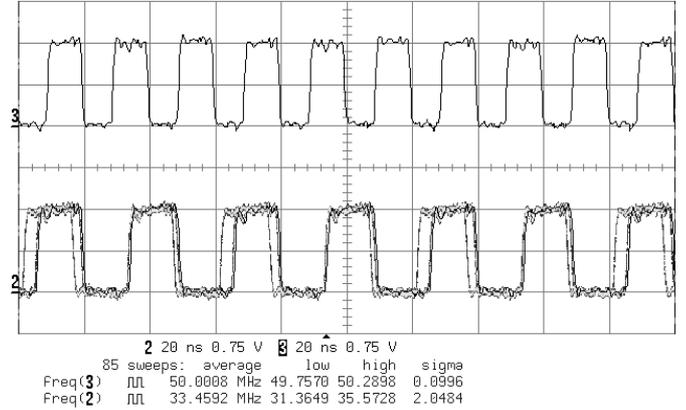


Fig. 9. Frequency synthesis by rapidly switching PDC

sible improvement by combining tree-like structure for coarse line with cascaded structure for the fine part. An implementation into ASIC shall yield significantly better performance, due to much lower multiplexer delays and unnecessary of differential delay elements.

REFERENCES

- [1] Z. L. M. Li, J. and Zheng and S. Wu, Dynamic range accurate digitally programmable delay line with 250-ps resolution, in 8th International Conference on Signal Processing, 2006.
- [2] A. de Castro and E. Todorovich, High resolution fpga dpwm based on variable clock phase shifting, IEEE Transactions on Power Electronics, vol. 25, pp. 1115–1119, 2010.
- [3] S. C. Huerta, A. de Castro, O. Garcia, and J. A. Cobos, Fpga-based digital pulsewidth modulator with time resolution under 2 ns, IEEE Transactions on Power Electronics, vol. 23, pp. 3135–3141, 2008.
- [4] J. Kalisz, A. Poniecki, and K. Ryc, A simple, precise, and low jitter delay/gate generator, Review of Scientific Instruments, vol. 74, 2003.
- [5] MC100EP195: 3.3 V ECL Programmable Delay Chip, ON Semiconductor. [Online]. Available: <http://www.onsemi.com/PowerSolutions/product.do?id=MC100EP195>
- [6] M.-A. Daigneault and J. P. David, A high-resolution time-to-digital converter on fpga using dynamic reconfiguration, IEEE Transactions on Instrumentation and Measurement, vol. 60, pp. 2070–2079, 2011.
- [7] Standard Delay Format Specification, version 3.0 (1995), Open Verilog International. [Online]. Available: http://www.eda.org/sdf/sdf_3.0.pdf
- [8] S. Chan, Programmable delay line using configurable logic block, U.S. Patent 7049845 B1, 2004. [Online]. Available: <http://www.freepatentsonline.com/7049845.html>
- [9] P. Panek, I. Prochazka, Time interval measurement device based on SAW filter excitation, IEEE Review of Scientific Instruments, 2004
- [10] SR620: Time interval and frequency counter, Stanford Research Systems. [Online]. Available: <http://www.thinksrs.com/products/SR620.htm>
- [11] Peca, M.; Vacek, M.; Michalek, V., "Time-to-Digit Converter Based on radiation-tolerant FPGA," European Frequency and Time Forum (EFTF), 2012, vol., no., pp.286,289, 23-27 April 2012 doi: 10.1109/EFTF.2012.6502384