Measurement 92 (2016) 103–113

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

# Multiple-tapped-delay-line hardware-linearisation technique based on wire load regulation

Dariusz Chaberski [a,*], Robert Frankowski [a], Marek Zieliński [a], Łukasz Zaworski [b]

[a] Institute of Physics, Faculty of Physics, Astronomy and Informatics, Nicolaus Copernicus University, Grudziądzka 5, 87-100 Toruń, Poland
[b] Apator S.A., Gdańska 4A, lok. C4, 87-100 Toruń, Poland

ARTICLE INFO

*Article history:*
Received 3 November 2014
Received in revised form 20 May 2016
Accepted 7 June 2016
Available online 8 June 2016

*Keywords:*
FPGA
LUT
CLB
Nonlinearity
TDL
VHDL
Time-stamp
Time-interval
QNM
MTDL
TIMM
TTDL
FTTL

ABSTRACT

This article describes designing, implementation and tuning processes of multiple-tapped-delay-line (MTDL). Obtained MTDL can be implemented in various field-programmable-logic-devices (FPGA) devices and applied for time-to-digital-converters (TDC) construction. The task of tuning process is the tapped-delay-line (TDL) linearisation, and consists of two stages. The first stage depends on selecting an appropriate configurable-logic-block (CLB) for particular delay-segment realization and selecting proper connection between these blocks. The second tuning stage, that is essential from this article viewpoint, depends on inter CLBs connecting wires delay regulation realized directly by load regulation. The Load regulation depends on connecting an appropriate number of unused three-state-buffers or CLB inputs to the wire which delay is adjusted. Depending on the number of inputs connected to the wire its capacitance changes that influences its time-constant and finally changes its time-delay.

The MTDL mathematical model, obtained characteristics and results of time-interval (TI) measurements are also presented. The derived TDL model provides information about how the particular wire delay should be changed and in which order the changes should be executed. This makes the designing process predictable and easy to carry out. Presented characteristics confirm the proper operation of presented linearisation technique. The proper operation of the whole measuring module is confirmed by obtained TIs histograms presentation.



## 1. Introduction

TDCs can have many applications and can be realized in many ways [1–3]. TDCs can be used directly or indirectly to measure other quantities by converting them temporarily to TIs [3–5]. Depending on use-domain and implementation-technology TDCs can vary in properties and can differ in cost [3,6]. The most important parameters of TDCs are TI measurement-resolution, measurement-range and maximal intensity of registered time-stamps (TS) [3,7–9].

The measurement resolution of TDC can be improved by applying the vernier method [9–18] the interpolation and the multiple-stage interpolation [13,19–22], the time-stretching [1,12,20,21] or multiple-measurement [13,23–25]. The first two mentioned methods utilize TDLs and the TDC resolution is determined by TDL segment-delay value [1,7,12]. The TDL segment-delay value can be decreased either by using MTDL or by changing implementation technology to the finer one [1,26]. The time-stretching decreases the maximal intensity of registered TSs but can be applied with connection with any other method [1,2]. Multiple-measurement can also be connected with any other method and can be realized either parallelly or sequentially.

The measurement range can be extended by standard-clock period counter implementation [1,7,13,27,28], but is limited by long-term standard-clock stability [29,30]. The maximal-intensity of registered TSs usually can be increased by parallelization of critical TDC blocks or by changing implementation technology to the faster one. Currently the FPGA and the application-specific-integrated-circuit (ASIC) are two technologies that play an important role for TDC implementation [1,2,7,8,12,14,16,27,31–44]. Both these technologies allows to implement regularly-placed elements of equal delay that are necessary to construct TDLs [7,9,26,38].

## 2. Motivation

Implementation of TDL requires many regularly-placed blocks. The sum of all tap delay values should be equal to or should be

* Corresponding author.
*E-mail addresses:* daras@fizyka.umk.pl (D. Chaberski), robef@fizyka.umk.pl (R. Frankowski), marziel@fizyka.umk.pl (M. Zieliński), lukasz.zaworski@gmail.com (Ł. Zaworski).

http://dx.doi.org/10.1016/j.measurement.2016.06.002
0263-2241/

larger than the standard-clock period. This condition is sometimes difficult to be met especially when segment delay values are small (e.g. carry-chain where $\tau \simeq 50$ ps) and FPGA structure is not very large (e.g. maximally 32 CLBs in one column or row). In such cases the required standard-clock frequency might be too high to be introduced into FPGA structure. This problem can partially be solved by making the code-converter sensitive to any edge (rising and falling), then the required standard-clock frequency can be reduced twice, but the standard-clock duty must be very close to 0.5. Additionally complementary-metal-oxide-semiconductor (CMOS) consumes more power when the clock frequency is higher [45]. Lowering the standard-clock frequency allows for lowering the supply power requirements of the constructed TDC.

Another possibility of obtaining simultaneously high-resolution and standard-clock match is MTDL usage. The MTDL that uses $m$ TDLs allows to increase resolution up to $m$ times, but does not requires $m$ times greater (in one axis) FPGA structure. In the approach presented in this article the MTDL is calibrated and the characteristic is always predictable. Here, the TDL resolution always increases $m$ times relatively to resolution obtained when one TDL is used. It means that when $n$-segment TDLs are used then the average resolution increase (relatively to standard-clock period) is always equal to $mn$, where $m$ is the number of used TDLs. For the solution presented in [46] the resolution increase can be equal to or can be less than $mn$ and for the solution presented in [24] the maximal obtained resolution increase is less than or equal to $nm - m + 1$.

Usually described in very-high-speed-integrated-circuits-hard ware-description-language (VHDL) MTDLs would posses originally high relative nonlinearity comparing with single TDLs. To obtain both high-resolution and high-linearity the special linearisation-process occurred to be required. In the literature one can find many examples of TDC characteristic linearisation or TSs calibration methods. Both these operations aim to obtain lower uncertainty of measurements. In case of TDCs implemented in FPGAs or ASICs, the multi-path delay line [47,48] or TDLs converted to ring-oscillators [49] are often used as a calibration solutions. For ASIC implemented TDCs additionally the linearisation techniques include incrementing the capacitive load of the regulated TDL segment [50] or TDL segment supply voltage regulation [15,43]. For TS calibration one can use either neural-networks [51] or calibration to the centers of the TDC bins [1,52]. In this article the linearisation process depends on TDL segment-connecting-wire load regulation that effectively reduces to incrementing the capacitive load of the regulated TDL segment, similarly as it is applied for TDCs implemented in ASICs [50]. This method of characteristic linearisation has not been yet used for TDCs implemented in FPGAs. Presented in this article linearisation methodology allows for effective MTDL characteristic regulation, that significantly decreases the time needed for hardware linearisation process of the MDTLs implemented in FPGAs. The linearisation methodology presented in this article perfectly fits for implementing TDLs or MTDLs in the FPGA structures with proper thermometric code without any bubbles or sparkles [50]. The combination of MTDL and linearisation-process allowed obtaining linear and relatively high resolution TDL.

## 3. The model of multiple-tapped-delay-line

The MTDL consists of $m$ of $n$-segment TDLs and one (vertically-placed) $m$-segment TDL of high-resolution (Fig. 1). The purpose of vertical high-resolution TDL is to deliver to each horizontal TDL input properly shifted in time standard-clock signals. This solution requires only one and relatively short high-resolution TDL. The sum of all vertical high-resolution TDL delay-segment values must merely be equal to one segment horizontal TDL delay. All horizontal TDLs can be implemented by using easily accessible blocks like look-up-tables (LUT) in the FPGA structure.

In case when twofold-TDL (TTDL) is being used ($m = 2$) segment-delay value can be expressed as

$$\tau_i = \frac{\tau}{2} + \sum_{j=0}^{i-1} (-1)^{j+i+1} \delta_j, \; i \in [1, 2n-1]. \tag{1}$$

For fourfold-TDL (FTDL) ($m = 4$) the segment-delay value can be written in a similar way, so

$$\tau_i = \frac{\tau}{4} + \sum_{j=0}^{\left[\frac{i+2}{4}\right]+\left[\frac{i+3}{4}\right]-1} (-1)^j \delta_{(i-1-j \bmod 2) \bmod 4 + 4\left[\frac{j}{2}\right]}, \; i \in [1, 4n-1]. \tag{2}$$

Eqs. (1) and (2) can simply be derived from MTDL model presented in Fig. 1. They will be utilized in Section 5 for TTDL and FTDL characteristic correction.

## 4. Implementation of MTDL

Presented in Section 3 MTDL has been described in VHDL and successive placements of particular segments have been restricted by using user-constraints-file (UCF). Delay-segments have been created by using FPGA LUT elements. The LUT elements have been added to the project by using VHDL `port` and `generic map` constructions. The VHDL code

```
CLB0: lut1 generic map("10")\
        port map(clock, clock_common);

CLB2: lut1 generic map("10")\
        port map(clock_common, clock_even(0));
CLB4: lut1 generic map("10")\
        port map(clock_even(0), clock_even(1));
        ...
CLB32: lut1 generic map("10")\
        port map(clock_even(14), clock_even(15));

CLB1: lut1 generic map("10")\
        port map(clock_common, clock_odd(0));
CLB3: lut1 generic map("10")\
        port map(clock_odd(0), clock_odd(1));
        ...
CLB31: lut1 generic map("10")\
        port map(clock_odd(14), clock_odd(15));
```

implements 32 segment TTDL.

The `generic map (''10")` construction gives the information about the function performed by LUT. In this case it is simply a buffer, because '`0`' is placed at element number 0, and '`1`' is placed at element number 1. In case when the map argument would be equal to '`01`' one could implement an inverter. The state from input, after parasitical delay, is transferred directly to the output. All connections are being done by `port map` construction. The first argument determines the signal that is connected to the LUT input, the second one states the output signal. The placement of each LUT