International Journal of Engineering Research and General Science Volume 3, Issue 2, Part 2, March-April, 2015
ISSN 2091-2730

DESIGN AND ANALYSIS OF GLITCH FREE NAND BASED DIGITALLY CONTROLLED DELAY LINES USED TO SPREAD SPECTRUM CLOCK GENERATOR

POORNIMA M, SARANYA S

Graduate student, M.E – Applied Electronics, DSCET, ANNA UNIVERSITY, Tamilnadu, INDIA

poornimamuruganandam@gmail.com +91 9788033166

Abstract— This paper presents a glitch free NAND based digitally controlled delay lines (DCDL) by analyzing the glitching problem of existing NAND based DCDL. The proposed glitch free NAND based 32 and 64 bit DCDL overcome this limitation by opening their use in many applications. This NAND based 32 and 64 bit DCDL also maintains the same resolution and minimum delay of previously proposed NAND based DCDL. In this paper the proposed 32 and 64 bit DCDL is used to realize an All-digital spread- spectrum clock generator (SSCG). The use of proposed DCDL in this circuit allows reducing the peak- to –peak absolute output jitter of more than 40% with respect to SSCG using three state inverter based DCDLS. From a more practical point of view, nowadays, DCDLS are a key block in a number of applications, like all-digital PLL (ADPLL), all-digital DLL (ADDLL), all-digital spread-spectrum clock generators (SSCGs) and ultra-wide band (UWB) receivers.

Keywords— All-DIGITAL DELAY-LOCKED LOOP (ADDLL), ALL-DIGITAL PHASE-LOCKED LOOP (ADPLL), DELAY-LINES, DIGITALLY CONTROLLED OSCILLATOR (DCO), FLIP-FLOPS, SENSE AMPLIFIER, SPREAD-SPECTRUM CLOCK GENERATOR (SSCG).

INTRODUCTION

The Digitally Controlled Delay Line (DCDL) is a digital circuit whose delay is controlled by a digital control word. The Digitally Controlled Delay Line is designed previously using two methods. They are

- TINV based DCDL
- MUX based DCDL

A Multiplexer is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. In the TINV based DCDLs the tristate inverters are used to create delay which is controlled by an input of tristate inverter. The tristate inverter will only allow the inverter to drive the bus if EN is high. In these MUX-based DCDLs, the MUX delay increases with the increase of the number of cells. This results in a tradeoff between the delay range and minimum delay t_{min} of the DCDL. The large t_{min} of MUX-based DCDLS can be reduced by using a tree-based multiplexer topology. This however results in an irregular structure which complicates layout design and consequently, also increases the nonlinearity of the DCDL. The DCDL uses again a structure of cascaded delay elements. Differently each element is constructed by using three state inverters (TINV), obtaining a resolution $t_R = 2t_{INV}$. Since the pull-up network of a TINV requires two series devices whereas a NAND gate uses a single device in the pull-up network. In next case each delay element is constructed by using an inverter and an inverting multiplexer, this topology creates two drawbacks first is due to the different delays of the inverter and the multiplexer which results in a mismatch between odd and even control-codes. Second drawback is due to the large multiplexer delay, which provides a resolution of both NAND based DCDLs and TINV based DCDLs.

The proposed system of NAND based DCDL design has certain considerations. Glitching is a common design problem in systems employing DCDLS. In the most common applications, DCDLs are employed to process clock signals; therefore a glitch free

operation is required. A necessary condition to avoid glitching is designing a DCDL which have no glitch in presence of a delay control-code switching. This is an issue at the DCDL design level.

METHODOLOGY

Analysis of NAND based DCDL and Glitching

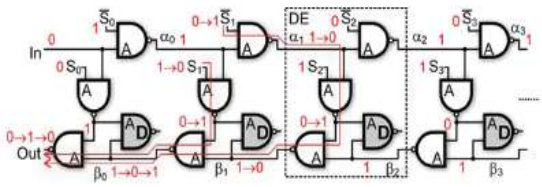


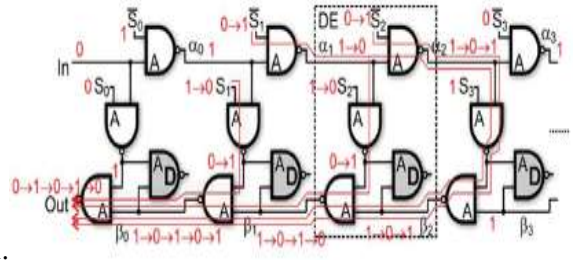
Fig2.1 Glitching when the delay control code increases by one.

The circuit is composed by a series of equal delay- elements (DE), each composed by four NAND gates. In the figure2.1 "A" denotes the fast input of each NAND gate. Gates marked with "D" are dummy cells added for load balancing. The delay of the circuit is controlled through control-bits Si, which encode the delay control-code with a thermometric code: $S_i = 0$ for $i < c$ and $S_i = 1$for $i >= c$. By using this encoding, each DE in figure2.1 can be either in pass- state or in turn-state. In figure2.1 all NAND gates present the same load (two NAND gates) and, therefore in a first order approximation, present the same delay. This consideration allows writing the delay ∂, from *In* to *out*, as follows:

$$\partial = 2t_{NAND} + 2t_{NAND}.C$$

It is interesting to observe that it holds both for low-to-high and high-to-low out commutations. In DCDL applications, to avoid DCDL output glitching, the switching of delay control-bits is synchronized with the switching of *in* input signal. Glitching is avoided if the control-bits arrival time is lower than the arrival time of the input signal of the first DE which switches from pass-state or to the turn-state. Unfortunately in the DCDL of figure2.1 this condition is not sufficient to avoid glitching .In this circuit,it is possible to have output glitches also considering only the control-bits switching, with a stable input signal. Some examples of glitching problems of this DCDL are highlighting in figure2.1.

Let us name the vector of the control-bits of the DCDL. In figure2.1 it is assumed that and that the control-code of the DCDL is switched from 1to 2. It can be easily verified that the same glitching behavior exists when input *In* is 1, and the delay control-code is increased by 1 starting from an even value and the delay control-code is increased by



1.

Fig2.2 Glitching when the delay control code increases by two.

Figure2.2 shows that the structure exhibits a more severe glitching problem when the delay control-code is increased by more than 1.In particular the figure2.2 considers the case in which control-code of the DCDL is switched from 1 to 3.The analysis of the figure2.2, in this case, reveals that, in the worst case, four paths propagate within the DCDL structure and may create a multiple-glitch at the delay-line output.

More in general the glitching problem of NAND-based DCDL grows up because, for a control-code equal to c, all α_i and β_i signals in figure2.2, which are stuck-at 1, while for, the logic state of signals depends on the input signals *In*. When the control code is increased, the logic state of the output becomes dependant on a portion of the DCDL for which and switch from 1 to a logic state dependant on In.

This switching may determine output glitches. This consideration also demonstrates that no glitching can occur when the control-code is decreased. When the control-code is increased by one, the glitching problem could be avoided by delaying S_i signals with respect to $S_{i\ bar}$ signals. This solution, however, does not solve the glitching problem when the control-code is increased by more than one.

EXAMPLE FOR GLITCHING PROBLEM

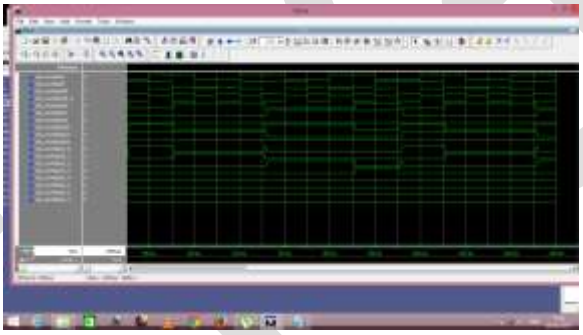


Fig2.3 Simulation Result-Glitch present in ripple counter

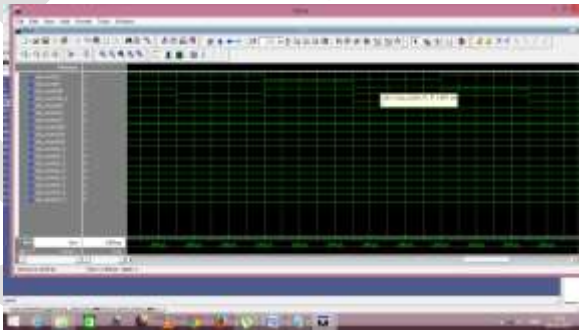


Fig2.4 Simulation Result-Glitch free in ripple counter

PROPOSED SCHEME

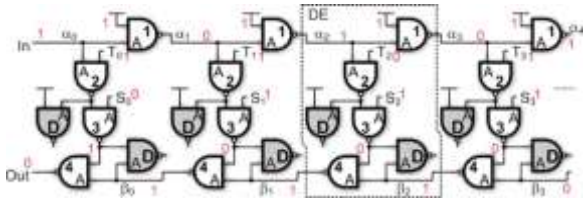


Fig3 Glitch free NAND based DCDL

The structure of proposed DCDL is shown in fig 3. In this figure3 "A" denotes the fast input of each NAND gate. Gates marked with "D" represents dummy cells added for load balancing. Two sets of control-bits, si and Ti, control the DCDL. The si bits encode the control code c by using a thermometric code: si = 0 for i < c and si = 1 for i ≥ c. The bits Ti encode again c by using a one control code: Tc+1=0, Ti=1, for i ≠ C+1. The fig 3 shows the state of all signals in the case In = 1, c = 1.According to the chosen control-bits encoding, each delay element (DE) can be in one of three possible states.

The Des with i < c are in pass state (si = 0, Ti=1).In this state the NAND "3" output is equal to 1 and the NAND "4" allows the signal propogation in the lower NAND gates chain.The DE with i = c is in turn-state (si = Ti = 1).In this state the upper input of the DE is passed to the output of NAND "3". The next DE (i = c+1) is in post turn state (si = 1, Ti = 0). In this DE the output of the NAND "4" is stuck- at 1, by allowing the propagation, in the previous DE (which is in turn state), of the output of NAND "3" through NAND "4". All remaining DEs (for i > c+1) are again in turn-state (si = Ti =1).The three possible DE states of proposed DCDL and the corresponding values are summarized in Table I.

Si	Ti	DE State
0	1	Pass
1	1	Turn
1	0	Post-Turn

TABLE I LOGIC-STATES OF EACH DE IN PROPOSED DCDLS

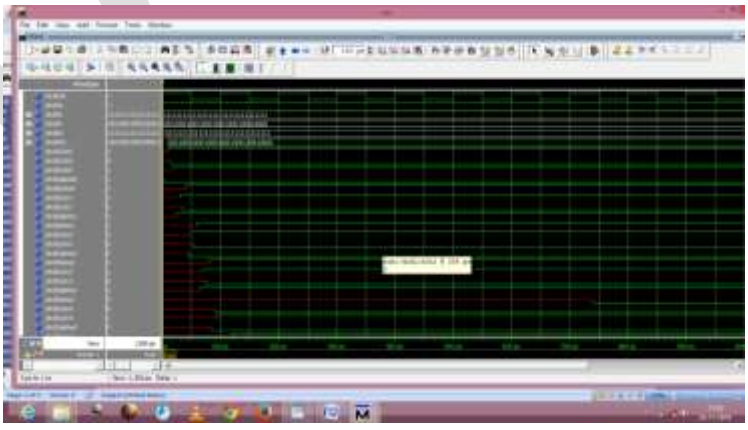


Fig3.1 Simulation Result-32 bit DCDL

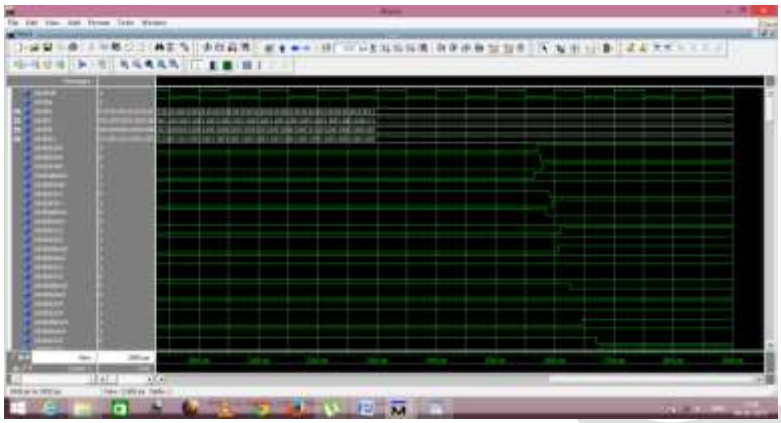


Fig3.2 Simulation Result-64 bit DCDL

SSCG APPLICATION

The SSCG proposed in fig is basically composed of aDigital processor and a delay line block, including two digitally controlled delay lines (DCDLs).

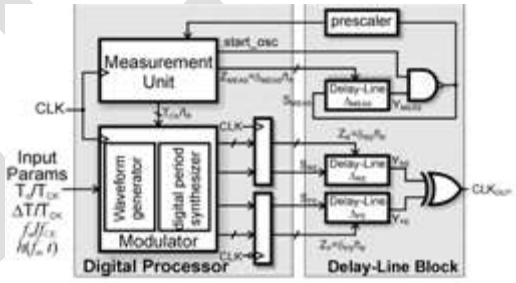


Fig4.1 Architecture of spread spectrum clock generator

A.MODULATOR

1. WAVEFORM GENERATOR

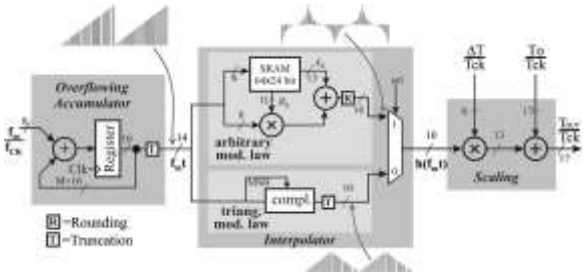


Fig4.2 Architecture of waveform generator

Waveform generator is used to compute the desired instantaneous period of the output waveform, normalized to the clock period: TOUT/TCK. An M-bit overflowing accumulator is used to generate a saw tooth waveform with a frequency fm.

The modulation frequency is set with a precision of fCK/2M by using the control word fm/fCK. In this design M=16.The output of the overflowing accumulator corresponds to the argument fmt of the function h, and is truncated to 14 bits to simplify the Interpolator implementation. From a practical point of view, there is no use to have a modulation Frequency much larger than the maximum Resolution Bandwidth (RBW-1 MHz) considered in EMI standards. So the number of input bits used for fm/fCK is limited to 8.

The interpolator in is able to generate an arbitrary waveform h(fmt).It is to be noted that interpolator can be largely simplified if only triangular modulation profile is required. In this case, In fact, the SRAM, the multiplier and the adder can be eliminated from the circuit and replaced by a 1's complementary.

2. DIGITAL PERIOD SYNTHESIZER:

The digital period synthesizer generates the input signals for the two delay lines ΔRE and ΔFE. It is obtained with the help of a finite state machine (FSM). The outputs of the FSM (WF and WR) drive the two delay lines ΔRE and ΔFE through a scaling block.

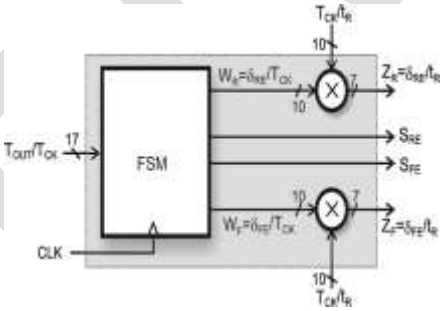


Fig4.3 Architecture of Digital Period Synthesizer

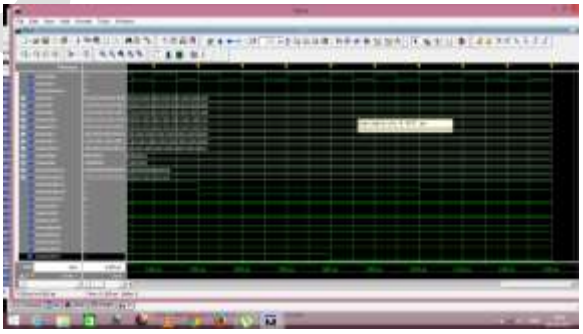


Fig 4.3 Simulation result-SSCG using DCDL

CONCLUSION

A glitch free NAND based 32 and 64 bit DCDL which avoids the glitching problem of previous circuit has been presented. The developed model also provides the timing constraints that need to be imposed on the DCDL control bits, in order to guarantee a glitch free operation. The proposed DCDL is used to realize SSCG. The use of proposed DCDL allowed to reduce the peak-to-peak absolute output jitter of more than 40% with respect to an SSCG using other DCDLS.

REFFERENCE:

[1] R. B. Staszewski and P. T. Balsara, All Digital Frequency Synthesizer in Deep Submicron CMOS. New York: Wiley, 2006.

[2] C. C. Chung and C. Y. Lee, "An all-digital phase-locked loop for high speed clock generation," IEEE J. Solid-State Circuits, vol. 38, no. 2, pp. 347–351, Feb. 2003.

[3] P. L. Chen, C. C. Chung, and C.Y. Lee, "A portable digitally controlled oscillator using novel varactors," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 52, no. 5, pp. 233–237, May 2005.

[4] P. L. Chen, C. C. Chung, J. N. Yang, and C. Y. Lee, "A clock generator with cascaded dynamic frequency counting loops for wide multiplication range applications," IEEE J. Solid-State Circuits, vol. 41, no. 6, pp. 1275–1285, Jun. 2006.

Website Reference

1) en.wikipedia.org
2) www.Mathworks.Com