Design and Implementation of Glitch-Free NAND-Based Digitally Controlled Delay-Lines

SHIVA MADHUNAIK
M.Tech , Department of IT
Dayananda Sagar College of Engineering
Bangalore, India
shivamadhunaik@gmail.com

THYAGARAJ S
M.Tech Department of IT,
Dayananda Sagar College of Engineering
Bangalore, India
thyagaraj.s.shekar@gmail.com

PARAMESHWAR REDDY
M.Tech ,Department of IT,
Dayananda Sagar College of Engineering
Bangalore, India
parameshec72@gmail.com

***Abstract*— Digitally controlled delay line is a digital circuit used to provide the desired delays. Glitches are the most considerable factor that limits the use of DCDL in many applications such as DLL and clock generators. The NAND-Based circuit eliminates the glitches. The recently proposed NAND-based digitally controlled delay-lines (DCDL) present a glitching problem which may limit their employ in many applications. This paper presents a glitch-free NAND-based DCDL which overcame this limitation by opening the employ of NAND-based DCDLs in a wide range of applications .The proposed NAND-based DCDL maintains the same resolution and minimum delay of previously proposed NAND-based DCDL. Simulation results show that novel circuits result in the lowest resolution, with a little worsening of the minimum delay with respect to the previously proposed DCDL with the lowest delay.**

Keywords: Digitally controlled delay lines (DCDL) All-digital delay-locked loop (ADDLL), all-digital phase-locked loop (ADPLL), delay-line, digitally controlled oscillator (DCO), spread-spectrum clock generator (SSCG).

I. INTRODUCTION

Digital circuits are easy to handle compared to analog circuits and also digital circuits requires low power. Time-domain resolution of a digital signal is superior to voltage resolution of analog signals. DCDL is most important factor in the applications like PLL, ADDLL and ultra wide band receivers. In Recent deep-sub micrometer CMOS processes, time-domain resolution of a digital signal is becoming higher than voltage resolution of analog signals. This claim is now a days pushing toward a new circuit design paradigm in which the traditional analog signal processing is expected to be progressively substituted by the processing of times in the digital domain .With in this novel paradigm, digitally controlled delay lines (DCDL) should play the role of digital-to-analog converters in traditional, analog-intensive, circuits. 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.

The "classical" approach to design a DCDL is using a delay-cells chain and a MUX to select the desired cell output. In these mux-based DCDLs, if the number of cells increases then the MUX delay also increases. To obtain the good linearity and resolution the delay elements are constructed by using NAND gates. The DCDL proposed is based on a cascade of equal delay elements.

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. Many approaches are known to avoid glitching in mux-based DCDLs. It is interesting to observe that the DCDL topologies from a logical point of view, correspond to distributed MUX-based structure. This paper gives two contributions to the design of NAND-based DCDLs. First it is shown and analyzed the glitching problem of the NAND-based DCDL. Afterwards a novel glitch-free NAND-based DCDL is presented. The proposed NAND-based DCDL allows to achieve a resolution t_R=2. t_{NAND}.

The paper is organized as follows. The NAND-based DCDL is recalled in Section II. In the same section the glitching problem of this DCDL is analyzed. The structure of proposed, glitch-free, NAND-based DCDL is presented in Section III. Section IV analyzes theoretically the novel DCDL structure by deriving the conditions (timing constraints) needed to avoid glitching in proposed circuit.

II. PREVIOUSLY PROPOSED NAND-BASED DCDL AND GLITCHING

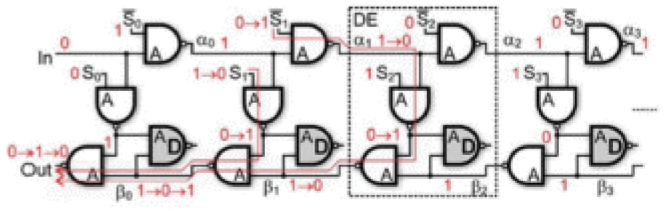

FIG 1(a).Glitching when the delay control-code increases by one

Fig. 1(a) shows the NAND-based DCDL. The circuit is composed by a series of equal delay-elements (DE), each composed by four NAND gates. In the figure "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 S_i, which encode the delay

control-code c with a thermometric code: $S_i=0$ for $i < c$ and $S_i=1$ for $i \geq c$. By using this encoding, each DE in Fig. 1(a) can be either in pass-state ($S_i=0$) or in turn-state ($S_i=1$) . In Fig. 1(a) all NAND gates present the same load (two NAND gates) and, therefore, in a first order approximation, present the same delay. This consideration allows to write the delay δ, from *In* to *Out*, as follows:

$$\delta = 2t_{NAND} + 2\, t_{NAND} \cdot c \qquad (1)$$

where $t_{NAND} = (t_{NAND\ LH} + t_{NAND\ HL})/2$ while $t_{NAND\ LH}$ and $t_{NAND\ HL}$ represent the delay of each NAND gate for a low-to-high and high-to-low output commutation, respectively. It is interesting to observe that (1) holds both for low-to-high and high-to-low *Out* commutations. Equations (1) suggests that $t_{min} = 2t_{NAND}$ and $t_R = 2t_{NAND}$.

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 or to the turn-state. Unfortunately in the DCDL of Fig. 1(a) this condition is not sufficient to avoid glitching. In this circuit, in fact, 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 Fig. 1. Let us name **S** = [S_0, S_1,……] the vector of the control-bits of the DCDL. In Fig. 1(a) it is assumed that *In = 0* and that the control-code of the DCDL is switched from 1(**S** = [0, 1, 1, 1, ……]) to 2(**S**= [0, 0, 1, 1, …..]) . Please note that, within the structure, the switching of S1 and Ś1 and results in two different paths that generate an output glitch. 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.

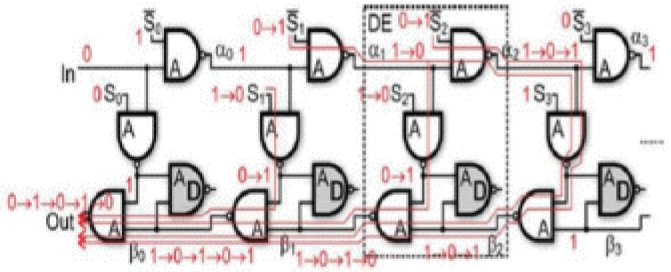

FIG 1(b).Glitching when the delay control-code increases by two

Fig. 1(b) shows that the structure exhibits a more severe glitching problem when the delay control-code is increased by more than 1. In particular the Fig. 1(b) considers the case in which control-code c of the DCDL is switched from 1(**S** = [0, 1, 1, 1, ……]) to 3(**S**= [0, 0, 0, 1, …..]) . The analysis of the figure, 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 Fig. 1, with $i \geq c$, are stuck-at 1 , while for $i < c$, the logic state of α_i and β_i signals depends on the input *In*. When the control-code is increased, the logic state of the output becomes dependant on a portion of the DCDL for which α_i and β_i 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.

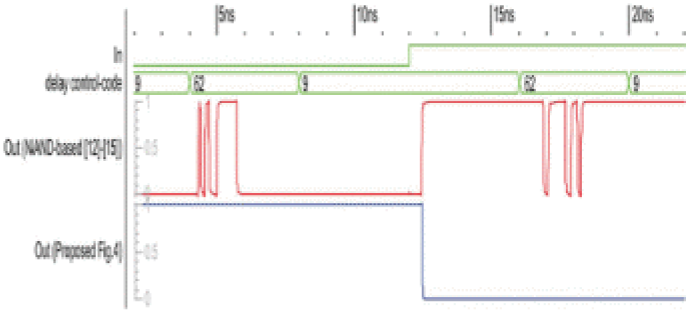

Fig 2. Transient simulations highlighting glitching problems of the NAND-based DCDL and the glitch-free operation of proposed DCDL.

The Fig. 2 shows a transistor-level simulation of a NAND based DCDL composed by 64 elements. In this simulation, first,*In = 0* . The control-code of the DCDL is firstly changed from 9 to 62 and, afterwards, changed back from 62 to 9. The DCDL output is reported in the third curve of the figure . When the code word is increased (62 to 9), it can be observed the presence of three glitches on the DCDL output. As observed before, when the code word is decreased(62 to 9) , no output glitching occurs. The second portion of the simulation shows that the same behavior can be observed for *In= 1* .

III. PROPOSED NAND-BASED DCDL

The structure of proposed DCDL is shown in Fig. 3. In this figure "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: $S_i = 0$ for $i < c$ and $S_i = 1$ for $i \geq c$ .The bits T_i encode again c by using a one-cold code:$T_{c+1} = 0$, $T_i = 1$ for *In* = 1, c = 1.. The Fig. 3 shows the state of all signals in the case , . According to the chosen control-bits encoding, each delay-element (DE) can be in one of three possible states.

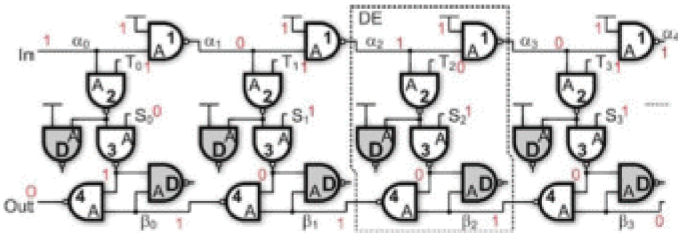

Fig. 3. glitch-free NAND-based DCDL (inverting topology).

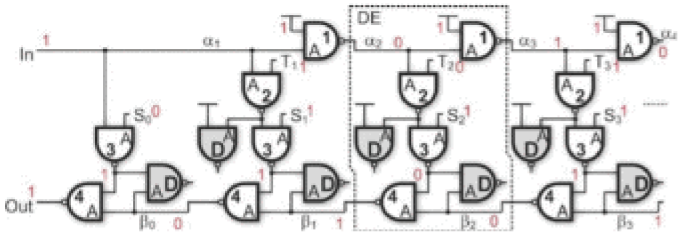

Fig. 4. glitch-free NAND-based DCDL (non-inverting topology).

TABLE I
LOGIC-STATES OF EACH DE IN PROPOSED DCDLS

S_i	T_i	DE state
0	1	*Pass*
1	1	*Turn*
1	0	*Post-Turn*

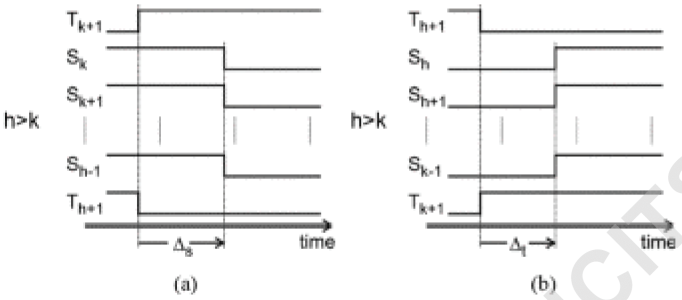

Fig. 5. Control-bits waveforms of DCDL for a control-code switch from c = k to c = h: (a) h > k case; (b) h< k case.

The DEs with i < c are in pass-state ($S_i = 0$, $T_i = 1$) . In this state the NAND "3" output is equal to 1 and the NAND "4" allows the signal propagation in the lower NAND gates chain. The DE with i = c is in turn-state ($S_i = T_i = 1$). 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 ($S_i = 1$, $T_i = 0$) . 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 ($S_i = T_i = 1$). The three possible DE states of proposed DCDL and the corresponding S_i and T_i values are summarized in Table I.

In the proposed DCDL the state of all α_i and β_i signals depends on the input ($\alpha_{2i} = \beta_{2i} = In$ and $\alpha_{2i+1} = \beta_{2i+1} = \check{I}n$) with the only exception of β_c , which is stuck-at 1. The glitch-free switching property of the proposed DCDL is conceptually simple to demonstrate. Let us assume a switching of the delay control-code from c = k to c = h. In the initial state of the line, $\alpha_{2i} = \beta_{2i} = In$ and $\alpha_{2i+1} = \beta_{2i+1} = \check{I}n$, with the exception of β_k, which is stuck-at 1. Let us suppose to first switch the k+ 1 th DE from the post-turn-state to the turn-state. By looking to Fig. 3 it can be observed that, in these conditions, β_k switches from 1 to α_k . The signal β_k is the input of the NAND "4" gate of kth DE. The switching of β_k is glitch-free since the other input of this gate is stuck-at αk , therefore the NAND "4" output remains equal to α_k . After the k+1 th DE switching, all cells are either in pass-state or in turn-state. In these conditions it is possible to freely change the state of Des from pass-state to turn-state, since this change does not affect the logic state of signals α_i and β_i. After this phase the h+1 th DE can be switched from turn-state to post-turn-state. This switching is again glitch free, since only β_h signal switches from α_h to 1. This procedure has the drawback to require a three-step switching of the DCDL. The following section provides a more detailed analysis of the glitching of proposed circuit in order to show that a glitch-free operation can also be achieved by using a properly designed two-step switching mechanism.

The last signal plotted in Fig. 2 is the output of proposed DCDL of Fig. 3, simulated by using the above described three step switching mechanism, and in the same conditions of the NAND-based DCDL. This simulation confirms that no glitching is obtained at the output of proposed DCDL.

The circuit of Fig. 3 is an inverting DCDL. In this circuit it is interesting to observe that the first DE is never in post-turn state, therefore T_0 is always 1 (see Table I). This observation allows to construct a non-inverting DCDL by modifying only the first DE, as shown in Fig. 4. In this circuit the NAND gates "1" and "2" of the first DE have been deleted, together with signal T_0. The signal $\alpha 1$ of the second DE is now equal to *In*, therefore the whole behavior of the DCDL is non-inverting. This topology maintains the same $t_R(2 \cdot t_{NAND})$ of previous solution, while it can easily verified that the minimum delay t_{min} is now 2. The non-inverting DCDL of Fig. 4, therefore, maintains the same performances of the NAND-based DCDL , while avoiding its glitching problem.

IV. GLITCH-FREE SWITCHING OF PROPOSED DCDL AND CONTROL-BITS DRIVING CIRCUIT

In the previous section we have seen that the glitch-free operation of the proposed DCDL can be obtained with a three-step switching mechanism: for a switching from a delay control code c= k to a delay control code c = h , first, the k=1th DE is switched from post turn-state to the turn-state; next all DE are switched from pass to turn-state (or vice versa) and finally the h + 1th DE is switched to post-turn-state. This switching mechanism presents the drawback of being slow and can result in a not simple driving circuit for the DCDL control-bits.

Actually, a more detailed analysis of proposed DCDL (see Appendix I), shows that a sufficient condition to achieve a glitch-free operation in proposed DCDL is imposing the following two timing constraints:

$$t_{SHL} - t_{T\,LH} > t_{NAND} \qquad (2)$$

$$t_{T\,HL} - t_{S\,LH} > -3t_{NAND} \qquad (3)$$

where t_{SHL} ,t_{SLH} , $t_{T\,HL}$ and $t_{T\,LH}$ represents the arrival times of HL and LH switching of T_i and S_i signals, respectively.

In order to show how this timing constraints can be, in practice, realized let us define two times, Δs and Δt, as follows:

$$\Delta_s = t_{S\,HL} - t_{T\,LH} \quad \Delta_t = t_{S\,LH} - t_{T\,HL} \quad (4)$$

By using the above definitions, the two timing constraints (2),(3) becomes

$$\Delta s > t_{NAND} \quad (5)$$
$$\Delta t < 3t_{NAND} \quad (6)$$

By using the above definitions, the Fig. 5 shows the waveforms of the DCDL control-bits for a switching of the DCDL delay, from a control-code c = k to a control-code c = h . The Fig. 5(a) considers the case in which the delay control-code needs to be increased (h > k). In this case the relevant timing constraint is (2). This constraint corresponds to imposing that, in Fig. 5(a), the delay Δs is greater than t_{NAND} [see (5)]. The Fig. 5(b), on the other hand, considers the case in which the delay control- code needs to be decreased (h < k). In this case the relevant timing constrain is (3). This constraint corresponds to imposing that, in Fig. 5(b), the delay Δ_t is lower than $3t_{NAND}$ [see (6)]. Please note that, formally, the choice $\Delta t = 0$ verifies this constraint. However, from a practical point of view, realizing a $\Delta_t < \Delta_s$ may be not obvious (see Fig. 5).

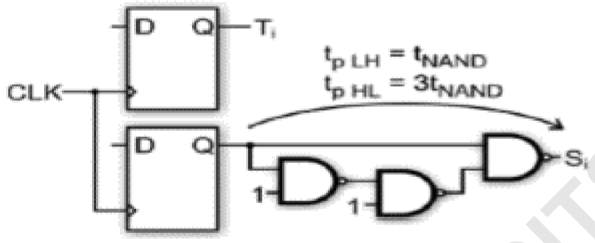

Fig. 6. Driving circuits for the control-bits of DCDL with S_i signals delayed with different LH/HL delays by using a NAND-based circuit

The Fig. 6 shows three possible driving circuits that can be used to generate the control-bits of the proposed DCDL, following the waveforms shown in Fig. 5. By analyzing Fig. 5 it can be noted that S_i signals have to be delayed with respect to T_i signals and that it could be useful to have a different delay for LH and HL transitions. It is also worth to note that, to avoid glitching of the DCDL, S_i and T_i signals must themselves be glitch-free. By following this reasoning, in all presented driving circuits, it is assumed that T_i signals are generated as output of flip-flops, which, at the same time, both properly time the DCDL considering system-level aspects, and act as deglitching elements.

The Fig. 6 shows the driving technique. In this solution each S_i signal is obtained by using a flip-flop followed by a NAND-based circuit which presents different LH and HL propagation delays($t_{p\,LH} \neq t_{p\,HL}$) . In this solution, as shown in the figure, $t_{p\,LH} = t_{NAND}$ while $t_{p\,HL} = 3t_{NAND}$. By using the definitions (4), we have

$$\Delta_s = 3t_{NAND} \quad \Delta_t = t_{NAND} \quad (7)$$

The two constrains (5) and (6) are therefore verified, both with a timing-margin of $2t_{NAND}$.

V. CONCLUSION

A Digitally Controlled Delay Lines plays an important role in many applications. Glitches are the drawback of this DCDL.A NAND-based DCDL which avoids the glitching problem has been presented. Control bits are used to control the Digitally Controlled Delay Line circuit. The simulation results confirm the correctness of developed model and show that proposed solutions improve the resolution with respect to previous approaches.

REFERENCES

[1] R. B. Staszewski, K.Muhammad, D. Leipold, C.-M. Hung, Y.-C. Ho, J. L. Wallberg, C. Fernando, K. Maggio, R. Staszewski, T. Jung, J. Koh, S. John, I. Y. Deng, V. Sarda, O. Moreira- Tamayo, V. Mayega, R. Katz, O. Friedman, O. E. Eliezer, E. de Obaldia, and P. T. Balsara, "All-digital TX frequency synthesizer and discrete-time receiver for bluetooth radio in 130-nm CMOS," *IEEE J. Solid State Circuits*, vol. 39, no. 12, pp. 2278–2291, Dec. 2004.

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

[3] 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.

[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.

[5] L.Wang, L. Liu, and H. Chen, "An implementation of fast-locking and wide-range 11-bit reversible SAR DLL," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 6, pp. 421–425, Jun. 2010.

[6] J. S. Wang, C. Y. Cheng, J. C. Liu, Y. C. Liu, and Y. M. Wang, "A duty cycle distortion tolerant half delay line low-power fast lock in all digital delay locked loop," *IEEE J. Solid-State Circuits*, vol. 45, no. 5, pp. 1036–1047, May 2010.

[7] S. Damphousse, K. Ouici, A. Rizki, and M. Mallinson, "All digital spread spectrum clock generator for EMI reduction," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 145–150, Jan. 2007.