ORIGINAL RESEARCH PAPER
VOLUME-6 | ISSUE-8 | AUGUST - 2017 | ISSN No 2277 - 8179 | IF : 4.176 | IC Value : 78.46
INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

IMPLEMENTATION OF FREQUENCY MULTIPLIER USING DPLL AND ITS BIST

Engineering

Degala Krishna Chitanya Tarun — M.Tech, Department of ECE, MVGR College of Engineering (A), Vizianagaram, A.P, India

D. Raja Ramesh — Assistant Professor, Department of ECE, MVGR College of Engineering (A), Vizianagaram, A.P, India

ABSTRACT

The clocks are required to synchronize between Integrated Circuits and functional blocks and mainly to progress the operations of digital systems for excellence in performance. For each architecture of the processor and the processing technology generation, the frequencies and data rates of clocks have been increasing gradually. In order to generate the on-chip clocks to have well-timed, Phase locked-loops (PLLs) can be broadly used. The bandwidth of the system loop can be changed easily in Fully Digital PLLs. When compared to the traditional designs such as charged-pump PLLs, All DPLLs are more suitable for the monolithic design with other circuits. Any variations in the process cannot disturb the functionality of the All-Digital PLLs and so can be applied easily to any technology. This DPLL is based on a self-biased architecture that automatically tracks process, voltage, and temperature variations, producing stable and consistent DPLL performance over a broad range of operating conditions. Frequency agility is achieved with programmable dividers for the reference clock, feedback clock, and voltage-controlled-oscillator (VCO) output. A bypass multiplexer (MUX) allows either the reference clock or the PLL VCO output to be output from the PLL. Lock detect circuitry indicates the DPLL lock state. The experimental results of DPLL with waveforms are shown for a 0.2GHz DPLL. The blocks of the DPLL, synthesized from the hardware description language (HDL) are also provided. The static timing analysis of the DPLL is also done by using dc shell. The Built-in Self-Test (BIST) circuit is generated which tests the parameters include RMS jitter, loop gain, lock range, lock time and the natural frequency is calculated from the loop gain which are done by using Tessent Tool.

KEYWORDS:

DPLL, ESD, VCO, APLL, BIST, STA.

I. INTRODUCTION

The Analog-PLLs (APLLs) are broadly used up to this time, but Digital-PLLs (DPLLs) are attracting more concentration for the remarkable benefits of digital systems compared to their analog equivalents [1]. These benefits include supremacy in reliability, performance, reduction in size, speed & cost. DPLLs alleviated many problems associated with APLLs. A brief comparison is given in the following:

1. APLLs deteriorate from the sensitivity of the VCO to change in the power supply & temperature, hence the need for initial calibration and periodic adaptions. DPLLs do not deteriorate from such a problem [6].
2. The most well-known error detectors used in APLLs employ analog multipliers (balanced modulators) which are delicate to d.c. variations, a problem that does not survive in DPLLs.
3. DPLLs can perform at very low frequencies that cause issues in APLLs. These issues are associated to the operation of the analog LPF in withdrawing the lower frequency element, as it requires more time for superior frequency resolution, and this will minimize the locking speed.
4. Self-acquisition of APLLs is often time-consuming and untrustworthy, while DPLLs, a basic block diagram is shown in Figure 1, have superior locking speeds [6]. This is because of the fundamental operation of the analog LPF and the analog multiplier in the phase detector (PD).

The LPF cannot withdraw the lower frequency within less input cycles since the narrow time windowing will demolish the statistics in the frequency domain (due to the tradeoff between time & frequency resolution). Same judgments applies for the balanced modulator in the PD.

In opposite, a digital filter action is established on a difference equation with convergence marked by the coefficients of the equation, and the PD action. The examination of the positive-going zero-crossing sinusoidal DPLL was proposed in 1980 using fixed point theorems.

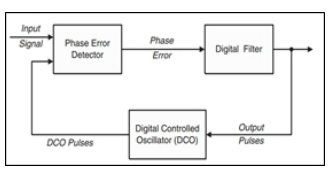

Fig. 1: Basic Block Diagram of DPLL

II. PROPOSED BLOCK DIAGRAM

The principle behind the implemented approach is completely depends upon the dividers used at input, output and feedback. This dividers are encoded to bit flag values which are completely in the form of powers of 2. Then the output frequency came out with multiplied version of input which depends on the equation followed by,

$$Op_{fr} = \frac{F_{ref} * Fb_{en}}{In_{en} * Op_{en}} \quad \ldots (1)$$

Where,
Op_{fr} = Output frequency,
In_{en} = Input divider encoded value,
Op_{en} =Output divider encoded value,
Fr_{ef}=Reference clock frequency.

The proposed Architecture is as shown in the Figure 2.
The key features of the DPLL including power, estimated area are shown below.

Key Features:

- Global Foundry 14LPP Process.
- Single 0.8V Power Supply.
- Low power – 1.6mW typical.
- 800 – 3200 MHz Native VCO Range.
- 80x160µm2 Area.
- Low deterministic jitter.
- Output MUX to allow bypass mode.
- Programmable reference, feedback and output dividers provide frequency agility.
- $F_{out} = F_{ref} * [(M/(N * R)]$(rewritten eq.1)

Where, M = 10 – 255.

R = 1, 2, 4, 8, 16, 32.
N = 1, 2, 4

- PLL lock detection function
- Power-down capability
- No external components required
- ESD protection and power supply bypass are fully integrated in PLL.
- Lock_Timemin~=(4xLOCK_DETECT_COUNT)*(REF_CLK period * Encoded REFDIV value).

The pin descriptions of the DPLL shown in Fig. 2 are shown in Table 1.

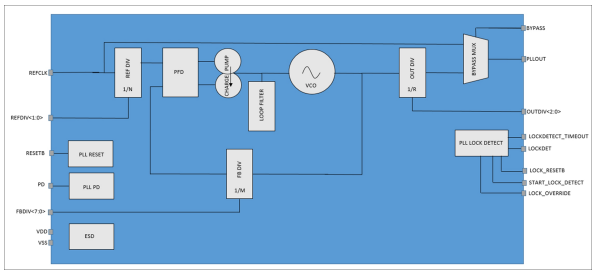

Fig. 2: The Proposed Block Diagram of DPLL

REFCLK	PLL Reference Clock. 25 – 200 MHz range
REF_DIV<1:0>	Divides reference clock frequency. Can be 1,2,4
RESETB	DPLL Reset
PD	PLL Power down
FBDIV<7:0>	PLL feedback divider. Divider range is from 10 to 255
VSS	PLL VSS supply
VDD	PLL VDD supply. 0.8V nominal
BYPASS	Enables bypass mode, which sends reference clock to PLL output
OUTDIV<2:0>	PLL output divider. Located outside of PLL loop. Can be 1, 2, 4, 8, 16, 32
PLLOUT	PLL Output. 25 MHz – 3200 MHz frequency range
LOCKDET	Indicates PLL lock state. 1=PLL LOCKED; 0=PLL UNLOCKED
LOCKDETECT_TIMEOUT	Indicates if lock detect state machine has timed out
LOCK_RESETB	Digital reset for PLL lock detect circuitry
START_LOCK_DETECT	Start operation of lock detect circuitry
LOCK_OVERRIDE	Overrides lock detect circuitry to force LOCKDET=1

Table 1. Pin Descriptions of Fig. 2

The corresponding waveforms for the generation of signal of frequency 200 MHz are shown in Figures 3-5. The input reference clock frequency can be found from the Figure 3 by observing the time period of the REFCLK which is 20,000ns i.e. the frequency is 50MHz.

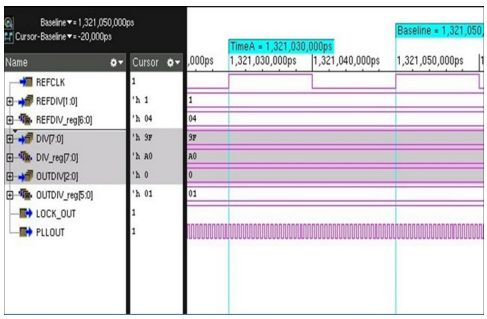

Fig. 3: Observation of input REFCLK frequency

The lock state of the DPLL can be observed from the Figure 4.

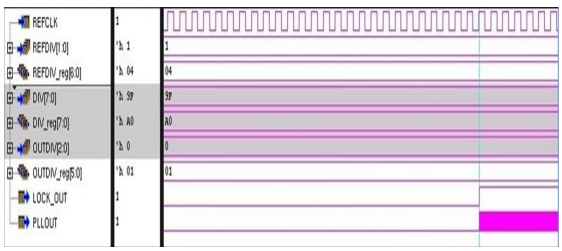

Fig. 4: Observation of lock state indicated by the signal, LOCK_OUT.

The output waveform is generated with the frequency of 200MHz as shown in the Figure 5.

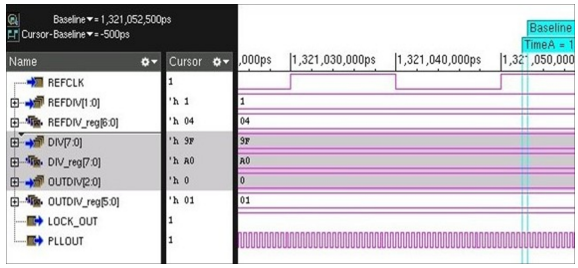

Fig. 5: Observation of the output signal PLLOUT frequency.

III. STATIC TIMING ANALYSIS (STA)

An STA is a process of checking the timing performance of a design by examining all achievable paths for setup & hold violations under worst-slack conditions. It checks each and every logic cell for the worst delay possibilities but does not bother the functionality of the circuit. In STA, the word static suggests to the reality that this timing examination is accomplished in an input unconventional way. It searches the worst-slack path of the design over all possible paths. In any complex design of a chip, logical paths are large in number [7]. There are three important steps to examine a design or we can say that to operate STA on any synchronous design. They are:

1. Divide the design into different timing paths.
2. Calculate the propagative delay of each and every logic cell along those timing paths.
3. Finally, checks for setup and hold violations w.r.t constraints applied on inputs and outputs in the design.

The dc shell (STA tool) examines from each and every path & over crosses it over the constraint which exists on that path.

Definitions:

For any circuit, timing analysis evaluates the quantity of time needed for inputs either clock pins or data pins to flow from top module inputs to top module outputs through many design cells and their interconnect. Genuine function of a flip-flop needs to be stable logical value at the data input pin for a particular time period before the capture edge of the design clock in the present clock cycle [7]. This amount of time is denoted by the setup time tsetup. Similarly, the logical value of the data input pin also require to be stable for a particular time period after the capture edge of the design for the successive clock cycle.

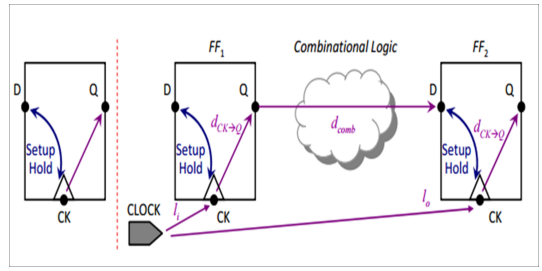

Fig. 6. Basic D flip-flop and its delay model (left), and two Flip-Flops in series and their delay models (right).

This amount of time is denoted by the hold time thold. The flip-flop delay models are portrayed in Figure 6 (left).

Timing Paths:

There are two main timing paths in any digital circuit. They are:

1. Data Path
2. Clock Path

Each and every timing path has a 'Start-Point' & as well as an 'End-Point'. Depending upon the type of timing-path, the start and endpoints may vary.

For a Data path,
Start-point: Input port of the design (since the data of the input can be launched with some delay from other circuitry) or any clock input pin of the flip-flop, latch or memory.

Endpoint: Data input pin of the flip-flop, latch or memory or an output port of the design (since the data of the output can be captured by some other circuitry).

For a Clock Path,
Start-point: Clock input pin
End Point: Clock-pin of the flip-flop, latch or memory (sequential element).

Then the paths can be denoted as IR, RR, RO, or IO.
Where IR-Input to Register Path, RR-Register to Register Path, RO-Register to Output Path, IO-Input to Output Path. All these types of Paths are shown in the Figure 7.

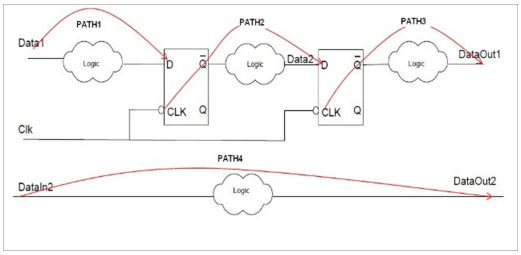

Fig. 7: Different types of Timing Paths

PATH1 Start-point: An Input port and End-point: The data input of a sequential element (Ex: Flip-flop). (IR path).

PATH2 Start-point: The clock pin of a sequential element (Ex: Flip-flop) & end-point: The data input of a sequential element (Ex: Flip-flop). (RR path)

PATH3 Start point: The clock pin of a sequential element and end-point: An output port. (RO path)

PATH4 Start-point: An Input port & end-point: An output port. (IO path)

STA results:

The input signal, REFCLK has the frequency range of 25MHz - 200MHz. If we increase the frequency beyond 200MHz, it causes setup and hold violations. The constraints are required to analyze the setup and violations. Initially clock (REFCLK) is created whose frequency is taken as 200MHz (Maximum frequency the design can withhold). This can be done by a command C1 (· 5ns 200MHz).

C1:
create_clock -period 5 REFCLK
Then the setup and hold time violators in the design can be observed by using a command C2. No violations occur, shown in Figure 8.
C2:
report_constraint -all_violators

```
****************************************
Report : constraint
         -all_violators
Design : IN14LPPPLL0315_ddr
Version: E-2010.12-SP3
Date   : Wed Jun  7 09:29:52 2017
****************************************

This design has no violated constraints.

1
```

Fig. 8: Observation of no violators when REFCLK has frequency of 200MHz.

Now, let we change the frequency to 500MHz. Then setup and hold violations will occur, that's why the input frequency range is fixed to 25MHz-200MHz. This can be done by a command C3 (2ns 500MHz).
C3: create_clock -period 2 REFCLK

Now, the hold and setup violations can be observed by the command C2. The violations are shown in the Figure 9-10.

```
min_delay/hold ('REFCLK' group)

                          Required        Actual
Endpoint                  Path Delay      Path Delay      Slack
-------------------------------------------------------------------
LOCK_OUT                    0.30            0.17 f        -0.13  (VIOLATED)
LOCKDETECT_TIMEOUT          0.30            0.18 f        -0.12  (VIOLATED)
```

Fig. 9: Hold violations when REFCLK has frequency of 500MHz.

```
max_delay/setup ('refclk' group)

                                    Required        Actual
Endpoint                            Path Delay      Path Delay      Slack
-------------------------------------------------------------------------
LOCK_DETECT/timeout_cnt_reg[9]/D
                                      0.84            4.16 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[0]/D
                                      0.84            4.17 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[3]/D
                                      0.84            4.17 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[4]/D
                                      0.84            4.17 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[5]/D
                                      0.84            4.17 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[10]/D
                                      0.89            4.22 f        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[12]/D
                                      0.84            4.16 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[16]/D
                                      0.84            4.16 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[2]/D
                                      0.84            4.16 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[6]/D
                                      0.84            4.16 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[8]/D
                                      0.84            4.16 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[13]/D
                                      0.84            4.16 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[1]/D
                                      0.84            4.15 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[7]/D
                                      0.84            4.15 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[11]/D
                                      0.84            4.15 r        -3.32  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[14]/D
                                      0.84            4.15 r        -3.32  (VIOLATED)
LOCK_DETECT/fbcount_req_refclk_reg/D
                                      0.88            4.18 f        -3.30  (VIOLATED)
LOCK_DETECT/timeout_cnt_reg[15]/D
                                      0.84            4.13 r        -3.29  (VIOLATED)
LOCK_DETECT/lockdetect_timeout_reg/D
                                      0.84            4.09 r        -3.25  (VIOLATED)
LOCK_DETECT/state_reg[0]/D
                                      0.83            4.05 r        -3.23  (VIOLATED)
LOCK_DETECT/state_reg[1]/D
                                      0.84            4.04 r        -3.20  (VIOLATED)
LOCK_DETECT/state_reg[2]/D
                                      0.82            3.87 r        -3.05  (VIOLATED)
LOCK_DETECT/state_reg[3]/D
                                      0.97            3.93 f        -2.96  (VIOLATED)
LOCK_DETECT/u_sync_detect_lock_unlock/sync_in_meta_reg[0]/D
                                      0.78            3.00 r        -2.22  (VIOLATED)
LOCK_DETECT/u_sync_lock_override/sync_in_meta_reg[0]/D
                                      0.78            3.00 r        -2.22  (VIOLATED)
LOCK_DETECT/u_sync_start_lock_detect/sync_in_meta_reg[0]/D
                                      0.78            3.00 r        -2.22  (VIOLATED)
```

Fig. 10: Setup violations when REFCLK has frequency of 500MHz.

IV. BUILT-IN SELF-TEST FOR DPLL

The controlling parameter at the input REFCLK is phase and the controlling parameter at the output PLLOUT is frequency. The main block for this BIST is TAP Controller and the concept behind this is Boundary Scan. The concept, Boundary Scan can be simply demonstrated by using the Figure 11. The scan path which is serially connected between the pins & device core is known as the Boundary-Scan Register (BSR). This register have plenty of cells known as BSCs (Boundary-Scan Cells). These BSCs are not observable in standard operation.

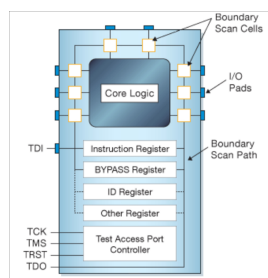

Fig. 11: Schematic Diagram of the Device having enabled JTAG

Interface Signals:
The main block for any JTAG interfacing system is Test-Access Port (TAP). It uses the following signals:

1. TCK – The signal called as Test-Clock used for state-machine operations.
2. TMS – The signal called as Test-Mode Select, changes at the rising edge of TCK. This signal is used to find the next state.
3. TDI – The signal called as Test Data-input which portrays the data shifted into the device's logic either for testing or programming. It is also changes at the positive edge of TCK.
4. TDO – The signal called as Test-Data output which portrays the data shifted out of the device's logic either for testing or programming and operates at the negative edge of TCK.
5. TRST – The signal called as Test-Reset which can be used to reset the state-machine of the TAP. This pin is optional.

TAP Controller:
It consists of a state-machine of 16-states. The movement in the states is commanded by TMS signal. The state alone decides the position of the JTAG device. Data can be write or read whether from DR or IR. The TAP state-machine is shown in the Figure 12 given below.

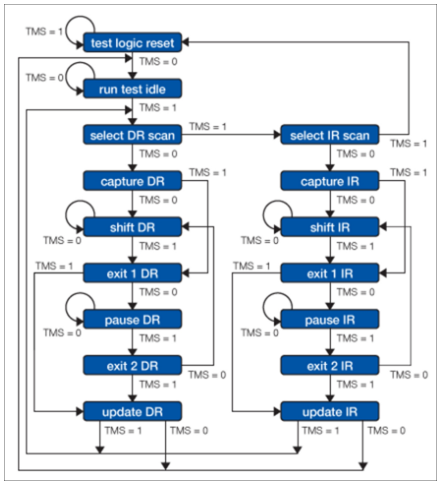

Fig. 12: TAP Controller State-machine

Boundary Instructions:

There are many instructions which are defined by the standard called as IEEE 1149.1. Some of those, explained below:

BYPASS – As name says, it bypasses the information between the lines TDI & TDO through the register called as BYPASS register. This instruction is used to test the devices of the JTAG chain to overcome unwanted overheads [8].

EXTEST – It is used to make the connection between the lines TDI, TDO & the register called as Boundary-Scan Register (BSR). The states of the TAP controller are captured by 'capture dr' and in the state 'shift dr', the information gets shifted into BSR and then gets passed into pins by using 'update dr' state.

SAMPLE/PRELOAD – It is also used to make the connections between the lines TDI, TDO & BSR. It is used for the purpose of preloading the information data into BSR.

Other instructions include:
IDCODE – It is also used to make the connections between the lines TDI, TDO & the register called as IDCODE Register. This instruction can be used to identify the device code.

INTEST – It is used to make the connections between the lines TDI, TDO & BSR. It is related to core-signals [8].

By using different instructions, the characteristics of DPLL are measured. The measured RMS Jitter for the DPLL for input frequencies 50MHz and 100MHz are shown in Figures 13 and 14 respectively.

The RMS jitter gets decreased when the input frequency is increased from 50MHz to 100MHz. It can be observed from the Figures 13 and 14.

```
================================================================================
LVDB Directory:      tcs_4_0_top.lvdb
LVDB Name:           tcs_4_0_top.lvdb
Test Config:    TEST
  TestStep: OutClk_Jitter  (PASSED)
  Execution Time Stamp: 05/25/16 18:01:59
    UltraController "BP0" (pad_PM_REF_CLK_I, 20.0100002288818ns)
      DLV_U_OutClk_Jitter_TestStepNo0: JitterFromCDF 30.34 ps RMS..
================================================================================
```

Fig. 13: RMS Jitter Measurement when input clock frequency = 50MHz.

```
================================================================================
LVDB Directory:      tcs_4_0_top.lvdb
LVDB Name:           tcs_4_0_top.lvdb
Test Config:    TEST
  TestStep: OutClk_Jitter  (PASSED)
  Execution Time Stamp: 05/25/16 17:06:12
    UltraController "BP0" (pad_PM_REF_CLK_I, 10.0100002288818ns)
      DLV_U_OutClk_Jitter_TestStepNo0: JitterFromCDF 8.23 ps RMS..
================================================================================
```

FIG 14: RMS Jitter Measurement when input clock frequency = 100MHz.

V. RESULTS AND DISCUSSION

The Frequency multiplier using DPLL is implemented for the input frequencies between 25MHz-200MHz. The simulation results for the output of frequency 200MHz are presented here. The Static Timing Analysis is done to check the setup and hold violations in the design of DPLL. The results of the same are also presented here. The Built-in self-test of the DPLL is done to measure its characteristics such as capture range, lock range, lock-time, natural frequency and RMS Jitter.

VI. CONCLUSION

The implemented DPLL can be used for the clock generation. It can also be used for DDR2/3. For small input frequency range, output signal of maximum frequency can be generated by using bit pattern encoding for the dividers used at input, output and at feed-back. In this paper, several techniques in full-digital are signified to test the DPLLs in the operating frequency range of 25MHz-200MHz.

REFERENCES

[1] L. Xia, J. Wu, Ch. Huang, M. Zhang, "Built-in self-test structure for fault detection of charge-pump phase-locked loop," IET Circuits, Devices & Systems, vol. 10, pp. 317-321, Jul. 2016.
[2] Z. Zong, M. Babaie and R. B. Staszewski, "A 60 GHz Frequency Generator Based on a 20 GHz Oscillator and an Implicit Multiplier," IEEE Journal of Solid-State Circuits., vol. 51, pp. 1261 – 1273, 2016.
[3] A. Elshazly, R. Inti, B. Young and P. K. Hanumolu, "Clock Multiplication Techniques Using Digital Multiplying Delay-Locked Loops, " IEEE Solid-State Circuits, vol. 48, pp. 1416-1428, May 2013.
[4] J. Y. Kim, Ch. Yao and A. N. Willson, "A Programmable 25-MHz to 6-GHz K/L Frequency Multiplier with Digital Krmvco Compensation," IEEE Transactions on Circuits and Systems, vol. 56, pp. 865 – 876, Mar. 2009.
[5] J. Ch. Hsu, Ch. Su, "BIST for Measuring Clock Jitter of Charge-Pump Phase-Locked Loops", IEEE Transactions on Instrumentation and Measurement, vol. 57, pp. 276-285, Jan. 2008.
[6] J. Vygen, "Slack in static timing analysis," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, pp. 1876 – 1885, Aug. 2006.
[7] .A. Taylor, B. Nelson, A. Chong, H. Lin, E. Chan, M. Soma, H. Haggag, J. Huard, J. Braatz, "Special issue on BIT CMOS built-in test architecture for high-speed jitter measurement," IEEE Transactions on Instrumentation and Measurement, vol. 54, pp. 975-987, May 2005.
[8] Ch. Lung Hsu, Y. Lai and Sh. Wang, "Built-in self-test for phase-locked loops," IEEE Transactions on Instrumentation and Measurement, vol. 54, pp. 996 – 1002, May 2005.
[9] M. J. Burbidge; A. Lechner; G. Bell; A. M. D. Richardson, "Motivations towards BIST and DfT for embedded charge-pump phase-locked loop frequency synthesizers," vol. 151, pp. 337-348, 2004.
[10] S. Kim and M. Soma, "An all-digital built-in self-test for high-speed phase-locked loops," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 48, pp. 141-150, Feb 2001.