AN ON CHIP ALL-DIGITAL CONFIGURABLE CLOCK GENERATOR FOR ASICs' AT-SPEED TESTING

BY

MOHAMMED AL-ASALI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER ENGINEERING

JANUARY 2013

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN- 31261, SAUDI ARABIA
DEANSHIP OF GRADUATE STUDIES

This thesis, written by MOHAMMED AL-ASALI direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING.**

Dr. Muhammad E. S. Elrabaa
(Advisor)

Dr. BASEM AL-MADANI

Department Chairman

Dr. Aiman Helmi El-Maleh
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Munir A. Kulaib AL-Absi
(Member)

13/2/13

Date

© Mohammed Al-Asali

2013

To my brothers

ACKNOWLEDGMENTS

I would like to acknowledge the guidance and support of my thesis advisor Dr. Muhammed Elrabaa. His advices, patience and the all set toward teaching me all concepts and materials that I need to finish this work is highly appreciated. The support and troubleshooting of Dr. Alaa El-Din Hussein and Mr. Amran Al-Aghbari is highly appreciated, too. Without their help and notes, this work wouldn't be finished.

TABLE OF CONTENTS

ACKNOWLEDGMENTS .... **V**

TABLE OF CONTENTS .... **VI**

LIST OF TABLES .... **IX**

LIST OF FIGURES .... **X**

LIST OF ABBREVIATIONS .... **XII**

ABSTRACT .... **XV**

1 CHAPTER 1 INTRODUCTION **1**

1.1 Testing and Characterization **1**

1.1.1 Virtual Prototyping **3**

1.1.2 Physical Prototyping **3**

1.2 Clock generators **5**

1.3 Thesis Organization **6**

2 CHAPTER 2 LITERATURE REVIEW **7**

2.1 Speed Characterization **7**

2.2 Ring Oscillators **10**

2.2.1 Shunt Capacitive Loads **15**

2.2.2 Current Starved Delay Stages **16**

3 CHAPTER 3 OVERVIEW OF THE TEST & CHARACTERIZATION PLATFORM 20

3.1 The TACP Support circuitry (TSC) **21**

3.1.1 The Configurable Clock Generator **24**

3.1.2 The Frequency Measuring Circuit (FMC) 24

3.1.3 The Clock Selection and Application Circuit (CSaAC) 25

3.1.4 The Port Selection Block 27

3.1.5 The Test Application/Result Ports (TAP/TRP) 29

3.2 The TACP 31

4 CHAPTER 4 THE DIGITALLY CONTROLLED RING OSCILLATOR 33

4.1 Design of the DCO 33

4.2 DCO Portability 38

5 CHAPTER 5 EXPERIMENTAL RESULTS 44

5.1 Implementation of the Configurable Clock Generator using TSMC 0.35U technology 44

5.1.1 Pre-Layout simulation of the chip 46

5.1.2 Post-Layout simulation of the chip 51

5.2 Implementation of the Complete Test Support Circuitry Using LFoundry 150nm Technology 60

6 CHAPTER 6 CONCLUSIONS AND FUTURE WORK 78

6.1 Conclusions 78

6.2 Future Work 78

REFERENCES 80

APPENDICES 85

APPENDIX "A" CCG Schematics 86

APPENDIX "B" CCG Simulations 91

APPENDIX "C" Support Circuitry Simulations using S820s Benchmark 99

APPENDIX "D" Synopsys Scripts 101

APPENDIX "E" Test Bench of the Support Circuitry 105

VITAE 112

LIST OF TABLES

Table 1 Summary of performance of Clock Generator of [41] 12
Table 2 CL estimation for various processes 37
Table 3 DCO capacitances for several switches sizes 40
Table 4 DCO periods for binary and non-binary switches 41
Table 5 Summary of the clock generator chip 54

LIST OF FIGURES

Figure 1 Classification of different ASICs' verification options 2
Figure 2 The generic data communications chip of [45] 8
Figure 3 BIST design flow of [47] 9
Figure 4 The DFT structure of [47] 9
Figure 5 At-speed LCCLK generator of [48]..10
Figure 6 Multi-capacitor approach of [1] 10
Figure 7 Delay time (measured) versus control word of [1]...11
Figure 8 Block diagram of the proposed DLL based clock generator of [41] 12
Figure 9 (a) Block diagram and (b) transfer curve of [42] 13
Figure 10 Ring oscillator with shunt capacitors of [43] 13
Figure 11 Scheme of the multiplier of [44] 14
Figure 12 N-stage ring oscillator model of [7]..15
Figure 13 Interconnection of identical ring oscillators...15
Figure 14 Delay line of [4]...17
Figure 15 Delay versus input vector of [4] 18
Figure 16 Cell controller that realizes the non-linearity test and correction of [13] 18
Figure 17 Schematic of [8] DCO 19
Figure 18 Overview of the proposed test and characterization method [49] 21
Figure 19 The fixed interface signals between the TACP and the prototyping chip 22
Figure 20 Block diagram of the TACP Support Circuitry (TSC) to be placed on the prototype chip..23
Figure 21 The configurable clock generator 24
Figure 22 The frequency measuring Circuit (FCM)...25
Figure 23 The state diagram of the control unit of the FMC 26
Figure 24 The Clock Frequency Control Register 27
Figure 25 The Clock Selection and Application Circuit 27
Figure 26 Logic Simulation Results for the CSaAC [49]..28
Figure 27 The Port Selection Circuitry..28
Figure 28 k-bits wide test application port (TAP)..30
Figure 29 l-bits wide test results port (TRP)...30
Figure 30 Scan test application/result ports 31
Figure 31 The utilized digitally controlled oscillator (DCO) 34
Figure 32 Period and the frequency of the DCO using TSMC 0.18U technology 38
Figure 33 DCO period for various technologies 39
Figure 34 The effect of using binary switches on DCO period 42
Figure 35 DCO frequency range from around 1.4GHZ to about 500MHZ, frequency range when dividing by two and by four 43
Figure 36 Flow diagram of the design in Tanner EDA 45
Figure 37 Top logic level of the CCG 46

Figure 38 CCG signals (maximum frequency); pre-layout simulation 49
Figure 39 CCG signals (minimum frequency); pre-layout simulation.............................51
Figure 40 Clock generator's core layout 52
Figure 41 Final tested chip of the clock generator 53
Figure 42 CCG signals (maximum frequency); post-layout simulation...........................56
Figure 43 CCG signals (minimum frequency); post-layout simulation 58
Figure 44 The fabricated chip; the core at the lower half of it..59
Figure 45 Design flow of the support circuitry 61
Figure 46 Layout of the support circuitry 62
Figure 47 (a) Schematic of the DCO using Synopsys tools; (b) Layout of the DCO 64
Figure 48 Final chip sent for fabrication 65
Figure 49 Result of dividing the DCO output by four..67
Figure 50 CCG spice simulation 72
Figure 51 Support circuitry spice simulation; 4-bit adder is used as an IUT 75
Figure 52 Two pulses of the selected clock produce when the AaC signal is high 76
Figure 53 Schematic of the DCO 86
Figure 54 Schematic of the frequency divider 86
Figure 55 Schematic of the control word register 87
Figure 56 Schematic of the FSM..88
Figure 57 Schematic of the high frequency counter...89
Figure 58 Schematic of the frequency register 90
Figure 59 Post-layout simulation of the CCG using DCO frequency divided by 2..........92
Figure 60 Post-layout simulation of the CCG using DCO frequency divided by 4..........94
Figure 61 Post-layout simulation of the CCG using DCO frequency divided by 8..........96
Figure 62 Post-layout simulation of the CCG using DCO frequency divided by 16........98
Figure 63 Support Circuitry simulation; test vector is 00011010001000110111111 and expected result is 000000000000000000110000 99
Figure 64 Support Circuitry simulation; test vector is 00001011001100011101001and expected result is 000000000000000000110000 99
Figure 65 Support Circuitry simulation; test vector is 11100101101010111011001 expected result is 000000000000001000100000 99
Figure 66 Support Circuitry simulation; test vector is 01001110100111000011100 expected result is 000110000000000111000000 100
Figure 67 Support Circuitry simulation; test vector is 11110010111011100100000 expected result is 100000000000000000110000 100
Figure 68 Pre-synthesis, post-synthesis and post-layout simulation of the support circuitry using 8 bit pipelined adder IUT; 00010101+11011010+1=11110000; the last same three signals are the pre-synthesis, post-synthesis and post-layout results 100

LIST OF ABBREVIATIONS

ASIC	:	Application-Specific Integrated Circuit
ATE	:	Automated Test Equipment
BIST	:	Built-In Self-Test
CCG	:	Configurable Clock Generator
CIN	:	Input Capacitance
CJ	:	Junction Capacitance
CLK	:	Clock
CSaAC	:	Clock Selection and Application Circuit
CW	:	Control Word
DCO	:	Digitally Controlled Oscillator
DFT	:	Design for Testability
DLL	:	Delay-Locked Loop
DNL	:	Differential Non-Linearity
DUT	:	Device under Test
EDA	:	Electronic Design Automation
FMC	:	Frequency Measuring Circuit

FR	:	Frequency Register
FSM	:	Finite State machine
HFCLK	:	High Frequency Clock
IP	:	Intellectual Property
IUT	:	IP under Test
LCCK	:	At-Speed Launch-Capture Clock
MOS	:	Metal Oxide Semiconductor
PLL	:	Phase-Locked Loop
RO	:	Ring Oscillator
SERDES	:	Serial-to-Parallel and Parallel-to-Serial data conversion
SoC	:	System on a Chip
STA	:	Static Timing Analysis
TACP	:	Test and Characterization Processor
TAP	:	Test Application Port
TDC	:	Time to digital converter
TPI	:	Test Point Insertion
TRPs	:	Test Result Ports

TSC	:	TACP Support Circuitry
TSMC	:	Taiwan Semiconductor Manufacturing Company

ABSTRACT

Full Name : [Mohammed Abdulqaher Ahmed Al-Asali]

Thesis Title : [An On-Chip All-Digital Configurable Clock generator for ASICs' At-Speed Testing]

Major Field : [Computer Engineering]

Date of Degree : [January 2013]

Recently a low-cost method for speed characterization of ASICs has been reported [49]. This method requires a portable on-chip configurable clock generator to change and measure the frequency of the applied clock. The purpose of this work is to design, implement and evaluate the performance of an all-digital, portable, digitally-controlled, configurable clock generator using industry-standard advanced Electronic Design Automation (EDA) tools such as those from Synopsys, Mentor Graphics and Tanner EDA. Also, as a proof of concept, an ASIC chip complete with several circuits under test and test support circuitry including the portable clock generator is designed. An intensive analysis of single ended digitally-controlled oscillators (DCOs) which represent the main building block of such configurable clock generators is provided. A thorough review of existing DCOs has been conducted to select the most suitable for this work. Based on this, a DCO with capacitive shunt load has been selected. Theoretical analysis of the selected DCO has been carried out and conditions for proper operation under different process conditions with minimum sized-inverters and maximum possible frequency range have been obtained. The DCO has been designed for an on-chip configurable clock generator for at-speed testing of ASICs with maximum possible frequency range, reasonable linearity and resolution. It was designed based on the developed theoretical

analysis and verified by pre-layout and post-layout simulations in many process technologies. The DCO was sent for fabrication together with the configurable clock generator using TSMC 0.35U technology. Also, to verify the operation of the configurable clock generator, a complete ASIC with four IPs under test (IUTs) has been designed using LFoundry’s 150 nm technology. In addition to the four IUTs, the ASIC contains test support circuitry that would be serially connected to an external specially developed test and characterization processor (TACP) to receive/send data from.

ملخص الرسالة

الاسم الكامل:محمد عبدالقاهر احمد العسلي

عنوان الرسالة: مولد ترددات رقمي قابل للتكوين و للوضع على الرقاقات لاختبار الدوائر المتكاملة محددة التطبيقات بسرعات مختلفة

التخصص: هندسة حاسب الي

تاريخ الدرجة العلمية: يناير ٢٠١٣

تم تطويرطريقة منخفضة التكلفة مؤخرا لتوصيف سرعة الدوائر المتكاملة محددة التطبيقات [49]. وهذه الطريقة بحاجة لمولد تردد قابل للدمج لكي يتسنى تغير التردد المطبق عليها وقياسه. فالغرض من هذا العمل هو تنفيذ وتقييم أداء مولدات التردد الرقميه القابله للدمج مع اي تقنية باستخدام أدوات تصميم متقدمة ك Synopsys و Tanner .كدليل على المفهوم تم تصميم شريحة كاملة بعدة دوائر تحت الاختباروتم ايضا تصميم دائرة مساعدة تتضمن مولد الترددات القابل للوضع. هذا العمل قدم تحليلا مكثفا للمذبذ الحلقي الرقمي (DCO) والذي يمثل جزء رئيسي لمثل هذه المولدات. ويتحقق تقييم (DCO) من خلال استكشاف باقي المذبذبات الرقمية الموجودة حاليا ومن ثم تم تحديد أنسبها لهذا المشروع. بناءا على ذلك تم اختيار مذبذب حلقي يعتمد على المكثفات لتغيير تردده والتحكم به. وبالإضافة إلى ذلك، تم تصميم مذبذب يتم السيطرة عليه رقميا لمولد الترددات على رقاقة لاختبار سرعة الدوائر المتكاملة محددة التطبيقات. ويجب أن يكون حجم الخلية ادنى ما يمكن والحد الأقصى للتردد اعلى ما يمكن مع زيادة خطية معقولة. تم تصميمه بناء على تحليلنا النظري والتحقق منه في العديد من التقنيات. تم إرسال DCO للتصنيع جنبا إلى جنب مع مولد التردد باستخدام التقنية TSMC 0.35U.

لكي يتم التحقق من عمل مولد الترددات تم تصميم الدائرة الداعمة على رقاقة باربع دوائر تحت الاختبار باستخدام تقنية LFoundry 150nm وأدوات Synopsys. بالإضافة الى ذلك هذه الدائرة ستكون موصلة بمعالج خارج الشريحة (TACP) وضع خصيصا لإختبار الشريحة وتلقي وإرسال البيانات من الدائرة الداعمة التي تحتوي على الدوائر تحت الاختبار.

CHAPTER 1

INTRODUCTION

1.1 Testing and Characterization

Many university researchers and chip designers in small companies are faced with a large problem when it comes to developing new electronic circuits or products; the cost of testing. In order to verify their product/project outcome (i.e. the developed electronic chip) they need to fabricate a prototype, test it and characterize its performance. With the current speeds of few Giga Hertz, these circuits would require very expensive testers and scopes. The high cost of such testing equipment (Millions of US$) is definitely prohibitive for most universities. At the same time, trends in electronic design have converged in the last few years to what is known as an IP-Based design. This is a design methodology based on re-using existing circuit blocks, namely the IP (intellectual property) blocks. These blocks are designed and verified (through prototyping and testing) by IP vendors and are then used and re-used by ASIC (application-specific integrated circuits) designers. This is a result of two factors; very short time-to-market windows fueled by fierce competition and ever increasing consumer expectations, and the high cost of design (designers' salaries). In fact, nowadays, most integrated circuits designed for electronic consumer products are being assembled from pre-designed, silicon-proven IP blocks.

Chip IPs are electronic circuits that are developed and licensed either as soft IPs (i.e. modeled using synthesizable Hardware-description languages such as VHDL and Verilog) or as hard IPs (layout macros). In both cases, the IP vendors have to show Silicon proof of their IP's performance (i.e. performance figures based on prototyping). Developing a cost-effective solution that would enable circuit designers to prototype, test and characterize their IPs at the operational speeds would be highly desirable.

As shown in Figure 1 below, there are currently two main methods of verifying ASICs' functionality and performance [19-30]; the conventional physical prototyping, and virtual prototyping. Physical prototyping is further divided into two options; 1) Prototyping using FPGAs (Field Programmable Gate arrays) [22-30], and 2) Prototyping via fabrication with a silicon foundry [21].

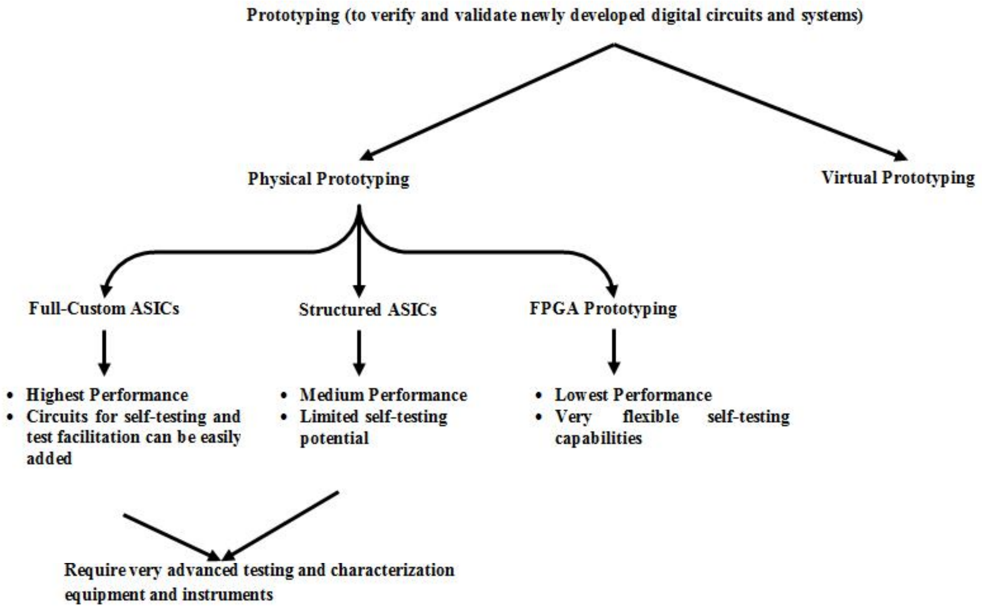

Prototyping (to verify and validate newly developed digital circuits and systems)
Physical Prototyping
Virtual Prototyping
Full-Custom ASICs
Structured ASICs
FPGA Prototyping
• Highest Performance
• Circuits for self-testing and test facilitation can be easily added
• Medium Performance
• Limited self-testing potential
• Lowest Performance
• Very flexible self-testing capabilities
Require very advanced testing and characterization equipment and instruments

Figure 1 Classification of different ASICs' verification options

1.1.1 Virtual Prototyping

Virtual prototyping is basically simulation-based verification. Silicon virtual prototyping tools accept RTL code, hundreds of macros (internal and third-party IP) and gate-level inputs to enable correct SOC construction throughout the design flow [19,20]. The tools attempt to capture the effects of all physical parameters (process and otherwise) through modeling. The prototype built is full-chip and relies on final detailed placement and very realistic routing. Vendors of these tools claim that the prototype correlates closely to the tape-out version of the chip. The problems with these tools are twofold; 1) They are very expensive software tools that are beyond the capabilities of most universities and small IP vendors, 2) The produced prototypes are only characterized via simulations. No matter how accurate the vendors claim their simulators are, they will never replace physical prototyping.

1.1.2 Physical Prototyping

Full-Custom fabrication is the most trusted and accepted method of prototyping, since it reveals the actual performance of the circuit being prototyped. Fabricated full-custom chips would achieve the highest performance but they would require very expensive testing equipment to test and characterize their performance at their operational speeds (called at-speed testing). Some on-chip characterization techniques have been proposed to overcome this problem [32-34] but they are either for specific class of circuits (e.g. memories) or can’t perform high speed testing and do not include the flexibility of the standalone testers.

Some low-cost testing platforms were proposed lately [36-37]. These platforms either can’t operate at high speed [36, 37] and/or would require very elaborate and unreliable solutions based on discrete components [36]. Also, most of these techniques are designed to assist automatic test equipment [37, 38], making them a high cost solution. In a recent patent [38], a method for characterizing integrated circuits is devised. Voltage and clock controllers are integrated on the DUT chip to characterize speed versus voltage. Test vectors could come from outside or from an on-chip BIST (Built-In-Self-Test) circuit. For high speed characterization, the test controller would be added on-chip in addition to the BIST circuit resulting in large area overhead. Also, the method does not provide a general way for applying stimuli and capturing results, thus requiring custom BIST circuit for each DUT.

As the above discussion shows, there is a need for a low-cost method for testing and characterizing digital integrated circuits containing prototypes of (possibly several) independent circuit IPs. Hence, the method should:

1. Allow functional (correctness) testing and speed characterization (high clock frequencies) of any number of circuits on the same chip. This requires a way to allow the user to specify his/her functional test procedure and data, capture the results, and specify the clock frequency.
2. It should also support any number of input/output ports per IP circuit under test (referred to as IUT hereafter) with arbitrary port widths (width refers to the number of bits per port).

1.2 Clock generators

Clock generators are widely used in ASICs and system-on-chip (SoCs) to generate on-chip clocks with or without an external input reference clock. Configurable clock generators are special clock generators that can produce clocks with tunable frequency [39].

Ring oscillator is an essential part of configurable clock generators. Ring oscillators, mainly comprised of ring-connected chain of delay elements (inverters) are popular due to their low cost and small size. The number of delay stages must be odd when using single ended inverters in order to produce the oscillation [10]. Ring oscillators can be single ended or differential and each of them has advantages and disadvantages in terms of jitter and power consumption [11]. There are many methods to digitally control the frequency of a ring oscillator such as shunt capacitance, current starving, and coupled ring oscillators. These methods will be investigated and simulated to identify the best method for at speed testing [12]. In addition to controlling the ring oscillator, clock generators can be controlled using digital circuits such as counters to divide the output frequency of the ring oscillator making the overall output signal controllable.

Speed characterization for IPs requires a special clock generator [40] that can easily be integrated with any IP using any fabrication process. This means it has to be all digital. Also, it should occupy small area compared to a phased locked loop (PLL) or a delay locked loop (DLL). PLL or DLL-based clock generators are a perfect choice as internal clocks that can reduce clock skew for instance, but are not a good choice for IPs at-speed testing as they contain analog components for filtering and pulse generation which make them difficult to integrate with ASICs and port from one process to another [4].

1.3 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 is a literature survey that includes the basics of clock generators, ring oscillators and various types of digitally controlled ring oscillators, at-speed testing concepts and methods, and recent advances in digitally controlled ring oscillators. An overview of the testing and characterization platform is given in details in Chapter 3. In Chapter 4, details of the proposed DCO are provided including the analysis and simulations of the DCO using many processes. In Chapter 5, the details of the implementation of the clock generator along with the DCO using TSMC 0.35U technology is provided. Also in that chapter the detailed implementation of the test support circuitry along with the configurable clock generator using LFoundry's 150 nm technology is provided. Finally, conclusions and future work are described in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

2.1 Speed Characterization

A new BIST methodology, Figure 2, suitable for functional testing of transceivers was reported in [45]. Practical circuits were presented which allow the at-speed testing of various functional blocks. The Controller of this method may use a low frequency clock in the beginning, and speed-up the clock until the chip fails to determine the maximum frequency of the device under test.

A design methodology for at-speed BIST, using a multiple-clock domain scheme was presented in [47]. Some experimental test results of large industrial designs, was also shown. Figure 3 shows the BIST design flow and Figure 4 illustrates the DFT structure of this methodology.

The above two methods require custom circuit design for each new IUT. They also consume large chip area.

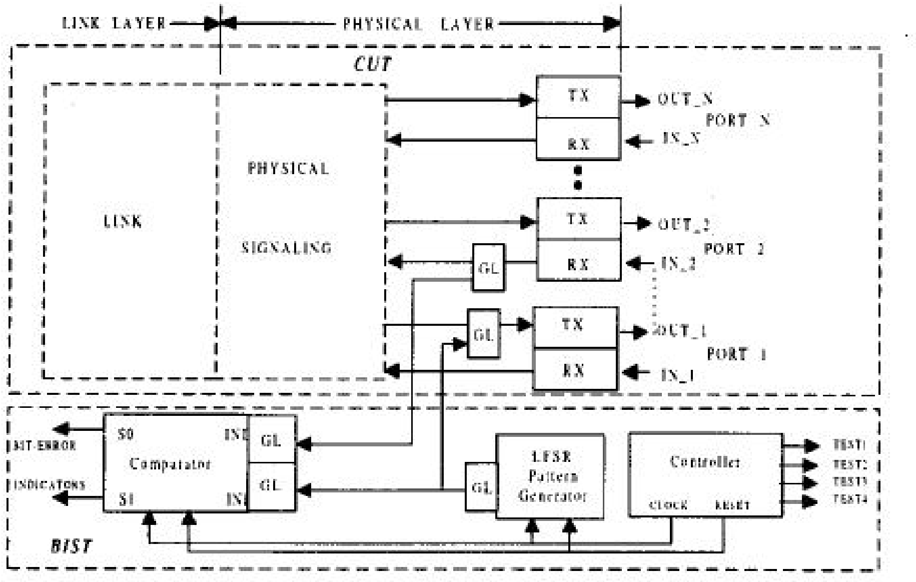


Figure 2 The generic data communications chip of [45]

A reduced pin count testing technique, Figure 5, was presented in [48]. This technique proposes that low-cost automated test equipment (ATE) be effectively utilized to reduce testing costs by using an on-chip test clock generator. Experiments of this technique show its effectiveness in utilizing the ATE channels and scan delay testing.

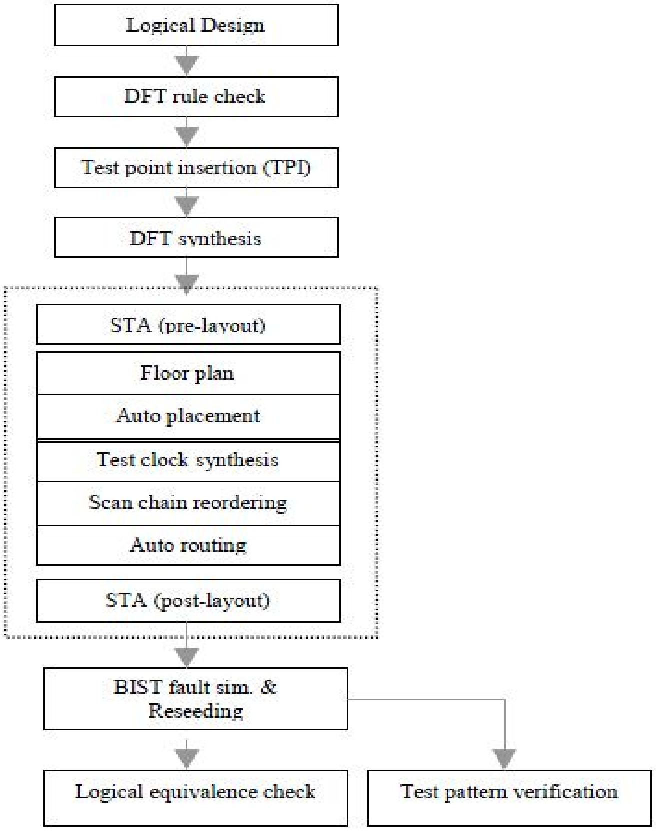


Figure 3 BIST design flow of [47]

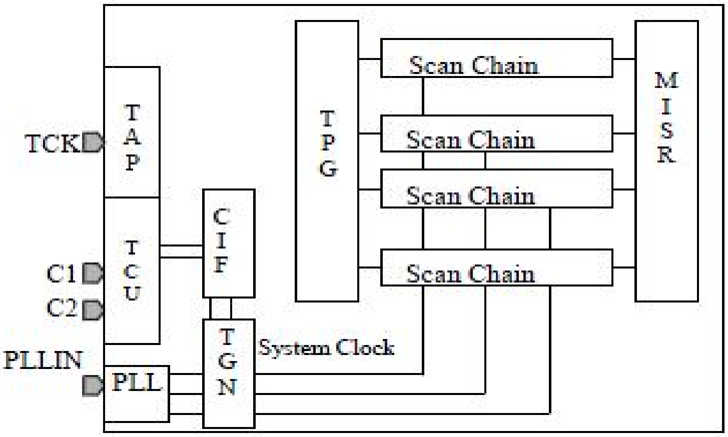


Figure 4 The DFT structure of [47]

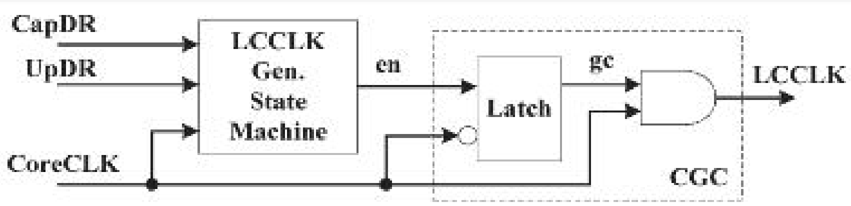


Figure 5 At-speed LCCLK generator of [48]

2.2 Ring Oscillators

Ring oscillators are constructed from delay lines connected in a ring-fashion (last output connected to first input). A new type of CMOS delay line as in Figure 6 was reported in [1]. The delay element is an array of capacitors controlled by a digital signal vector. The delay line produces tunable 16 × 0.5 ns delay under process variations as shown in Figure 7. This method, as the authors demonstrated, is able to produce a stable period steps but more capacitors with different sizes are needed which is not advantageous when designing for portability.

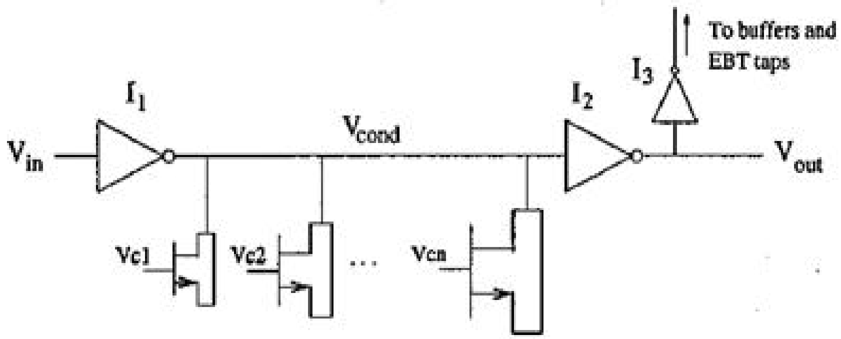


Figure 6 Multi-capacitor approach of [1]

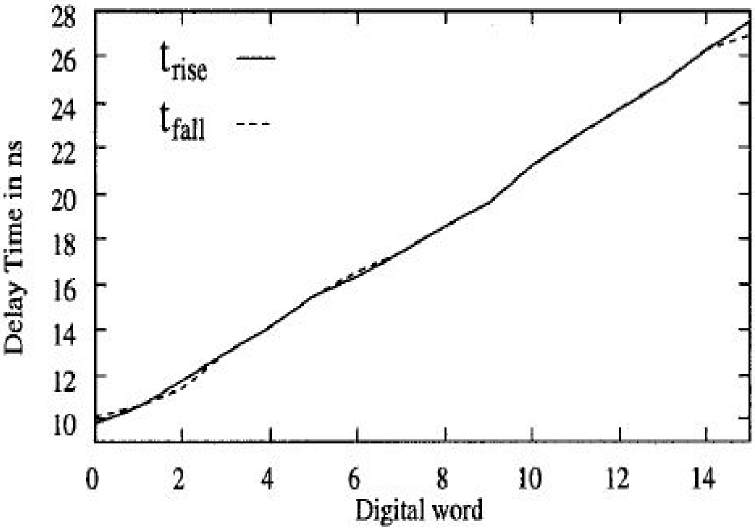


Figure 7 Delay time (measured) versus control word of [1]

A new method, Figure 8, to obtain a high frequency clock (1 GHz) in 0.25μm CMOS technology from a low frequency reference clock (10 MHz) was presented in [41]. Using a 2.5V power supply, the clock multiplication power is 0.822mW. The performance of the multiplication unit is tested on PSPICE and the results are summarized in Table 1. However, the operating frequency range is low and the eriod resolution was not mentioned.

An all-digital PLL was reported in [42] with a sub-exponent time-to-digital converter (TDC) which can scale its resolution according to the time difference, Figure 9. The TDC which was implemented in a 0.18 μm CMOS shows the minimum period of 1.25 ps and power consumption of 1.8 mW at 60 MHz.

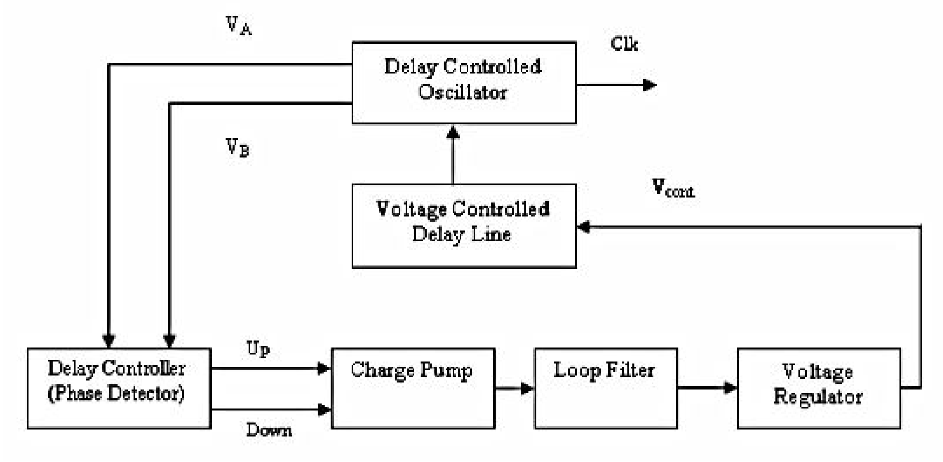
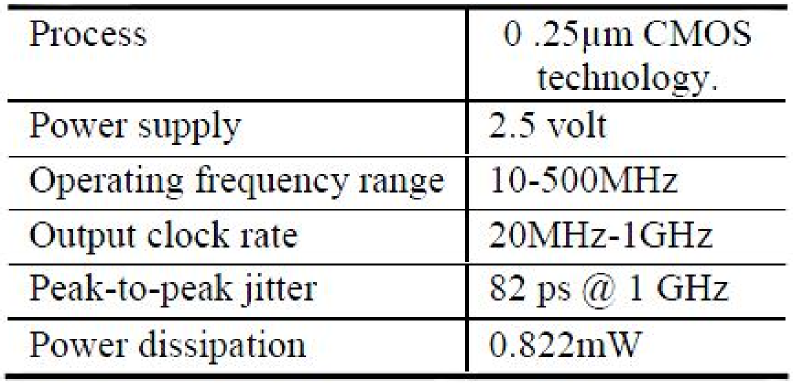

V_A
Delay Controlled Oscillator
Clk
V_B
Voltage Controlled Delay Line
V_{cont}
Delay Controller (Phase Detector)
Up
Down
Charge Pump
Loop Filter
Voltage Regulator

Figure 8 Block diagram of the proposed DLL based clock generator of [41]

Table 1 Summary of performance of Clock Generator of [41]



Process	0 .25µm CMOS technology.
Power supply	2.5 volt
Operating frequency range	10-500MHz
Output clock rate	20MHz-1GHz
Peak-to-peak jitter	82 ps @ 1 GHz
Power dissipation	0.822mW

A high frequency and low power digitally controlled clock generator was reported in [43] and fabricated using a 0.35 µm process, Figure 10. Using shunt-capacitive loads, the clock generator demonstrated a frequency range of up to 1.15 GHz at a 3.3 V supply voltage.

In [44] an all-digital DLL is reported. The circuit was fabricated to test the proposed method of clock frequency multiplication shown in Figure 11. It is mainly intended for ASICs and is generated by a parameterized generator which relies on a standard cell library, thus eliminating the need for implementing the multiplier as a full custom macro cell. 170 MHz clock signal was obtained from 8.5 MHz external clock using 1 µm CMOS process and a 5 V supply voltage.

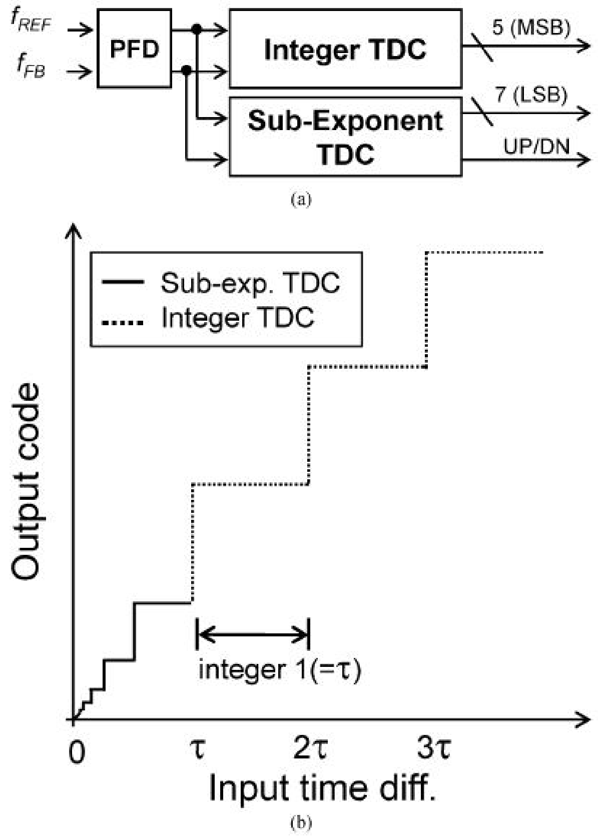


Figure 9 (a) Block diagram and (b) transfer curve of [42]

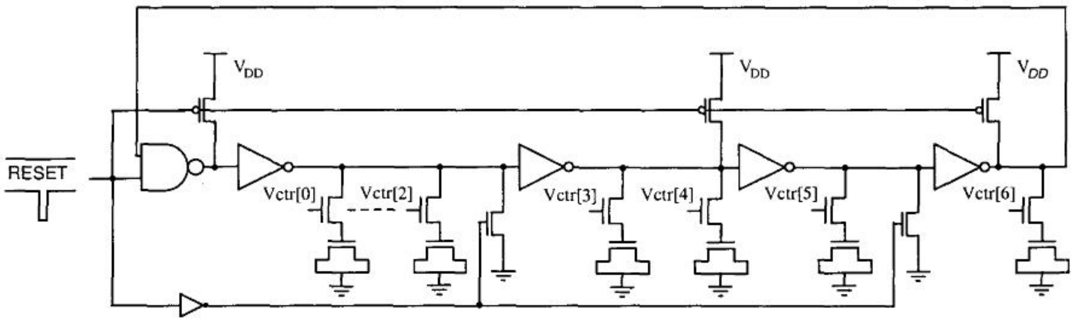


Figure 10 Ring oscillator with shunt capacitors of [43]

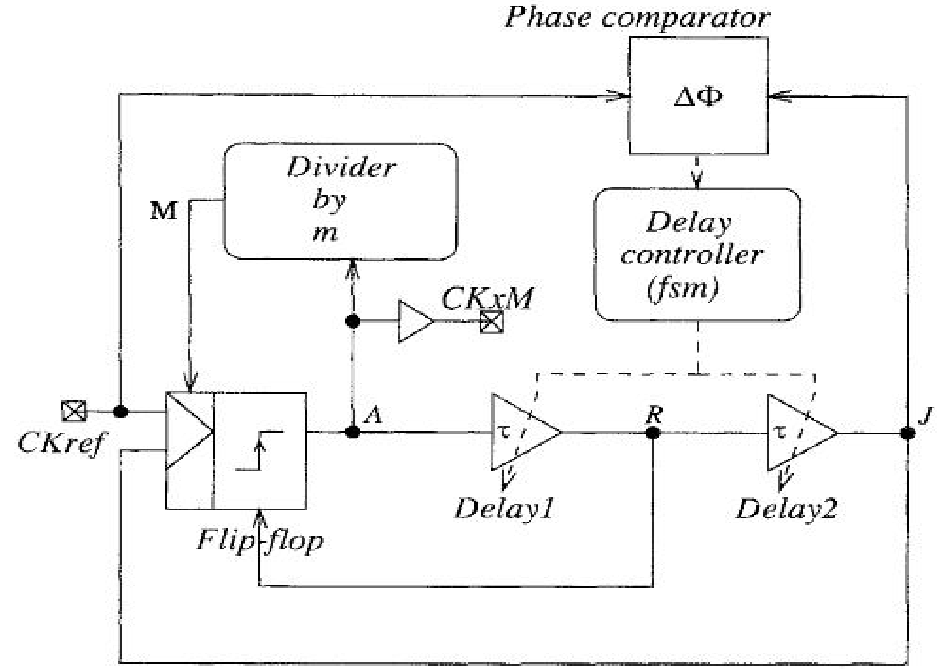


Figure 11 Scheme of the multiplier of [44]

Nonlinear analysis for an n-stage ring oscillator, such as the one in Figure 12, was carried out in [7]. Also a synchronization scheme for the interconnection of identical ring oscillators as in Figure 13 was proposed. By representing the n-stage ring oscillator as a cyclic system, a sufficient condition for global stability was derived. The synchronization of N identical ring oscillators connected through linear coupling was analyzed and the conditions for synchronization were obtained.

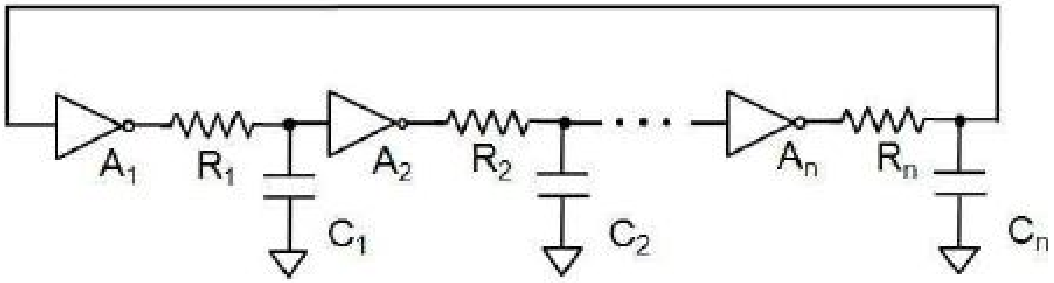


Figure 12 N-stage ring oscillator model of [7]

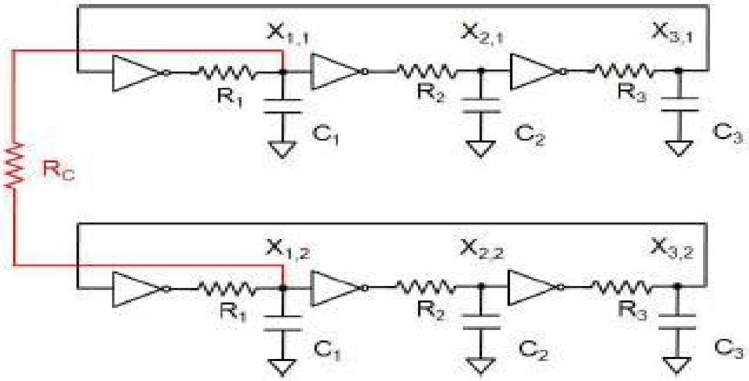


Figure 13 Interconnection of identical ring oscillators

2.2.1 Shunt Capacitive Loads

NMOS transistors were used as shunt capacitor in [2]. The authors mentioned that they used six of them and distributed them along a 5-stage single ended ring oscillator. They also utilized a fast reset technique that is based on applying certain initial voltage at each input of the inverters which yielded a predictable behavior of the ring oscillator. The main problem with that scheme is that it cannot be used for At-Speed-Testing since the frequency obtained using this method is relatively low (maximum of about 1GHZ).

Although the paper is not clear about the resolution the design can produce, the difference between the measured and simulated behavior shows that there is some problems in identifying the effect of the shunt capacitors on the frequency of the oscillator. Same approach of [2] was also used in [3].

An on-chip all-digital hardware calibration technique to reduce nonlinearity in delay-locked delay lines has been addressed in [5]. A shunt-capacitor circuit scheme was used and an iterative calibration algorithm was developed. A circuit efficiently realizing the proposed algorithm was designed and fabricated using a 0.6 µm CMOS technology. The technique was applied to a 32-stage DLL and each cell can be calibrated with a correction resolution of about 2% of the nominal cell delay. The silicon area occupied by the calibration circuitry was substantially the same as that occupied by the delay line itself which means no area overhead was produced using this method, and the test results demonstrated the effectiveness of the calibration technique, showing maximum final nonlinearity values close to 1%.

It is worth noting that the proposed method is of general use because it can be adapted to any process and can be applied to all applications that use delay lines to obtain high-resolution divisions. In addition, the size and number of calibration shunt capacitors can be changed in the design to achieve the target final nonlinearities.

2.2.2 Current Starved Delay Stages

Maymandi and Sachdev [4] propose a new digital delay line shown in Figure 14. This method achieves a very reasonable step and a relatively high frequency as in Figure 15. The method mainly utilizes the idea of current starved ring oscillator and not shunt capacitors. The problem of reducing the delay non-linearity of the cells of a DLL is also addressed in [13]. The cell non-linearity is mainly due to the process and geometrical parameter mismatch between the cells and can be reduced by applying a calibration of the cell delay. The paper shows how providing each cell with a controller that performs a calibration of the cell itself can reduce the delay non-linearity. The delay-line is exposed to a statistical test to estimate the values of the single cell delay and an individual delay correction to each cell is applied. The structure of the cell as in Figure 16, designed according to an all-digital shunt-capacitor circuit scheme, and that of the cell controller, responsible for the cell delay estimation and correction, are described. A 32-stage DLL is examined as case study in the paper.

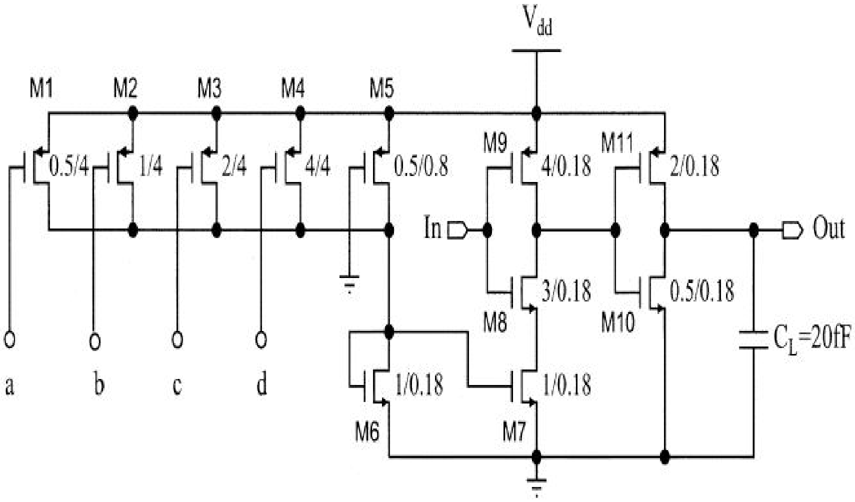


Figure 14 Delay line of [4]

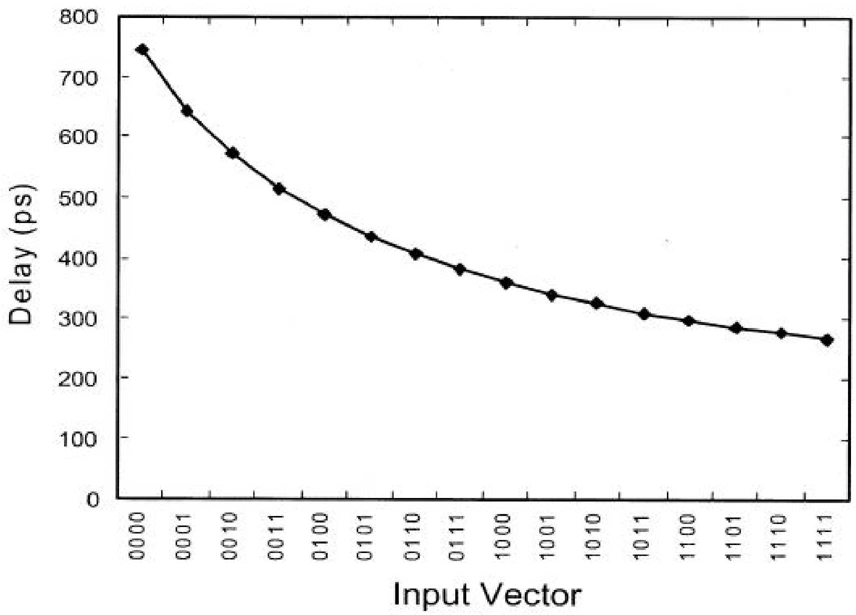


Figure 15 Delay versus input vector of [4]

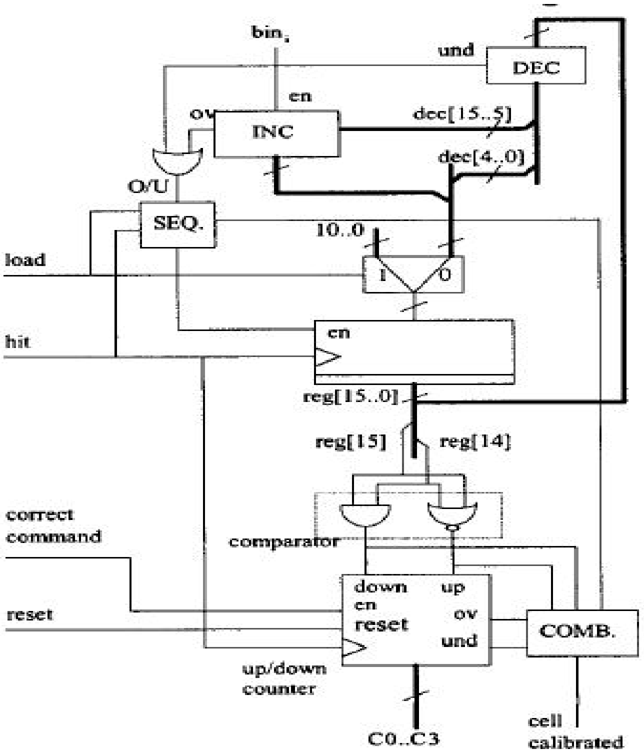


Figure 16 Cell controller that realizes the non-linearity test and correction of [13]

In [8], an 8-bit digitally-controlled oscillator, Figure 17, was designed. It is based on a ring topology using TSMC 0.35 um CMOS process parameters. The authors mentioned that one of the advantages of the ring topology oscillator is that it does not contain any inductor so that their chip size is very small. Also, it consumes 19 mA current at 3.3 V supply voltage and simulated phase noise is 106dBc/Hz at 1MHz.

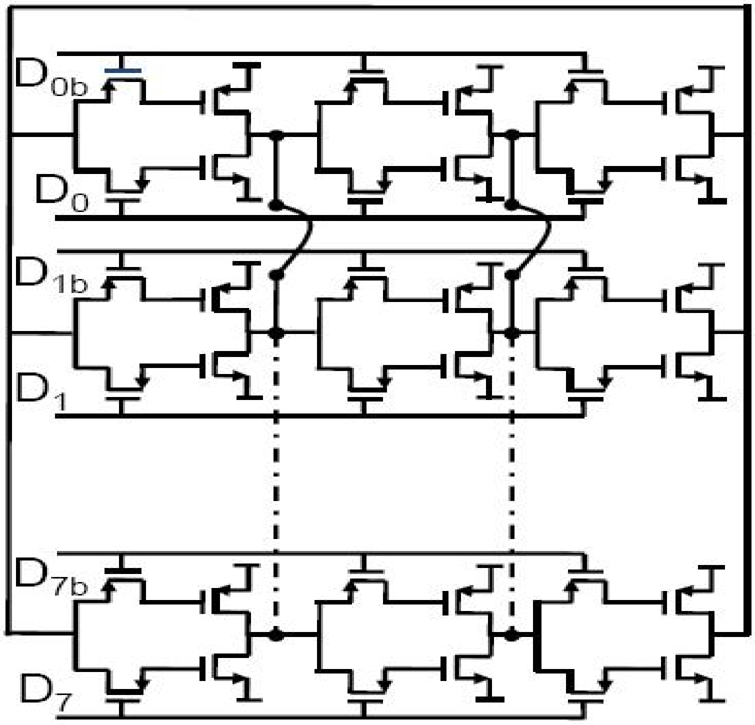


Figure 17 Schematic of [8] DCO

As the above survey shows, ring oscillators with shunt capacitive loads implemented with NMOS transistors can achieve good resolution and linearity while providing a large frequency range at reasonable area.

CHAPTER 3

OVERVIEW OF THE TEST & CHARACTERIZATION PLATFORM

This chapter gives an overview of the targeted test and characterization platform [49] and describes its components in details. Figure 18 shows the general architecture of the test and characterization platform. Unlike many previous techniques which either use a test circuit that is entirely on-chip with the device under test or entirely off the DUT's chip, the new method uses a hybrid approach. Also, unlike the approach in [20] where voltage and clock controllers are integrated on the DUT's chip while the test controller could be off-chip, this method provides a general way for applying stimuli and capturing results with fixed interfaces (i.e. the same test controller can be used to test and characterize any circuit). Also, unlike the approach in [20] no BIST circuitry is required. The test controller (TACP) can be implemented on an ASIC or a Field-Programmable Gate Array (FPGA). The TACP could be interfaced to a PC for receiving test instructions and data and sending the test results. The TACP's on-chip support circuitry provides the fixed interface (Figure 19) to the TACP and the controlled clock source for the IUTs. All interfaces use serial data communications to save I/O pins [49].

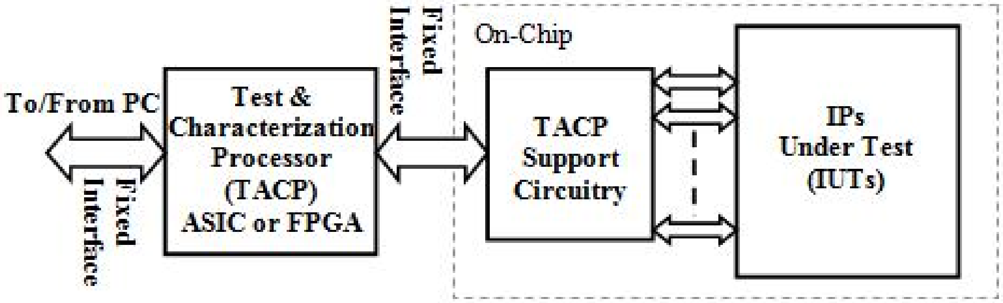


Figure 18 Overview of the proposed test and characterization method [49]

3.1 The TACP Support circuitry (TSC)

The TACP support circuitry (TSC), shown in Figure 20, performs the following functions:

- **Port Selection**: The proposed method supports testing and characterization of unlimited number of IPs on the prototype chip. Each IP could also have several input/output ports for different purposes (functional I/Os and scan I/Os). The TSC provides a mean to select a specific port to apply/receive test data to/from.
- **Serial-to-Parallel and Parallel-to-Serial data conversion (SERDES):** To have fixed logic interfaces between the TACP and the prototype chip all data communications are serial. As such, the TSC converts the received serial test data to parallel data to be applied to the IUT. It also converts back the captured test results from parallel form to serial form.
- **Controlled Clock Source:** All data transfer between the TACP and the prototype chip and functional characterization is carried out using the TACP relatively low frequency clock to ease the design of the interface. For speed characterization, a

high speed digitally controlled oscillator is provided as part of the TSC. The user can increase/decrease this oscillator frequency and use it for at speed testing of his/her IP(s).

Figure 19 shows the interface between the TACP and the prototype chip. This interface is fixed and will not change with any chip being tested or characterized. Figure 20 shows a block diagram of the TSC. The main components are the configurable clock generator, the port selection block, test application ports (TAPs), and test result ports (TRPs).

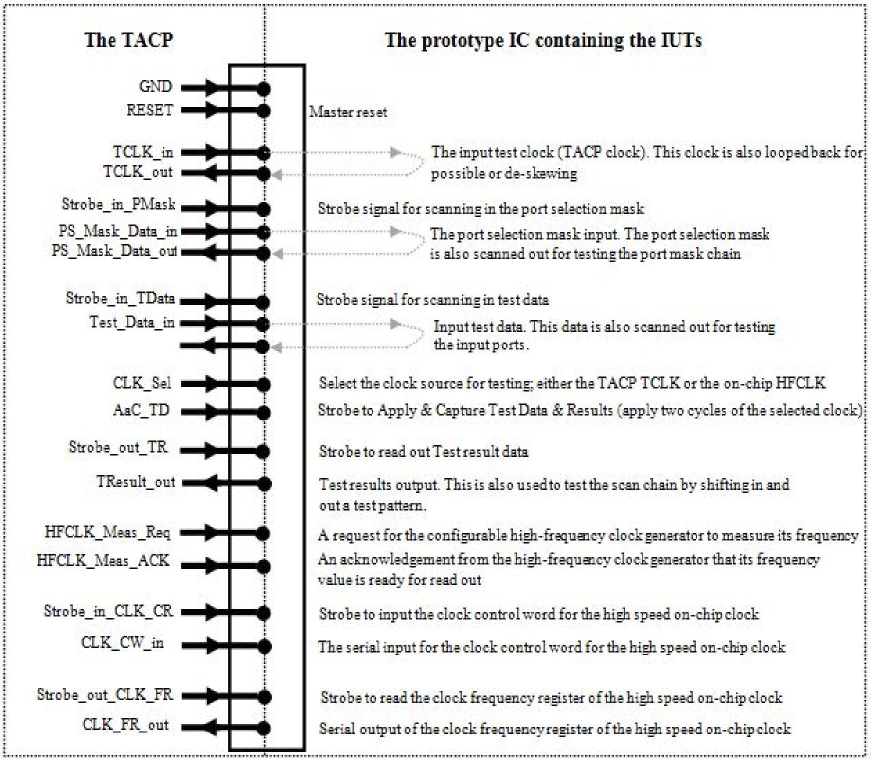

The TACP
The prototype IC containing the IUTs
GND
RESET
Master reset
TCLK_in
TCLK_out
The input test clock (TACP clock). This clock is also looped back for possible or de-skewing
Strobe_in_PMask
Strobe signal for scanning in the port selection mask
PS_Mask_Data_in
PS_Mask_Data_out
The port selection mask input. The port selection mask is also scanned out for testing the port mask chain
Strobe_in_TData
Strobe signal for scanning in test data
Test_Data_in
Input test data. This data is also scanned out for testing the input ports.
CLK_Sel
Select the clock source for testing; either the TACP TCLK or the on-chip HFCLK
AaC_TD
Strobe to Apply & Capture Test Data & Results (apply two cycles of the selected clock)
Strobe_out_TR
Strobe to read out Test result data
TResult_out
Test results output. This is also used to test the scan chain by shifting in and out a test pattern.
HFCLK_Meas_Req
A request for the configurable high-frequency clock generator to measure its frequency
HFCLK_Meas_ACK
An acknowledgement from the high-frequency clock generator that its frequency value is ready for read out
Strobe_in_CLK_CR
Strobe to input the clock control word for the high speed on-chip clock
CLK_CW_in
The serial input for the clock control word for the high speed on-chip clock
Strobe_out_CLK_FR
Strobe to read the clock frequency register of the high speed on-chip clock
CLK_FR_out
Serial output of the clock frequency register of the high speed on-chip clock

Figure 19 The fixed interface signals between the TACP and the prototyping chip

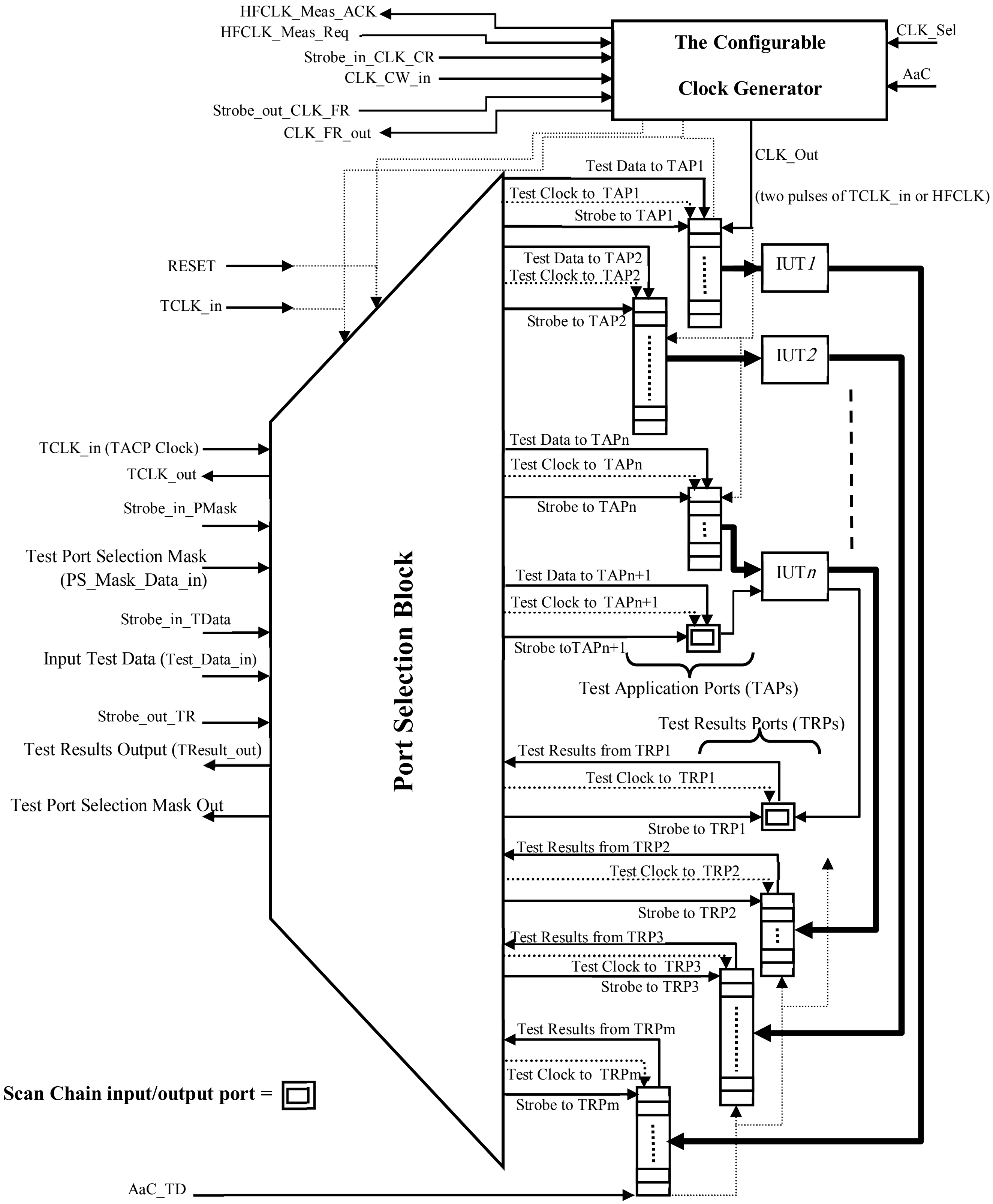
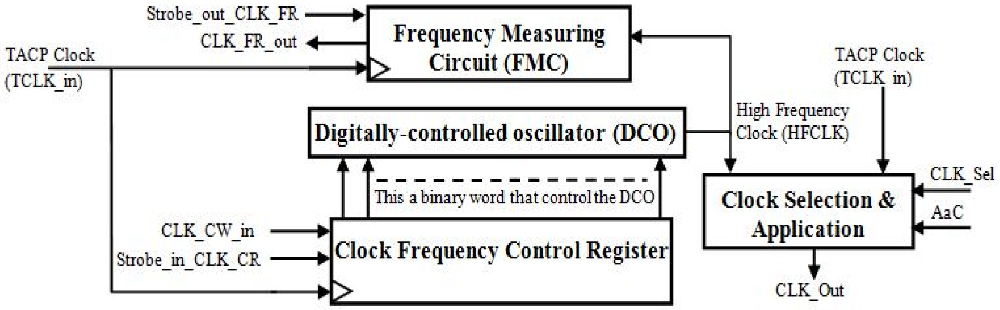


Figure 20 Block diagram of the TACP Support Circuitry (TSC) to be placed on the prototype chip

3.1.1 The Configurable Clock Generator

As mentioned before, the regular test clock is coming from the TACP which is off-chip. This clock is kept at a moderate frequency (50~100 MHz). Hence no special high-frequency transceivers or signal traces are required. This eases the design of the interface and keeps its cost to a minimum. At the same time this clock is adequate for scanning in/out the test data/results and performing functional characterization of the IUTs. Frequency characterization, however, requires a clock source that can be configured to produce a high-frequency clock. This configurable source is placed on the prototype chip and dubbed the Configurable Clock Generator in Figure 21. This generator, as illustrated in Figure 21, is made up of a frequency measuring circuit (FMC), Figure 22 and Figure 23, a clock frequency control register, Figure 24, and a clock selection and application circuit, Figure 25.

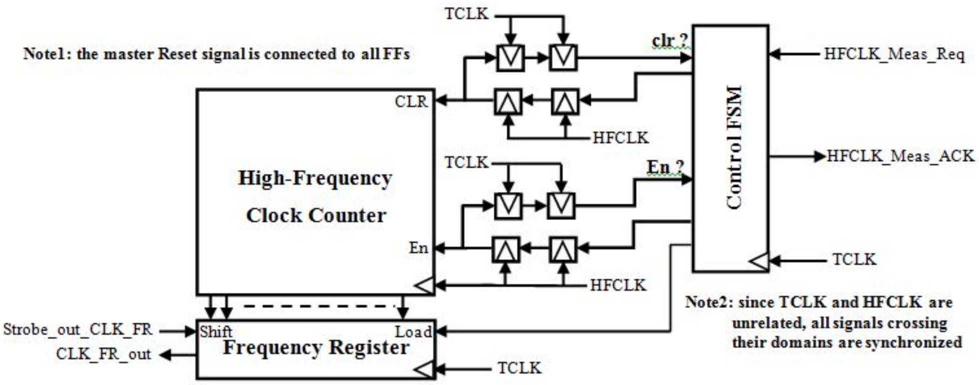


Strobe_out_CLK_FR
CLK_FR_out
Frequency Measuring Circuit (FMC)
TACP Clock (TCLK_in)
Digitally-controlled oscillator (DCO)
High Frequency Clock (HFCLK)
TACP Clock (TCLK_in)
CLK_Sel
This a binary word that control the DCO
Clock Selection & Application
AaC
CLK_CW_in
Strobe_in_CLK_CR
Clock Frequency Control Register
CLK_Out

Figure 21 The configurable clock generator

3.1.2 The Frequency Measuring Circuit (FMC)

The FMC, simply counts the number of high-frequency clock cycles within a certain period and puts the result in a shift register that would be shifted out by the TACP using the Strobe_out_CLK_FR strobe signal and through the CLK_FR_out pin. The measurement period is specified by the TACP as the difference between activating the measurement request (HFCLK_Meas_Req) and deactivating the request. When the FMC is done it activates the acknowledgement signal (HFCLK_Meas_ACK) which remains high till a new measurement request is received. The detailed design of the FCM including its controller's state diagram and its operation is shown in Figure 23. The user can control the accuracy of the measurement by having a longer measurement period.


Note1: the master Reset signal is connected to all FFs

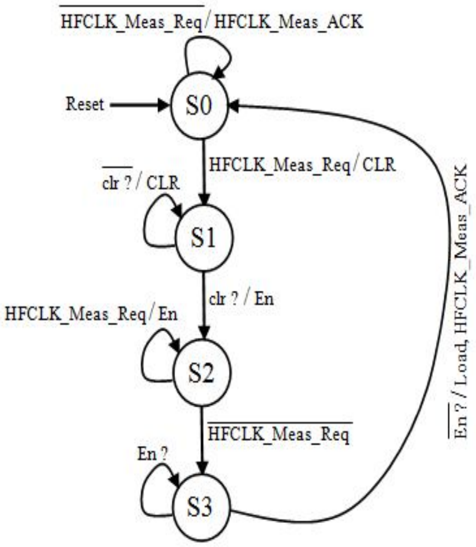
Note2: since TCLK and HFCLK are unrelated, all signals crossing their domains are synchronized

Figure 22 The frequency measuring Circuit (FCM)

3.1.3 The Clock Selection and Application Circuit (CSaAC)

The clock selection and application circuit (CSaAC), Figure 25, is responsible for selecting the required test clock (based on the CLK_Sel input signal from the TACP) and applying exactly two pulses of that clock to the selected TAP/TRP ports (in response to a strobe on the AaC input). The TACP triggers the CSaAC by setting the AaC signal to high for at least two cycles of the selected clock (Sel_CLK). The CSaAC will produce exactly two pluses of the selected clock for each AaC pulse, but in order for this circuit to fire again, the AaC signal must be reset for at least two cycles of the selected clock. The clock gating circuit ensures that the two pulses applied are complete with no glitches by enabling the output clock when the selected clock is low. The only constraint for this circuit is that the sum of the clock inverter delay, the FF's clock to Q delay and the clock-gating AND gate delay is less than the width of the negative pulse of the selected clock. Also, due to the required synchronization of the AaC input with the selected clock (3 FF synchronizer is used), the output clock pulses will have a latency of 3 cycles of the selected clock. The TACP takes care of all these issues by applying the AaC signal for two TCLK_in cycles (TCLK_in frequency is always ≤ than the selected clock frequency) and then resetting it for two more cycles before setting it again (in case of successive apply and capture commands).

Figure 26 shows logic simulation results of the CSaAC with unit gate delays. Figure 26(a) shows how the circuit functions correctly when the AaC pulse is at least two cycles of Sel_CLK and the so is the reset time in between AaC pulses. When the AaC pulse is less than two cycles or the reset time in between pulses is less than two cycles, the circuit fails, as shown in Figures 26(b) and 26(c), respectively.



$\overline{\text{HFCLK\_Meas\_Req}}$ / HFCLK_Meas_ACK
Reset
S0
HFCLK_Meas_Req / CLR
$\overline{\text{clr}}$? / CLR
S1
clr ? / En
HFCLK_Meas_Req / En
S2
$\overline{\text{HFCLK\_Meas\_Req}}$
En ?
S3
$\overline{\text{En}}$? / Load, HFCLK_Meas_ACK

Control FSM operation (state diagram on the left)

S0: waiting for a frequency measurement request (i.e. activation of the ***HFCLK_Meas_Req*** signal) from the TACP to start. The acknowledgement signal is held high. Once the request is activated, the clear (***CLR***) input of the high-frequency counter is activated and a state transition to S1 is carried out.

S1: The clear signal is held high waiting for the high frequency clock counter to be cleared. This wait is due to the unknown delay of the synchronizer. Once a clear is detected, the clear signal is de-activated, the ***En*** signal is activated and a state transition to S2 is carried out.

S2: ***En*** signal is held high counting the HFCLK cycles and waiting for the measurement request de-activation by the TACP. Upon de-activation of the measurement request the En signal is de-activated and a state transition to S3 is carried out.

S3: The ***En*** signal is held low waiting for synchronization. Once the de-activation of the ***En*** signal is detected (as ***En ?*** signal going low), the ***Load*** (load control input of the frequency register) and ***HFCLK_Meas_ACK*** signals are activated indicating measurement is done and frequency value can be read from the frequency register. The circuit goes back to state S0.

Figure 23 The state diagram of the control unit of the FMC

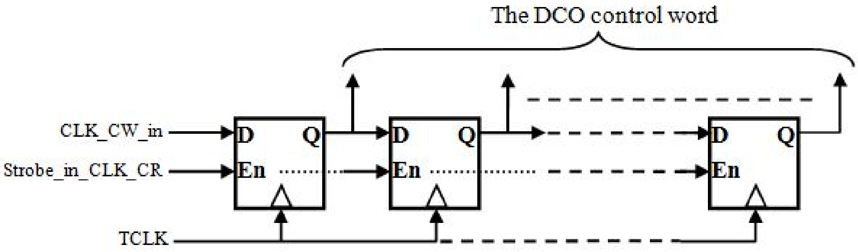


Figure 24 The Clock Frequency Control Register

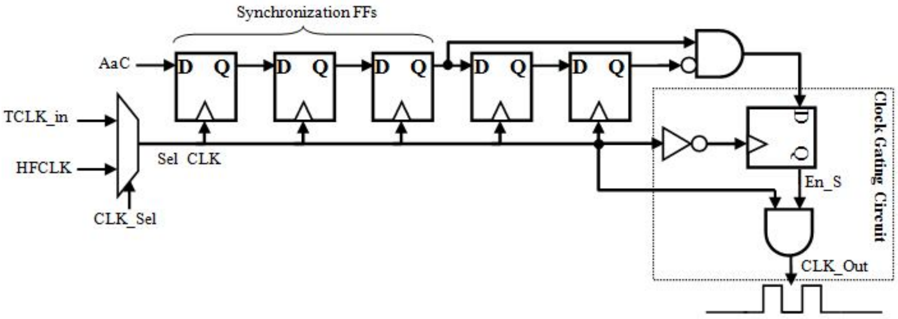
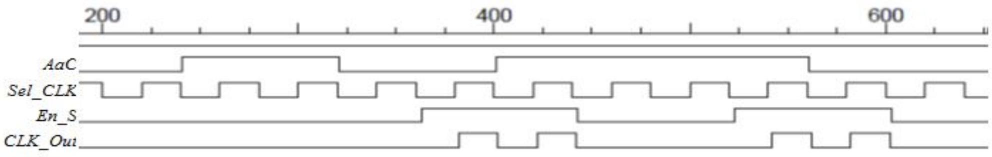


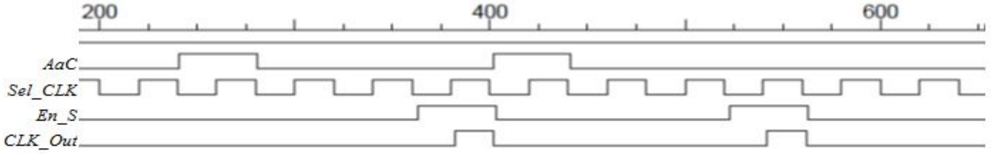
Figure 25 The Clock Selection and Application Circuit

3.1.4 The Port Selection Block

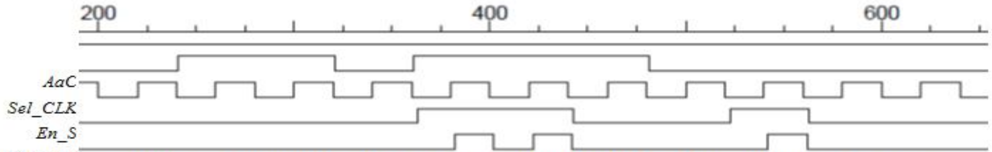
This block is responsible for selecting a specific test application/test result port to deliver the strobes, test clock and input test data to or receive test results from. The user can select a single input/output port or two ports (one input and one output). To make this block general yet with a fixed interface to the TACP, it is made up by cascading a basic cell as shown in Figure 27. The selection mask is loaded serially through the PS_Mask_Data_in input using the Strobe_in_PMask strobe signal. The TACP supports variable length selection mask (up to 216 bits). The port selection mask is also read out through PS_Mask_Data_out for testing the selection chain.



(a) Correct operation of the CSaAC with AaC pulses of at least two Cycles of Sel_CLK



(b) Incorrect operation of the CSaAC due to insufficient AaC pulse width



(c) Incorrect operation of the CSaAC due to insufficient reset time between AaC pulses

Figure 26 Logic Simulation Results for the CSaAC [49]

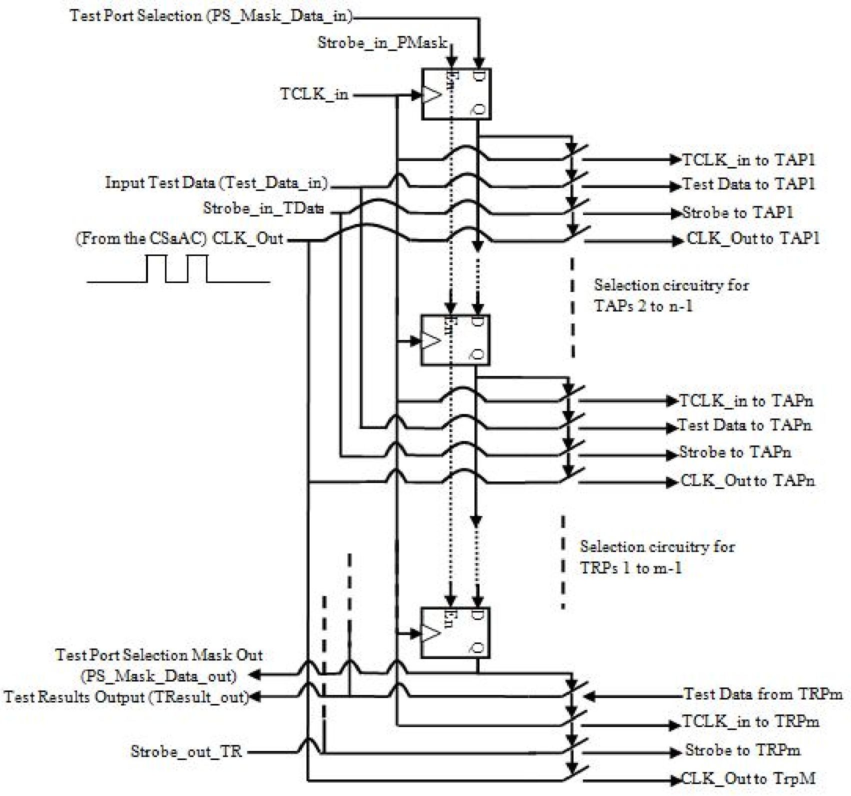


Figure 27 The Port Selection Circuitry

3.1.5 The Test Application/Result Ports (TAP/TRP)

There are two types of test application/result ports as was illustrated in Figure 20. The first type, shown in Figures 28 and 29, are used for applying and capturing primary inputs/outputs of an IUT. These are similar to boundary scan ports and are made of shift registers for scanning in/out the test data/results and parallel-load registers for applying/capturing the test data/results. As Figure 28 shows, each TAP (or TRP) is made of a cascaded number of identical cells equal to the port's data width. The shift registers use the TCLK_in and the application/capture registers use the selected apply and capture clock (CLK_Out). The CLK_Out clock is also used for the IUT's internal registers. For the TRP, the TACP needs to apply at least one TCLK_in cycle (to load the test results into the shift register) before activating the Strobe_out_TR signal to read out the results.

IP designers may also need to use full-scan designs in addition to/or instead of boundary-scan. This requires making all or part of the internal Flip Flops scanable (forming one long scan chain). Such scan chains could be used for debugging/diagnostics of an IUT internal circuitry or to fully test a sequential circuit which is difficult to do using only primary inputs/outputs. Special TAP/TRP scan ports were developed for scan chain inputs/outputs of IUTs, as shown in Figure 30. These ports have to be used (i.e. selected) in pairs where data is shifted through the chain when either the Strobe_in_TR or the Strobe_out_TR signals is activated. The TCLK_in, Scan_En and CLK_Out signals are made available for the internal scan FFs of the IUT. Regular TAP/TRP ports are used for non-scan primary inputs and outputs of the IUT. The TACP instructions support shifting test data in, shifting test results out, or simultaneous shifting in and out of test data and results, respectively.

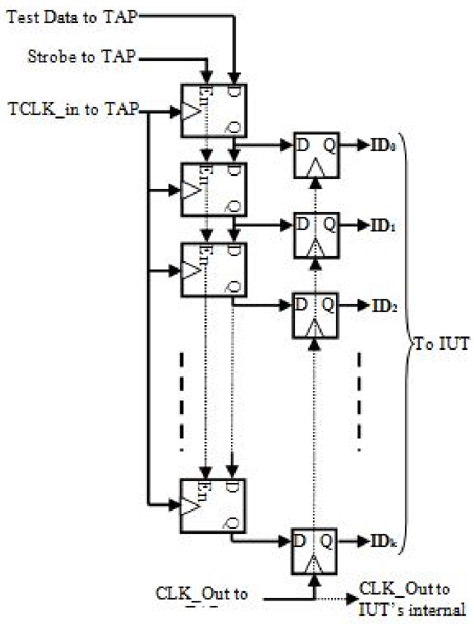


Figure 28 k-bits wide test application port (TAP)

Note: At least one TCLK_in cycle is needed before the strobe signal to the TRP is activated (to write the test results into the output chain).

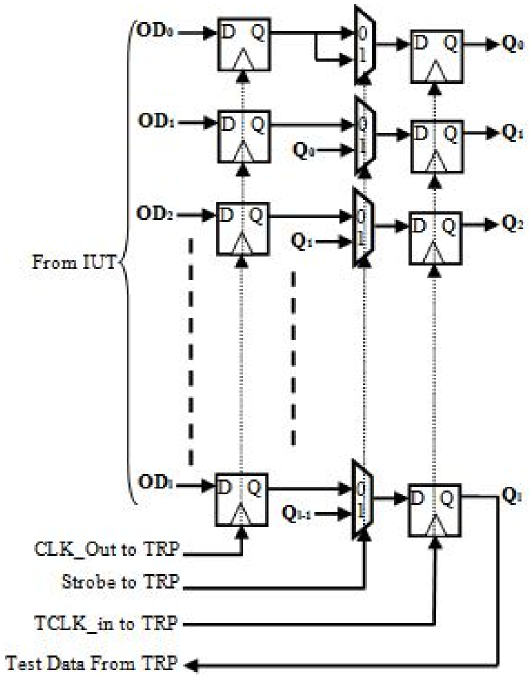


Figure 29 l-bits wide test results port (TRP)

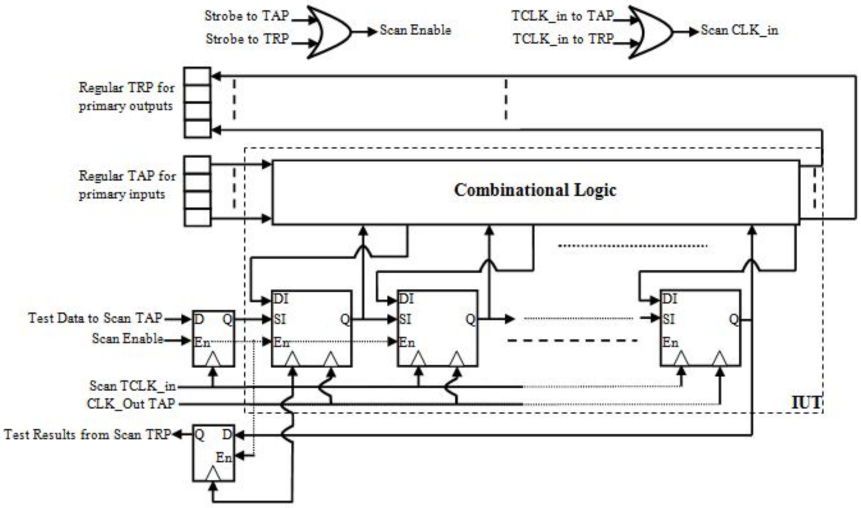


Figure 30 Scan test application/result ports

3.2 The TACP

The TACP is a special-purpose processor implemented as a data path and a micro-coded control unit. The data path is made of shift registers for shifting data out/in and counters to count the number of data bits shifted. The control unit decodes the test instructions, loads the counters with the length of bit streams and controls the shift registers. The TACP's memory is partitioned into three partitions; one for instructions, another for test data and another for test results. The memory is under direct control of the host computer that would down load test instructions and data and upload test results. The host communicates with the TACP via a very simple protocol using fixed size packets with a header that specifies the packet type and the required action from the TACP. Special interactive configuration commands allow the host to read the on-chip HFCLK frequency, increase it or decrease it. Other commands also allow the host to set the TACP frequency (TCLK) and read out its internal registers for debugging purposes. The TACP has special instructions for port selection, shifting out test data, shifting in test results, apply and capture test data/results, comparing test results with some value, conditional statements and loops. The simple TACP architecture allows the addition of more instructions if needed in future revisions.

CHAPTER 4

THE DIGITALLY CONTROLLED RING OSCILLATOR

As was explained in chapter 3, the DCO is at the heart of the configurable clock generator. In this chapter, the details of the DCO including its design and portability are presented. The DCO was designed to give frequency range from the maximum possible frequency to at least half of it. Since frequency binary dividers divide the frequency by multiple of two, the DCO should be controllable so that it covers the range (i.e. from the maximum frequency to at least half of it). In addition, the frequency characteristics of it should be monotonic to make it suitable for at-speed testing. Section 4.1 reports the design of DCO and its monotonicity and section 4.2 shows the DCO portability.

4.1 Design of the DCO

The oscillator that is used in this work is a digitally-controlled all digital ring oscillator (consists of three stages only), Figure 31. One of the stages is replaced with a 2/IP NAND gate for starting the oscillator correctly and controlling its startup time and to minimize power consumption since the oscillation stops when the Run signal is low. NMOS switched-load capacitances are added for two of the stages with 4-bit control each, to get reasonable frequency resolution (32 steps), and binary-weighted sizing is used for the capacitors and the NMOS switches, giving a ratio of two between minimum and maximum frequencies. This represents the basic frequency range of the oscillator. The frequency range is extended using division by a multiple of 2 using a binary counter,

Figure 32. All NMOS transistors sizes (W) of the inverters are kept to the minimum and all PMOS transistors sizes are double the size of the NMOS transistors to make their equivalent resistances approximately equal.

The capacitors and the NMOS switches (i.e. the control bits) should not be all in one stage output; but they have to be distributed on at least two stages depending on number of stages to increase their effect. Since a 3-stage ring oscillator was used, eight shunt capacitors were distributed evenly between the two stages as shown in Figure 31.

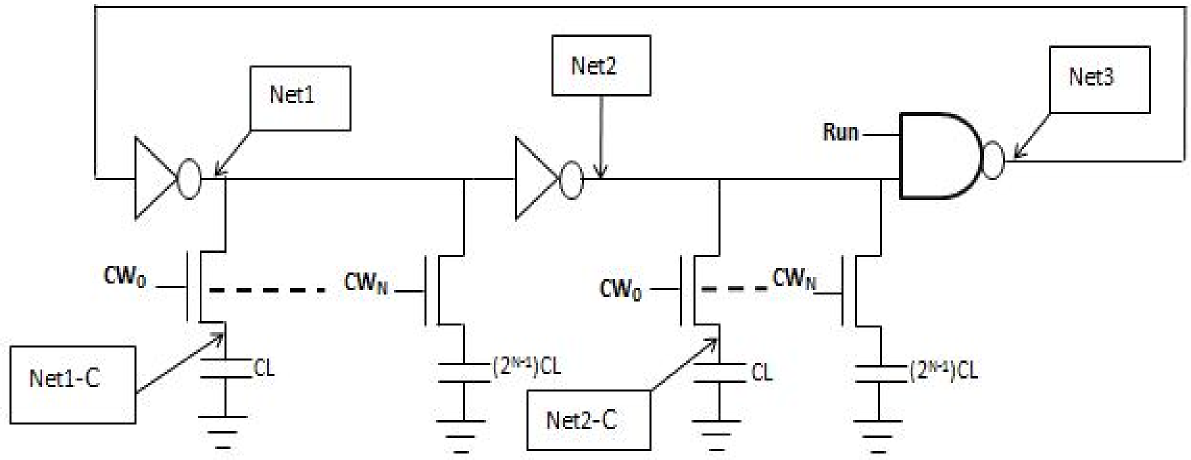


Figure 31 The utilized digitally controlled oscillator (DCO)

The unit capacitance CL (i.e. the capacitance of the varactor-connected NMOS device) required to ensure that the basic DCO's frequency range is 2x can be estimated as follows; ignoring the NAND gate delay. Let T_{Dmin} and T_{Dmax} be the minimum and maximum delay of one cycle of the DCO.

$$2T_{Dmin} \leq T_{Dmax} \qquad (1)$$

T_{Dmax} is twice or greater than T_{Dmin} to maintain 2x frequency range. If this condition is violated, there will be some frequencies that the DCO will not cover.

$$T_{Dmin} \approx 2R_{eq}\left(C_{in} + 2NC_j\right) \qquad (2)$$

C_{in} is the input capacitance of the DCO stage, R_{eq} is NMOS and PMOS equivalent resistance assuming that their equivalent resistances are equal, C_j is the drain junction capacitances of the NMOS switch and N is number of control bits in each stage. Since control bits are used in two stages of the DCO, C_j is multiplied by two. C_j is also multiplied by N because there are N C_j capacitances in each stage.

$$T_{Dmax} \approx 2R_{eq}\left(C_{in} + 2NC_j\right) + \left(\frac{2^{N+1}R_{on}}{2^{N+1} - 2}\right)\left(2NC_j + \left(2^N - 1\right)C_L\right) \qquad (3)$$

R_{on} is the equivalent resistance of the first NMOS switch and CL is the required unit capacitance. Since the control bits in each stage are binary weighted, their total capacitance is the sum of them$(\mathrm{i.e.}\, 2^N - 1)$. Furthermore, the equivalent resistance of each stage is the resultant of the parallel R_{on}.

The minimum delay is obtained when all NMOS switches are off and the maximum delay is obtained when they are all on. When the switches are off, there is no capacitance added to the DCO except C_{in} and C_j, while CL is added through the R_{on} equivalent resistance when the switches are on. Hence, the switches should be binary weighted to utilize the binary weighted varactors through it i.e. the equivalent resistance should be decreased to increase the effect of the varactors.

Substituting equations (2) and (3) in equation (1) implies that,

$$4\mathrm{R_{eq}}(\mathrm{C_{in}} + 2\mathrm{NC_j}) \le 2\mathrm{R_{eq}}(\mathrm{C_{in}} + 2\mathrm{NC_j}) + \left(\frac{2^{\mathrm{N}+1}\mathrm{R_{on}}}{2^{\mathrm{N}+1} - 2}\right)(2\mathrm{NC_j} + (2^{\mathrm{N}} - 1)\mathrm{C_L}) \quad (4)$$

By simplifying (4) we get,

$$4\mathrm{R_{eq}}(\mathrm{C_{in}} + 2\mathrm{NC_j}) - 2\mathrm{R_{eq}}(\mathrm{C_{in}} + 2\mathrm{NC_j}) \le \left(\frac{2^{\mathrm{N}+1}\mathrm{R_{on}}}{2^{\mathrm{N}+1} - 2}\right)(2\mathrm{NC_j} + (2^{\mathrm{N}} - 1)\mathrm{C_L})$$

$$\mathrm{R_{eq}}(2\mathrm{C_{in}} + 4\mathrm{NC_j}) \le \left(\frac{2^{\mathrm{N}+1}\mathrm{R_{on}}}{2^{\mathrm{N}+1} - 2}\right)(2\mathrm{NC_j} + (2^{\mathrm{N}} - 1)\mathrm{C_L})$$

$$\frac{(2^{\mathrm{N}+1} - 2) * \mathrm{R_{eq}}(2\mathrm{C_{in}} + 4\mathrm{NC_j})}{2^{\mathrm{N}+1}\mathrm{R_{on}}} \le (2\mathrm{NC_j} + (2^{\mathrm{N}} - 1)\mathrm{C_L})$$

Assuming that $\mathrm{R_{eq}} = \mathrm{R_{on}}$,

$$\frac{(2^{\mathrm{N}+1} - 2)(2\mathrm{C_{in}} + 4\mathrm{NC_j})}{2^{\mathrm{N}+1}} \le (2\mathrm{NC_j} + (2^{\mathrm{N}} - 1)\mathrm{C_L})$$

$$\frac{(2^{\mathrm{N}+1} - 2)(2\mathrm{C_{in}} + 4\mathrm{NC_j})}{2^{\mathrm{N}+1}} - 2\mathrm{NC_j} \le (2^{\mathrm{N}} - 1)\mathrm{C_L}$$

$$\frac{(2^{\mathrm{N}+1} - 2)(2\mathrm{C_{in}} + 4\mathrm{NC_j}) - 2^{\mathrm{N}+2}\mathrm{NC_j}}{2^{\mathrm{N}+1}} \le (2^{\mathrm{N}} - 1)\mathrm{C_L}$$

$$\frac{(2^{\mathrm{N}+1} - 2)(2\mathrm{C_{in}} + 4\mathrm{NC_j}) - 2^{\mathrm{N}+2}\mathrm{NC_j}}{2^{\mathrm{N}+1}(2^{\mathrm{N}} - 1)} \le \mathrm{C_L} \qquad (5)$$

Equation (5) can be written in terms of sizes instead of capacitances as follows:

By using $\mathrm{C_{in}} = \mathrm{C_{ox}}(\mathrm{W_p} + \mathrm{W_n})\mathrm{L}$ and $\mathrm{Cj} = (\mathrm{C_J} * \mathrm{AD} + \mathrm{C_{JSW}} * \mathrm{PD})$ and substituting them in equation (5),

$$\frac{2(2^{N+1}-2)(C_{ox}(W_p+W_n)L+2N(C_J*AD+C_{JSW}*PD))-2^{N+2}N(C_J*AD+C_{JSW}*PD)}{2^{N+1}(2^N-1)} \le C_L \qquad (6)$$

Where C_{ox} is the gate oxide capacitance per unit area, W_p and W_n are the PMOS, NMOS stage-inverter's sizes respectively, C_J is the junction capacitance per unit area, C_{JSW} is the junction capacitance per unit length, AD and PD are the drain area, drain periphery respectively. C_{ox}, C_J and C_{JSW} capacitances are all process dependent and can be found in process models.

The formula above is verified by estimating the value of CL and then determining the actual value by simulation for three CMOS process technologies for 3-stages DCO as in Table 2. N in the table represents number of control bits in each stage. The estimated CL is relatively close to the actual value which indicates the correctness of equation (5).

Table 2 CL estimation for various processes

	TSMC 0.35U			TSMC 0.180U			TSMC 0.130U		
N	Estimated CL in fF	Actual CL in fF	% of error	Estimated CL in fF	Actual CL in fF	% of error	Estimated CL in fF	Actual CL in fF	% of error
5	-0.3141	-0.3465	0.91	-0.0212	-0.0210	0.0000	-0.0125	-0.0106	15.13
4	0.6173	0.6384	3.30	0.5142	0.5400	0.0477	0.1941	0.1700	12.43
3	3.2902	3.0660	0.36	2.0820	2.0510	0.0149	0.8764	0.9100	3.69
2	14.3723	14.3900	0.12	7.2431	7.2000	0.0060	4.5558	5.9814	23.83

The negative capacitance when N=5 means that the DCO cannot have the required frequency range. As a result, the maximum number of shunt capacitors per stage for ring oscillator of three inverters is four.

Figure 32 shows the frequency and period for 120 steps and the basic range is up to 32 (using five bit code). Using all eight control bits will produce redundant codes and these codes will affect the monotonicity of the DCO period. Hence, control bits of one stage are considered as one bit giving total of five control bits. Small jump occur at the end of the basic range because the division is performed using a frequency divider.

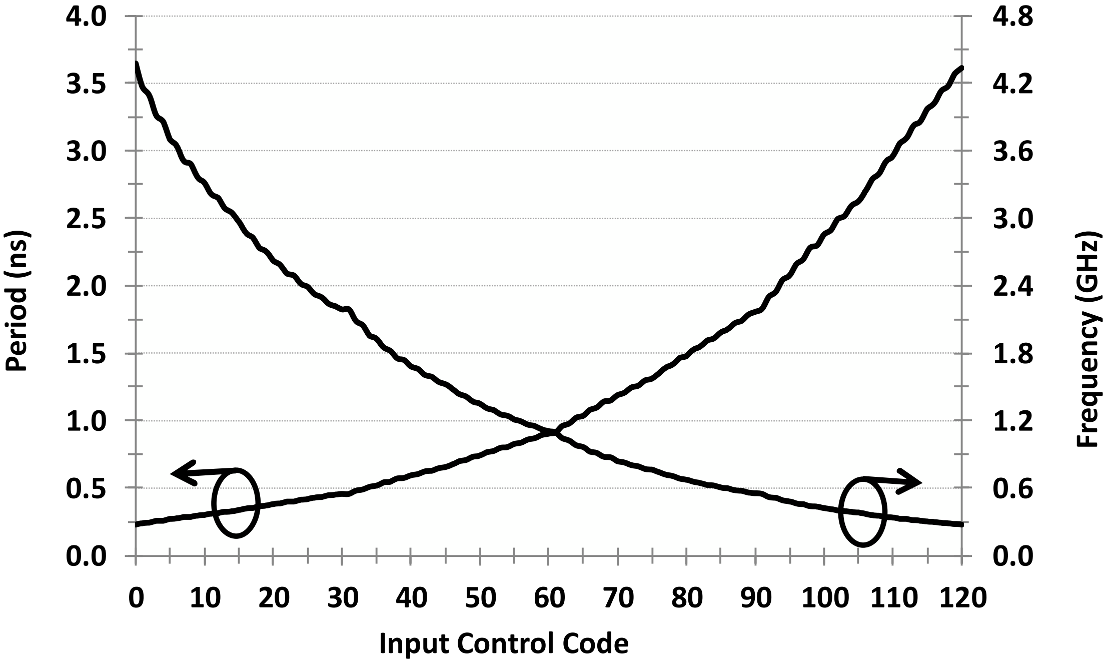


Figure 32 Period and the frequency of the DCO using TSMC 0.18U technology

4.2 DCO Portability

Portability of the DCO refers to the ability to port the DCO to any process technology (i.e. it can be ported to any CMOS technology). This is significant since the configurable clock generator is to be placed with the IUTs. Portability is illustrated for five different technologies in Figure 33 and it is obvious that the DCO is monotonic for all of these CMOS technologies. The voltage was used in TSMC 0.35U is 3.3V, TSMC 0.25U and TSMC 0.18U is 1.8V, TSMC0.13U is 1.2V and in TSMC90n is 1V. The NOMS transistor sizes of the DCO stages are 1.5*channel length (L) and the PMOS sizes are double the size of the NMOS transistor.

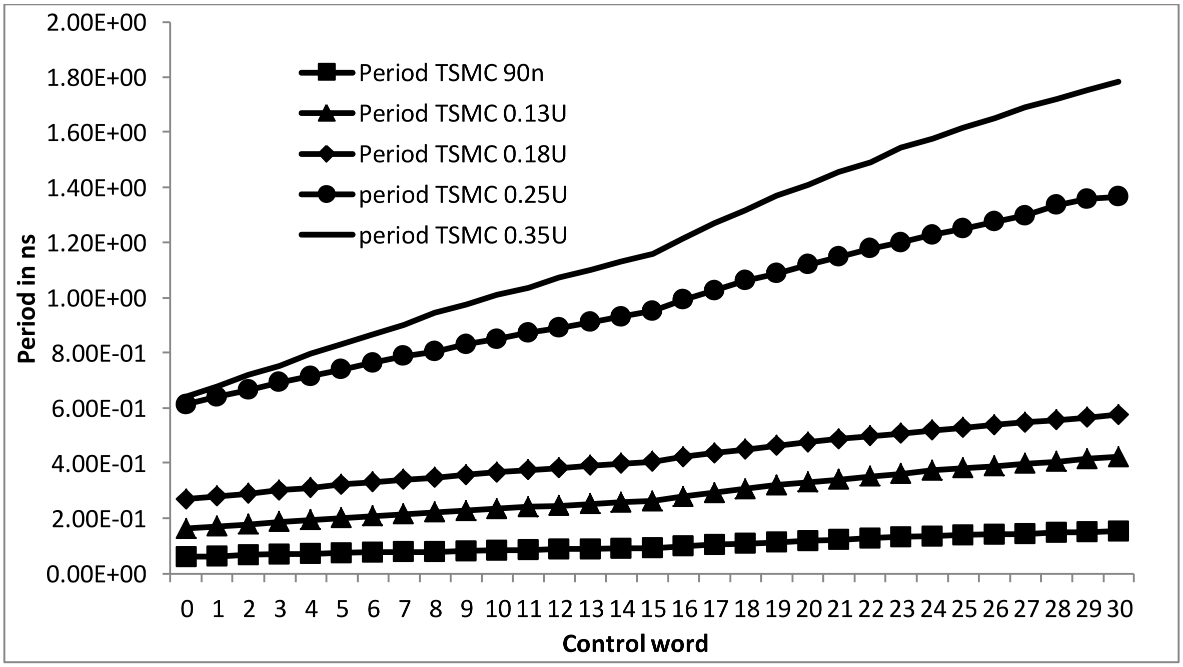


Figure 33 DCO period for various technologies

The DCO frequency characteristics are monotonic for all the technologies used to test it. However, the step between each control word and its adjacent control word has some nonlinearity due to the nature of the single-ended ring oscillator and from the switches and capacitances that control it. The capacitance is increasing and decreasing in nonlinear manner and the equivalent resistances of the inverters and the switches are varied nonlinearly. Investigation and analysis of this issue was done to estimate the unit capacitance and to predict the cause of the step nonlinearity.

The input capacitances of the stages, using TSMC 0.35U technology, are not equal for equal stages sizes due to the fact that the internal capacitances of the NMOS and PMOS are not the same when they are on and off and as the load capacitance is increased, the waveform slopes decreases which in turn increases the delay of the inverters and the NAND gate. Hence, at higher values of control word the delay step increases. This can be solved by using thermometer coding. The nonlinearity cannot be removed entirely because of the nature of the MOS devices; and these results in non-equal steps in the DCO. The linearly increasing or decreasing step is solved by using binary sized NMOS switches with the expense of slightly increasing delay. Table 3 summarizes the capacitances of the DCO when the switches are on and off in all of the three stages of DCO. When subtracting C_{in} from the capacitance at net1 when the size is W and the switch is off, for instance, the result is the C_j of the switch. This value of C_j is the same at net1-C (labeled in Figure 31) when the switch is off and is increased when the switch is on.

Table 3 DCO capacitances for several switches sizes

Switch size	Switch state	Capacitance in fF			
		net1	net2	net1-C	net2-C
W	off	10.651	10.774	1.608	1.608
	on	11.279	11.402	1.821	1.821
2W	off	12.045	12.168	3.002	3.002
	on	13.386	13.509	3.464	3.464
4W	off	14.833	14.956	5.790	5.790
	on	17.584	17.707	6.767	6.767
8W	off	20.409	20.532	11.366	11.366
	on	25.950	26.073	13.401	13.401
16W	off	31.561	31.684	22.519	22.519
	on	42.644	42.767	26.705	26.705

The binary weighted switches were introduced in this work and it proves that nonlinearity of the DCO is raised because of using non-binary switches. Table 4 shows the period and frequency of TSMC 0.35U technology by using non-binary and binary weighted switches. The Steps when using binary weighted switches are all positive while there are many negative steps when using non-binary switches which represent non-monotonicity. These negative steps are circled in Figure34.

Table 4 DCO periods for binary and non-binary switches

	Non-binary weighted switches		Binary weighted switches	
Control word	Period in ns	Step period in ns	Period in ns	Step period in ns
0	0.58		0.64	
1	0.64	0.05	0.68	0.04
2	0.65	0.02	0.72	0.04
3	0.70	0.05	0.75	0.04
4	0.68	-0.02	0.80	0.04
5	0.72	0.05	0.83	0.03
6	0.74	0.02	0.87	0.04
7	0.78	0.04	0.90	0.03
8	0.70	-0.08	0.94	0.04
9	0.75	0.05	0.97	0.03
10	0.76	0.01	1.01	0.03
11	0.81	0.04	1.04	0.03
12	0.79	-0.02	1.07	0.04
13	0.83	0.04	1.10	0.03
14	0.84	0.01	1.13	0.03
15	0.88	0.04	1.16	0.03
16	0.95	0.07	1.21	0.05
17	0.97	0.02	1.27	0.06
18	1.03	0.06	1.32	0.05
19	1.00	-0.02	1.37	0.05
20	1.06	0.06	1.41	0.04
21	1.08	0.02	1.45	0.04
22	1.13	0.05	1.49	0.04
23	1.06	-0.07	1.54	0.05
24	1.11	0.06	1.58	0.03
25	1.13	0.02	1.62	0.04
26	1.18	0.05	1.65	0.03
27	1.16	-0.02	1.69	0.04
28	1.21	0.05	1.72	0.03

29	1.23	0.02	1.75	0.03
30	1.28	0.05	1.78	0.03

Percentage of increase in the period is around 6% when control word is 0 and increased sharply to 50% when control word is 30, Figure 34. However, the increase is expected since the period obtained is linear while the period using non-binary switches is nonlinear.

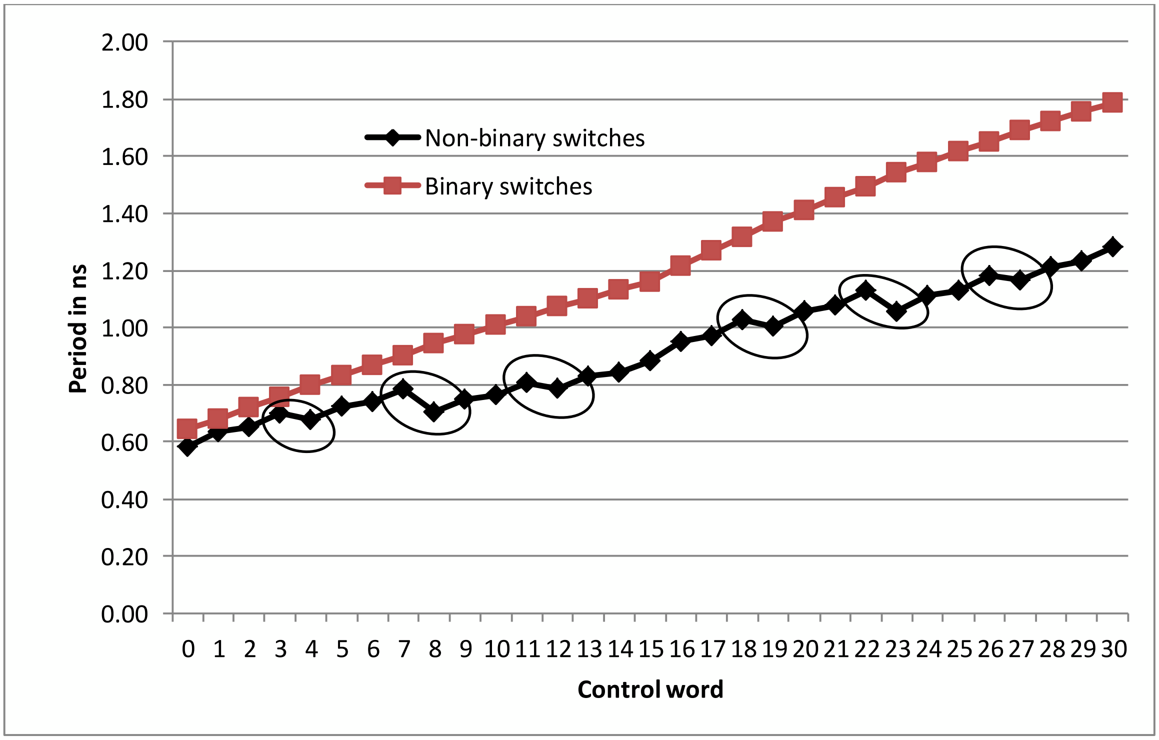


Figure 34 The effect of using binary switches on DCO period

Figure 35 shows the frequency of the DCO using TSMC 0.35U technology and the division by two and four by using normal divider. The figure illustrates the concept of using the DCO to control the frequency division that normal dividers cannot do and the use of the dividers to obtain any frequency between the maximum frequency and up to any value the user wants.

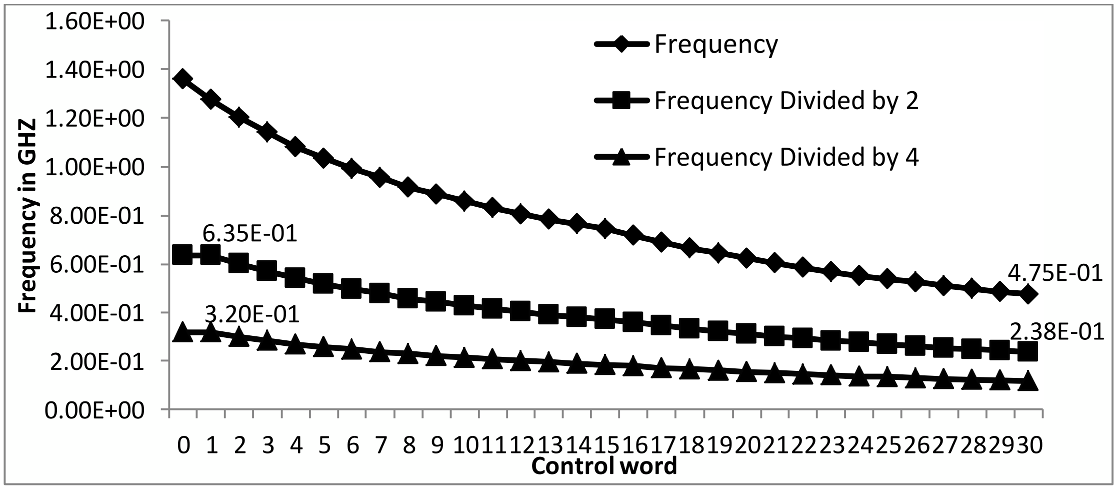


Figure 35 DCO frequency range from around 1.4GHZ to about 500MHZ, frequency range when dividing by two and by four

The results discussed in this chapter shows that the selected and enhanced DCO, Figure 31, can achieve the required frequency range with good linearity. It can also be ported to any CMOS process subject to the condition obtained in equation (5).the results also suggest that using binary-weighted NMOS switches and thermometer coding would decrease the frequency rang non-linearity.

CHAPTER 5

EXPERIMENTAL RESULTS

In This chapter, experimental results that verify the operation of the CCG are presented. There are two verification attempts; using TSMC 0.35U technology, and the second one using LF105nm technology. In the second, implementation of support circuitry that is designed using Synopsys tools for the clock generator to be tested within IUTs under the control of the TACP. The implemented support circuitry was sent for fabrication using LFoundry 150nm technology.

5.1 Implementation of the Configurable Clock Generator using TSMC 0.35U technology

The clock generator designed in this work was fabricated using TSMC 0.35U technology. The fabricated clock generator was designed with Tanner EDA tools. Figure 36 represents the design flow of the configurable clock generator and the tools that were used. The design starts with schematic entry using S-edit. Figure 37 shows the top level of the design and Appendix A shows the schematic of each component in the design. The design is simulated to verify the logic and then placed and routed. A spice netlist was extracted from the layout and simulated for the chip. The extraction can be done in two ways; the first way is normal extraction using extract option and the L-edit will extract the netlist hierarchal or flatten, and the second method is using the legacy extractor of L-edit. Legacy extractor will flatten the design and will extract the netlist and the capacitances and resistances within the netlist. Both methods were used in this work to verify the layout and it seems that legacy extraction is more accurate than the other.

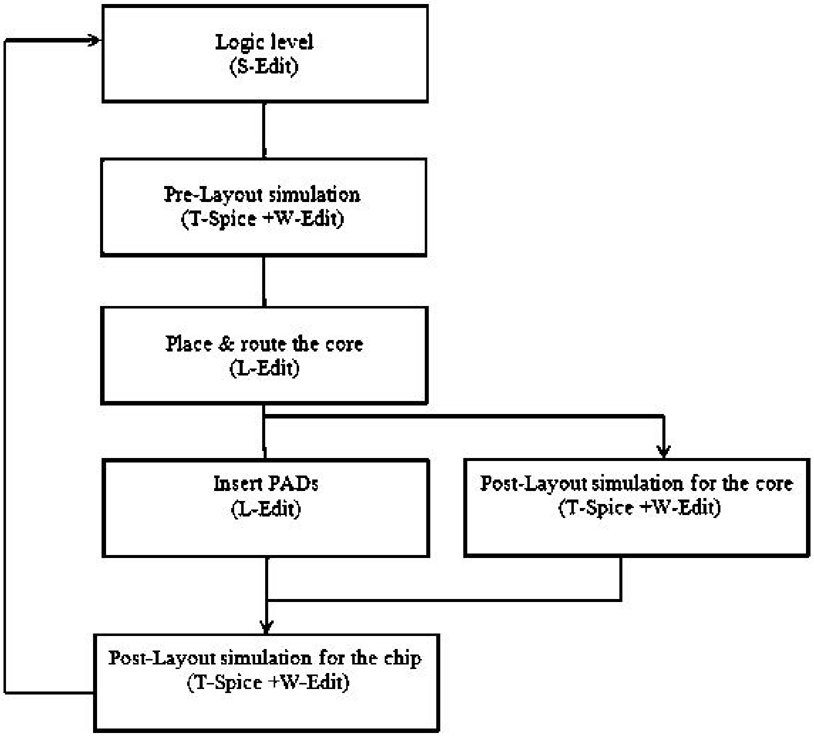


Figure 36 Flow diagram of the design in Tanner EDA

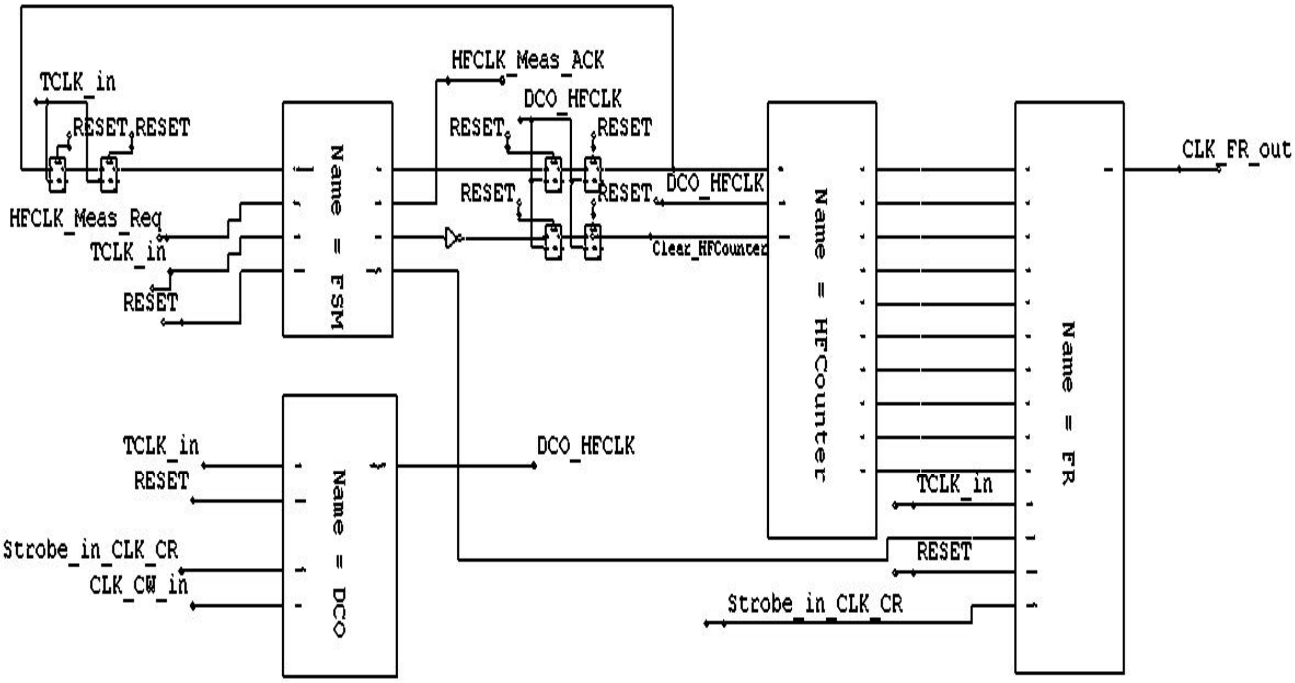
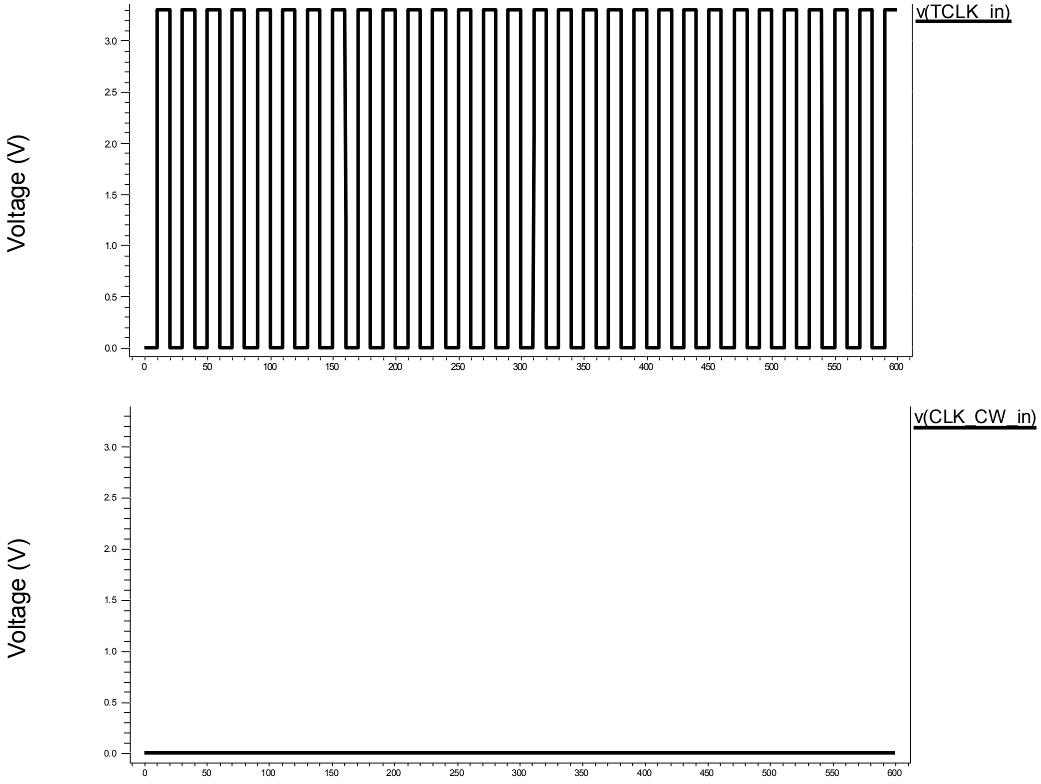


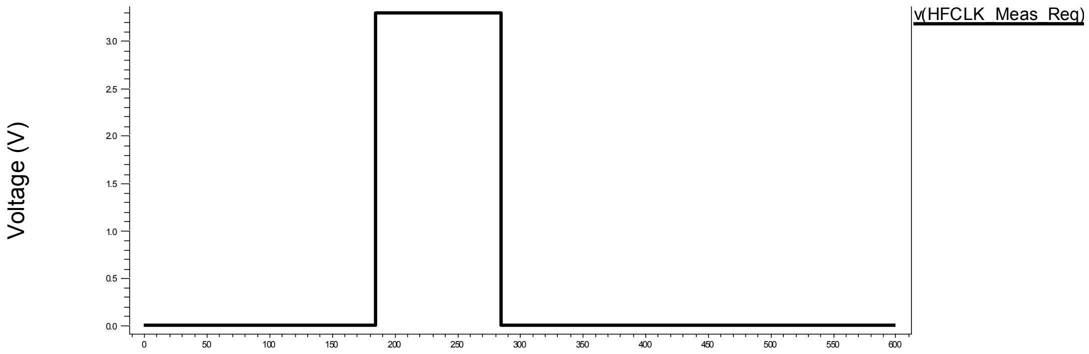
Figure 37 Top logic level of the CCG

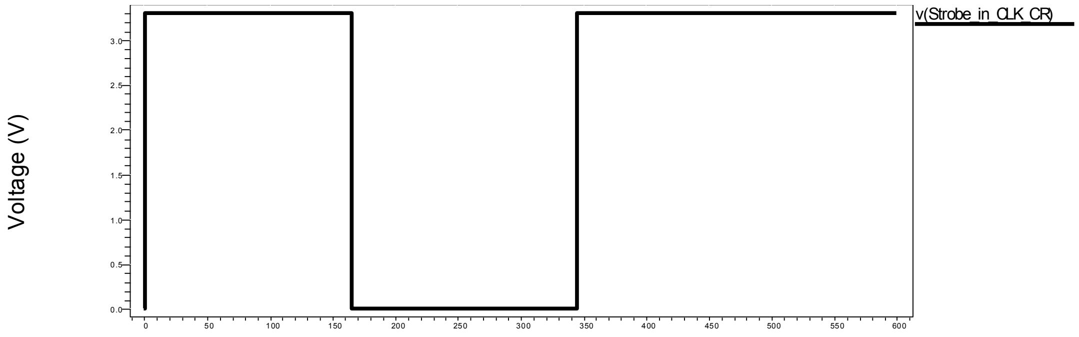
5.1.1 Pre-Layout simulation of the chip

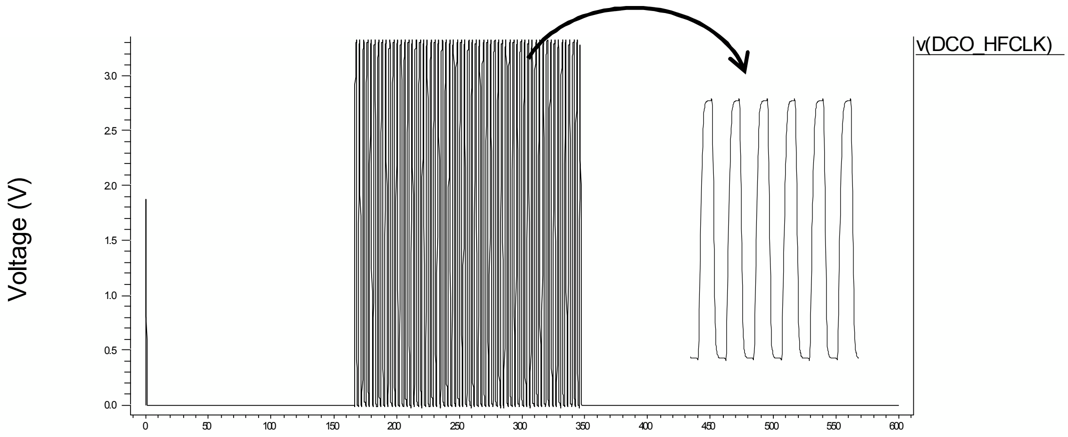
The simulation in this section was done to verify the correctness of the design using T-Spice, a spice simulator from Tanner EDA. There are five input signals, RESET, TCLK_in, CLK_CW_in, HFCLK_Meas_Req, Strobe_in_CLK_CR, and three output signals, DCO_HFCLK, HFCLK_Meas_ACK, and CLK_FR_out as in Figure 38. The reset signal is a master reset for all flip-flops in the design; TCLK_in is TACP clock which is the reference clock of the design. HFCLK_Meas_Req is the period in which the counter is activated to count number of DCO cycles and is activated for five TCLK_in cycles, Strobe_in_CLK_CR is the strobe for the shift registers to store results and/or data in, and is used also for activating the DCO, as in Figure 38, once the data is serially shifted in (i.e. when it is low), the output of DCO is DCO_HFCLK, once the results are stored the HFCLK_Meas_ACK signal is activated so that another measurement request can be processed, and number of counted DCO cycles is shifted out serially, CLK_FR_out signal.

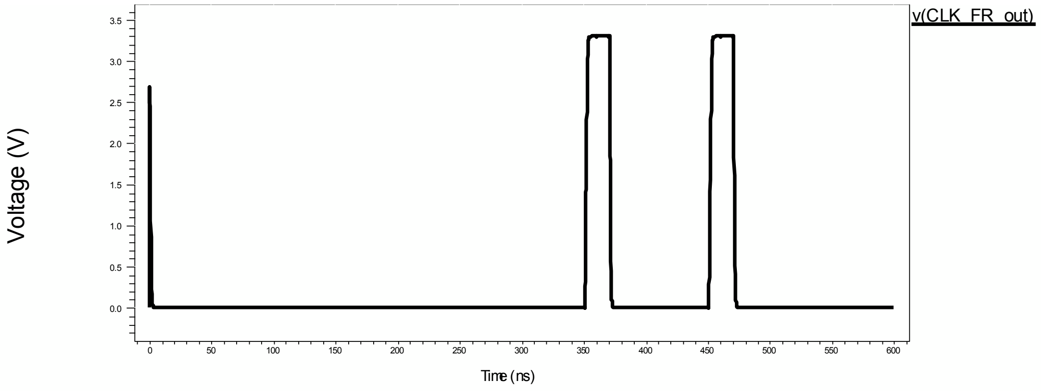
The maximum frequency (minimum period) is obtained when all switches are off, CLK_CW_in is 0. For the control word input (CLK_CW_in), the five control bits are shifted in and also three bits for the frequency selection. Hence, 0xx means no division and the output of the DCO is selected,100 means divide by 2, 101 means divide by 4, 011 means divide by 8, and 111 means divide by 16. The maximum frequency is found to be 330MHz and is observed from the high speed counter's output Figure 38, CLK_FR_out signal. Figure 38 also shows all signals of the design.




v(HFCLK_Meas_Req)
Voltage (V)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
50
100
150
200
250
300
350
400
450
500
550
600


v(Strobe_in_CLK_CR)
Voltage (V)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
50
100
150
200
250
300
350
400
450
500
550
600


v(DCO_HFCLK)
Voltage (V)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
50
100
150
200
250
300
350
400
450
500
550
600


v(CLK_FR_out)
Voltage (V)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
50
100
150
200
250
300
350
400
450
500
550
600
Time (ns)

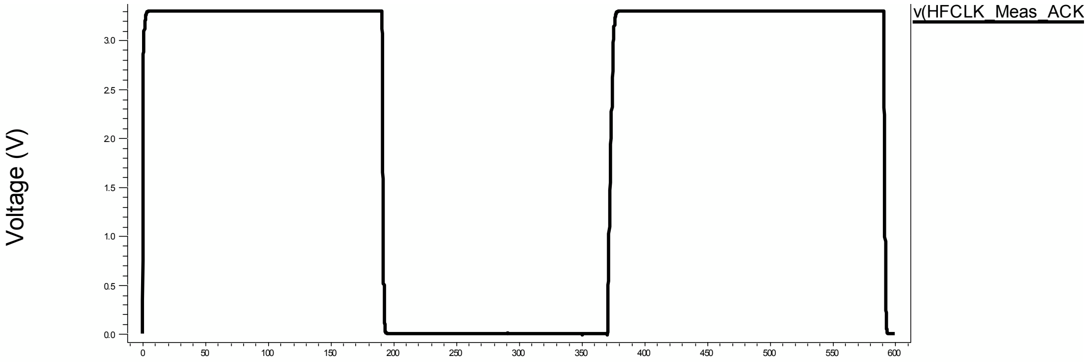
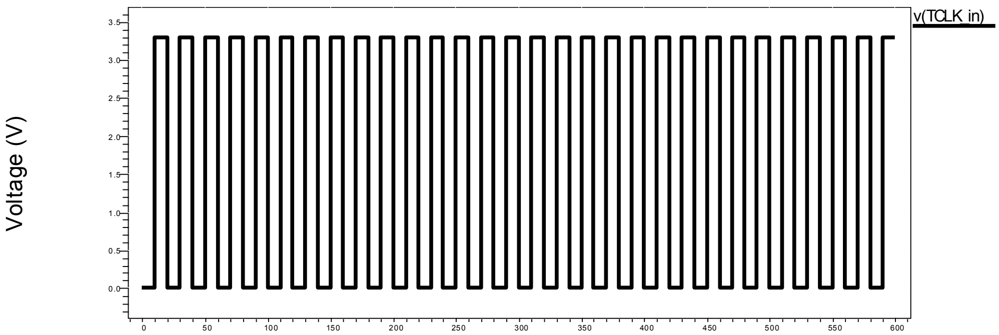
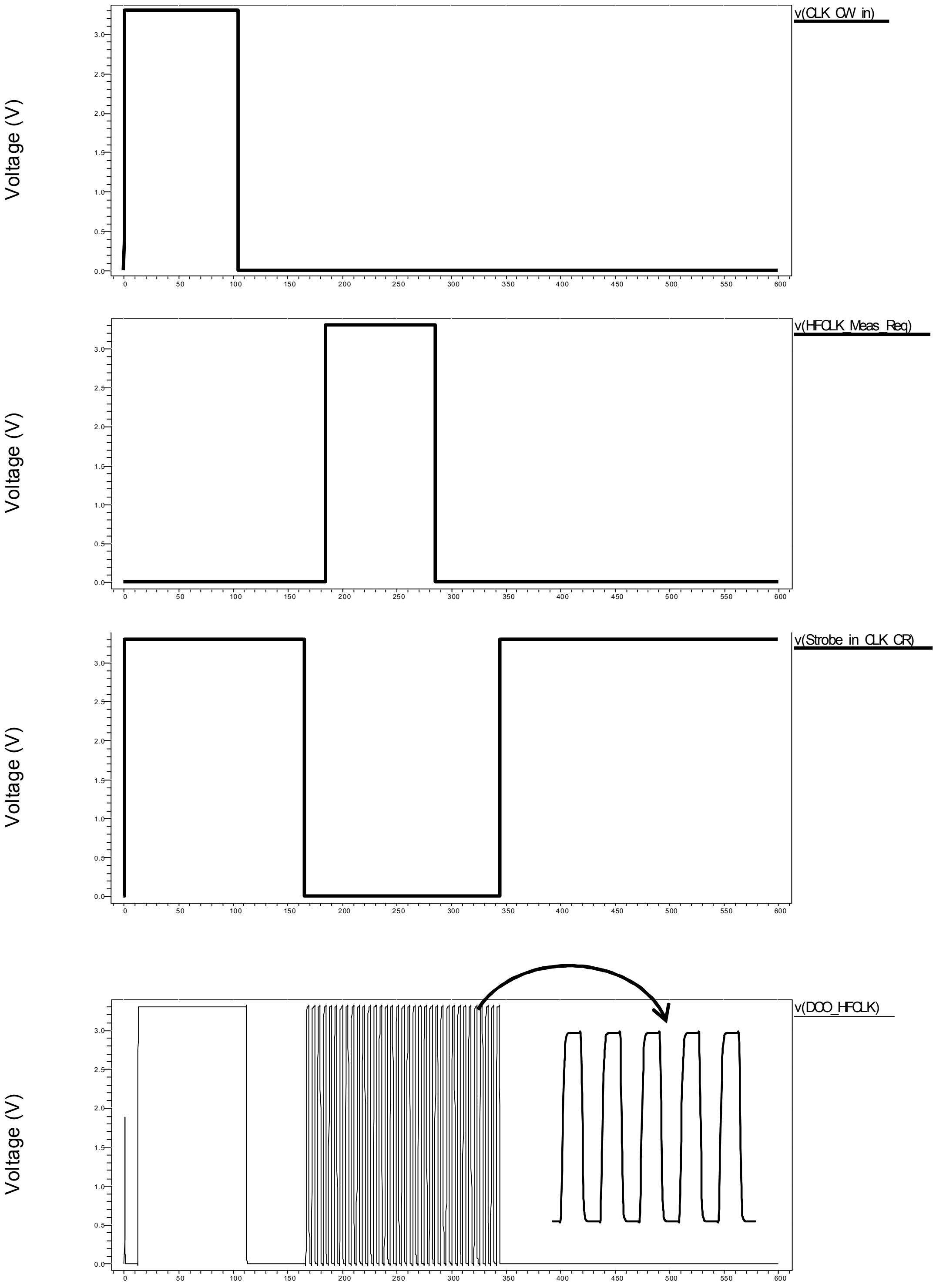


Figure 38 CCG signals (maximum frequency); pre-layout simulation

Measurement output in Figure 38 is 100001, in binary, for five TACP clocks which equal to 33 DCO cycles; Multiplying (50MHZ/5) by 33 gives 330MHz while the actual frequency, as measured from spice simulation, is 333MHZ.

Similarly, the minimum frequency is 193MHz, Figure 39. The control word in Figure 39 is 1 which means that all NMOS switches are on (CLK_CW_in is high for five TCLK_in cycles). It was noted that the DCO_HFCLK goes low and high when the control is shifted in and will oscillate normally when the shift is finished. This behavior is normal since the two inputs of the NAND gate are zeros at the beginning and one input will be inverted during the control word shifting.




Voltage (V)
v(CLK_CW_in)
v(HFCLK_Meas_Req)
v(Strobe_in_CLK_CR)
v(DCO_HFCLK)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
50
100
150
200
250
300
350
400
450
500
550
600

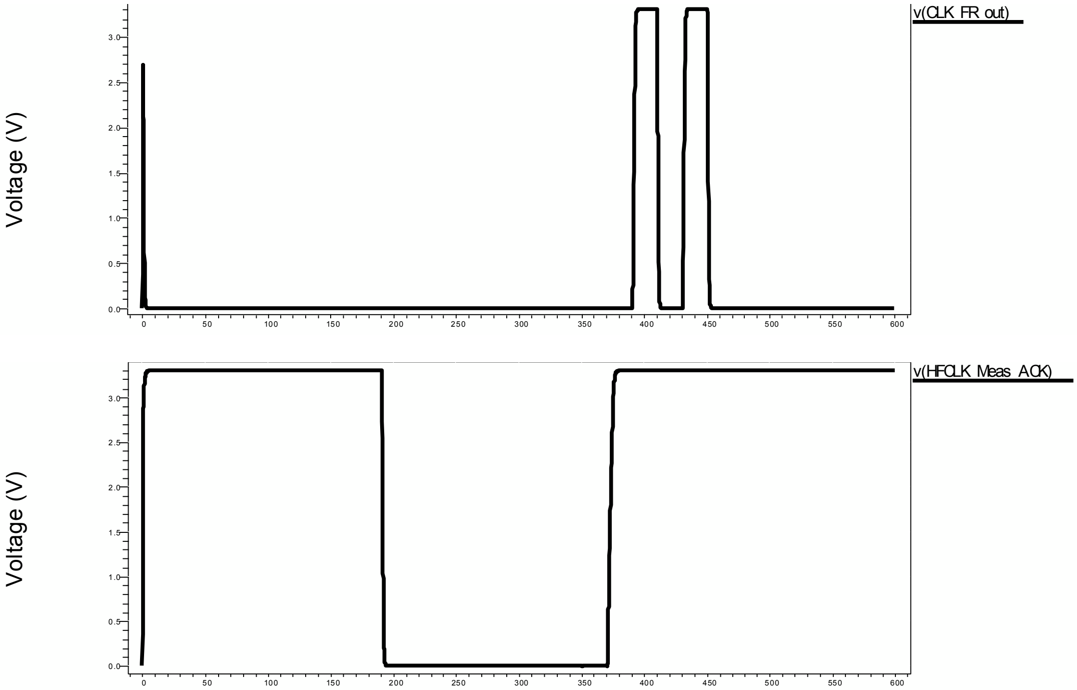


Figure 39 CCG signals (minimum frequency); pre-layout simulation

Measurement output is 10100, in binary, for five TACP clocks which equals 19 DCO cycles (190MHz). The actual frequency, as measured from spice simulation, is 193MHz and the accuracy could be increased by using more TACP clock cycles (i.e. increasing measurement period).

5.1.2 Post-Layout simulation of the chip

Figure 40 is the core of the layout, Figure 41 is the chip of the design used to simulate the circuit and Table 5 summarizes the area and nets of the core. It seems that the chips received have a problem in gates that have one or more input connected directly to the power pins which means that any input that is connected to the power pin will be floated. The testing was not done successfully and further investigation will be prepared to see what exactly the problem is.

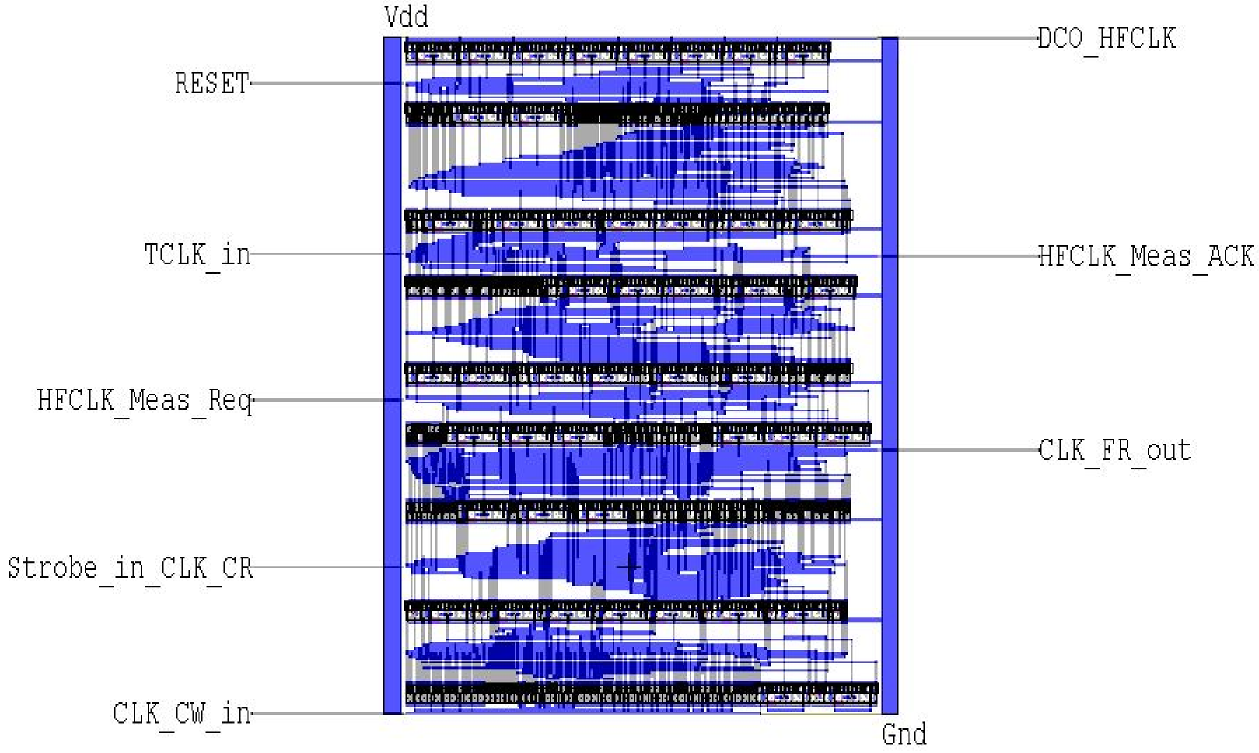


Figure 40 Clock generator's core layout

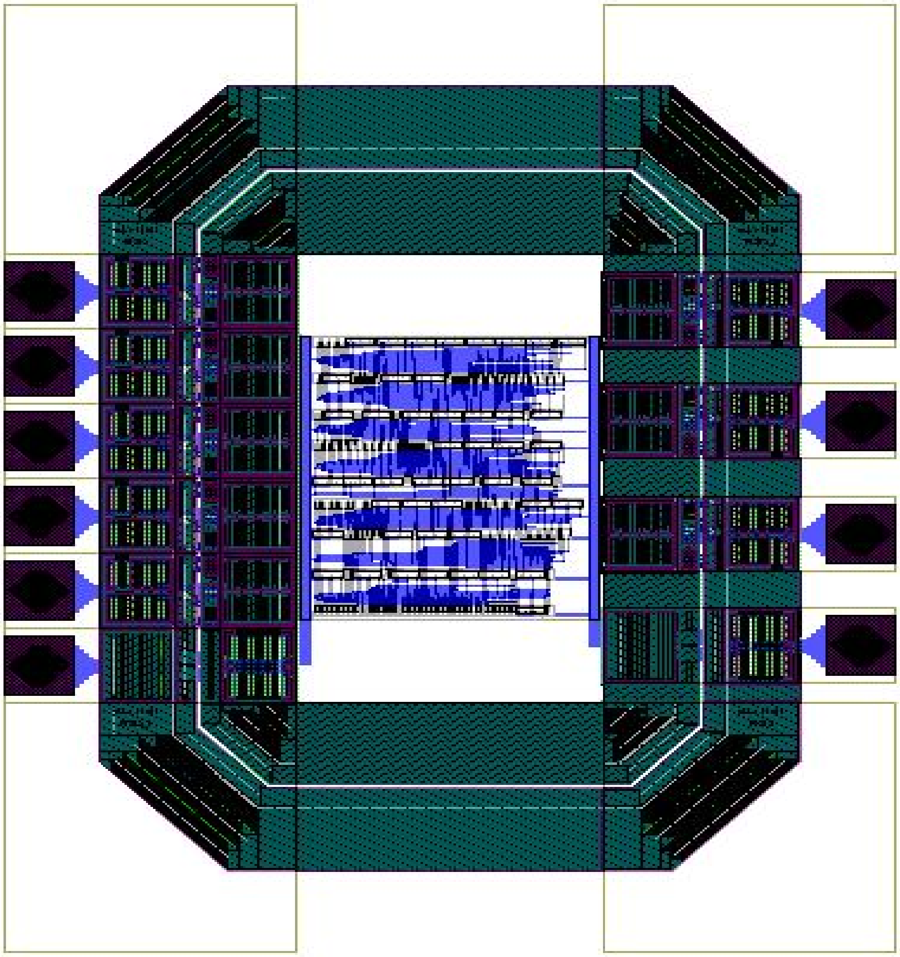


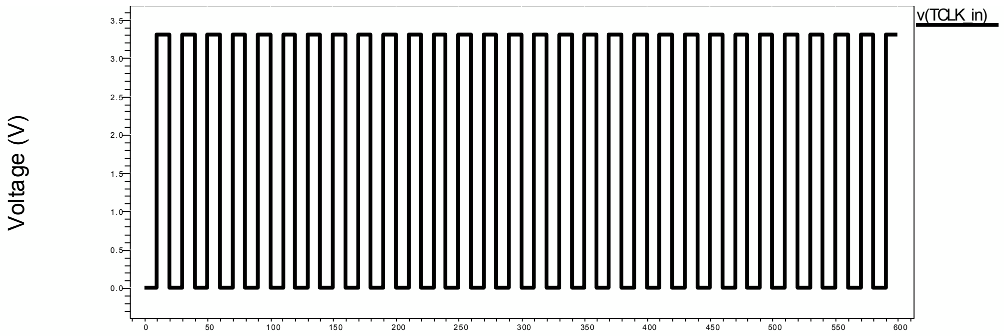
Figure 41 Final tested chip of the clock generator

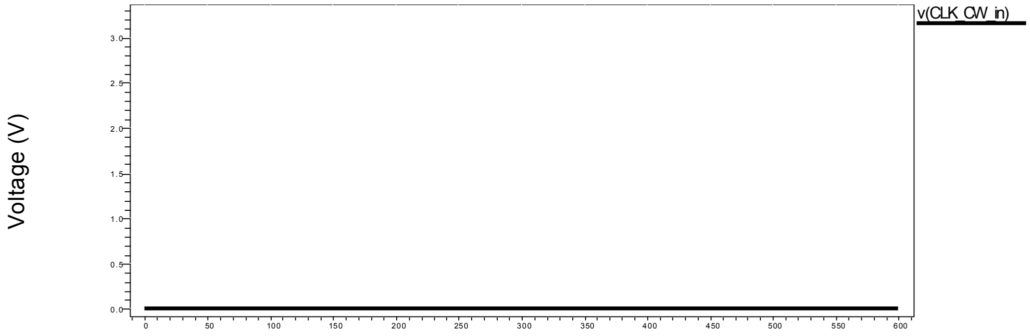
Table 5 Summary of the clock generator chip

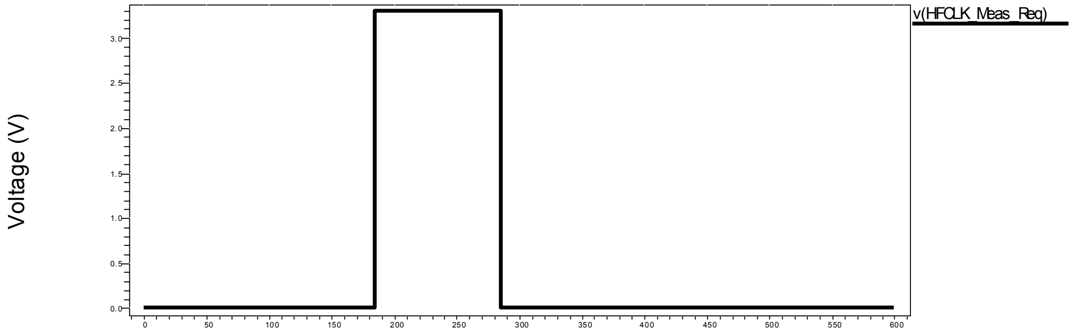
Number of standard cells	216
Number of signals in netlist	231
Core size in Microns	304.9 x 343.3
Core area (Microns^2)	104672.17
Frame size in Microns	916.4 x 1140
Frame area (Microns^2)	1044696
Length of nets in core	41803.5 Microns
Generated vias in core	922

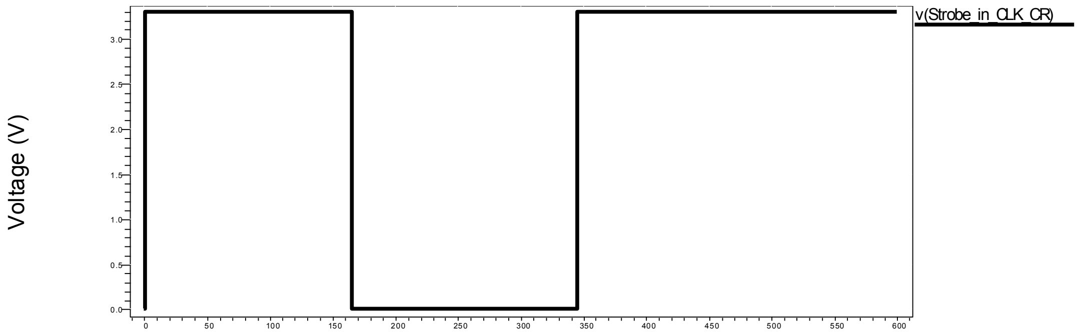
A spice post-layout simulation was done for the chip including the PADs to make sure that parasitic capacitances are not negatively affecting the design. The maximum frequency obtained was 462.38MHz and the minimum frequency was 221.396MHz which is about half the maximum frequency. Figure 42 illustrate the inputs and outputs signals.

With maximum frequency, the measurement output is 101110 for five TACP clocks which equal to 46 DCO cycles (460MHZ). The Actual frequency as measured from spice simulation is 463MHZ.


Voltage (V)
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(TCLK_in)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(CLK_CW_in)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(HFCLK_Meas_Req)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(Strobe_in_CLK_CR)

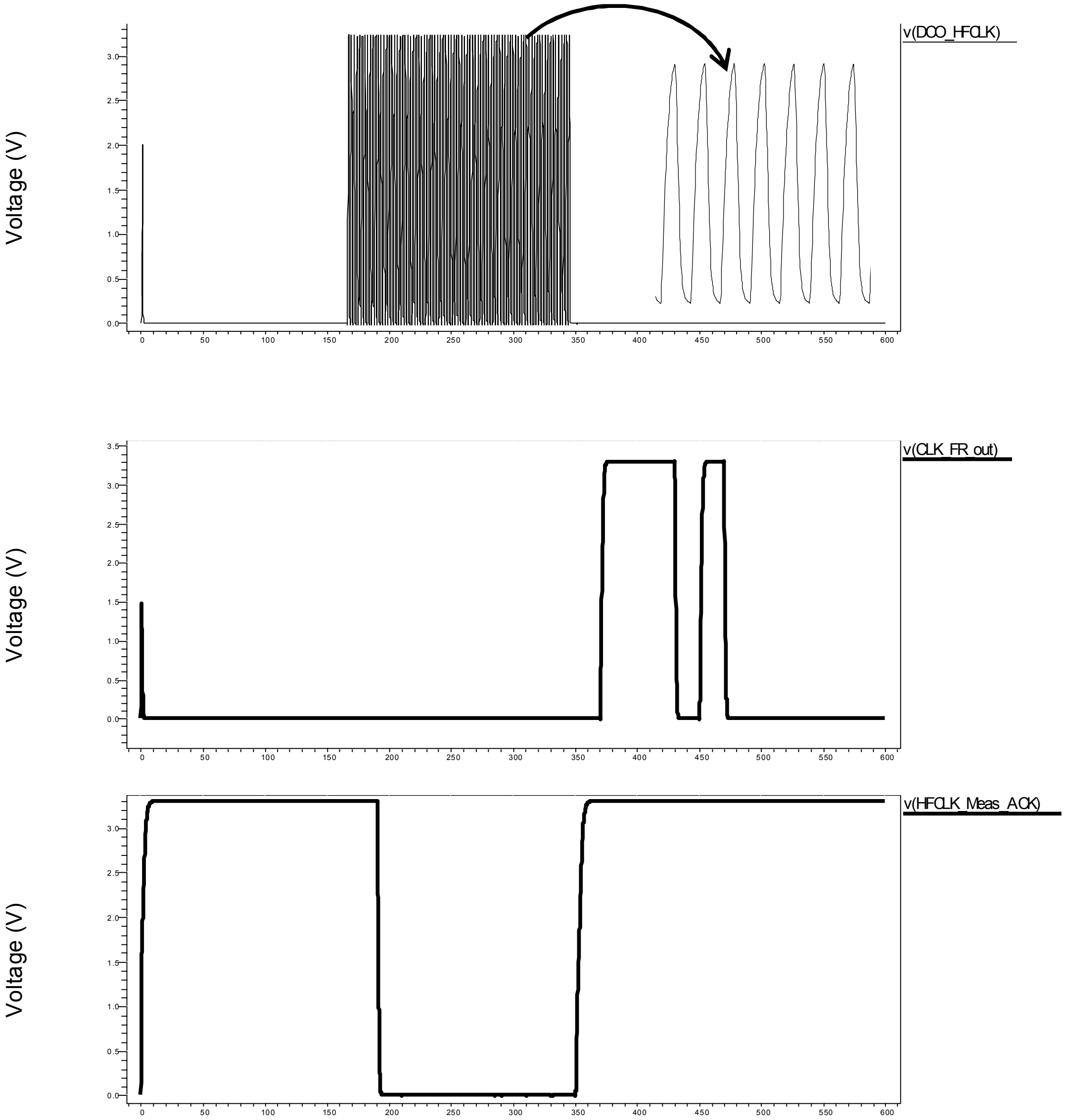
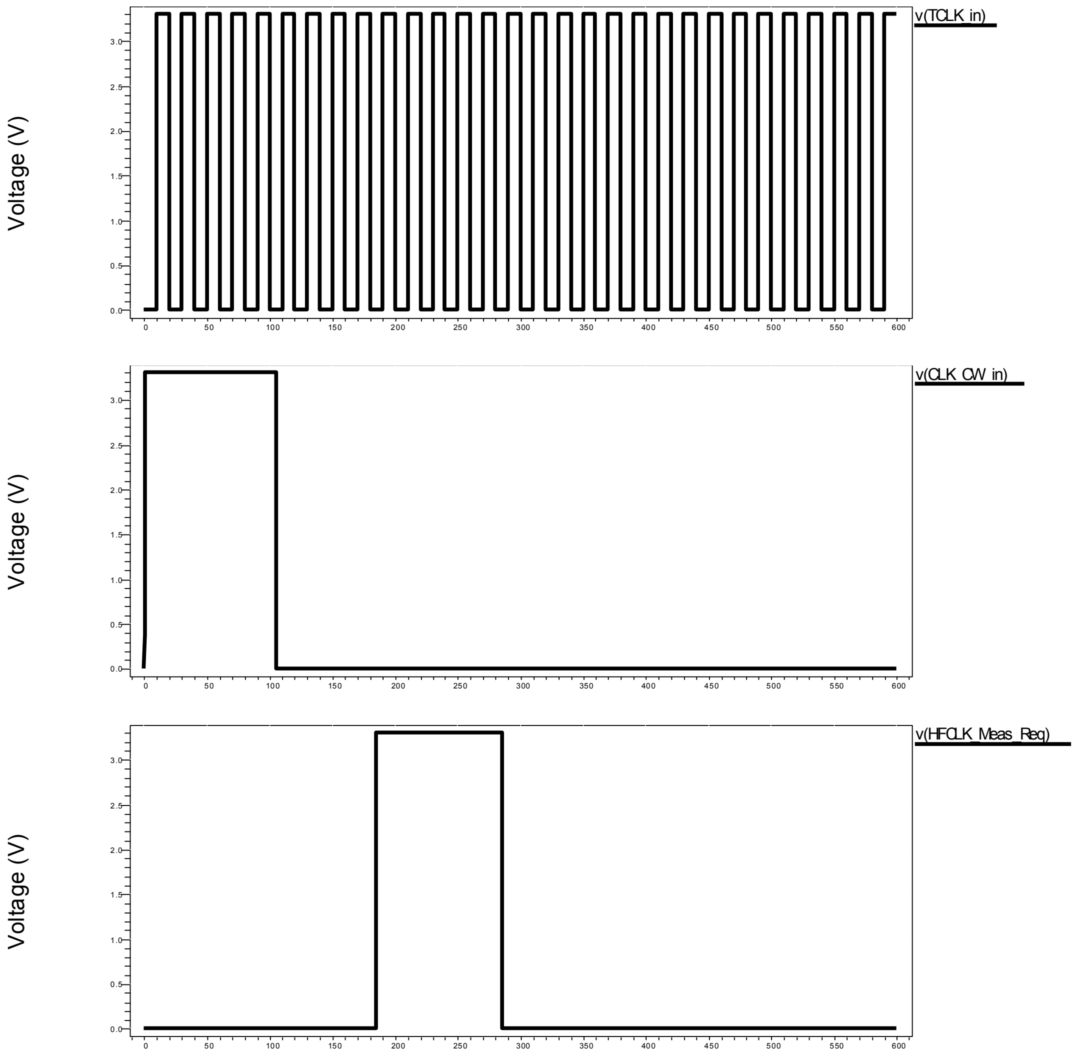


Figure 42 CCG signals (maximum frequency); post-layout simulation

On the other hand, measurement output for minimum frequency is 10101 for five TACP clocks which equal to 21 DCO cycles (210MHZ), Figure 43. The actual frequency as measured from spice simulation is 219MHZ. dividing the maximum frequency by two gives 230MHz which is greater than the minimum frequency. This means that the $2T_{Dmin}$

$\leq T_{Dmax}$ was not violated. Post-layout Simulations using DCO frequency division are also reported in Appendix B.



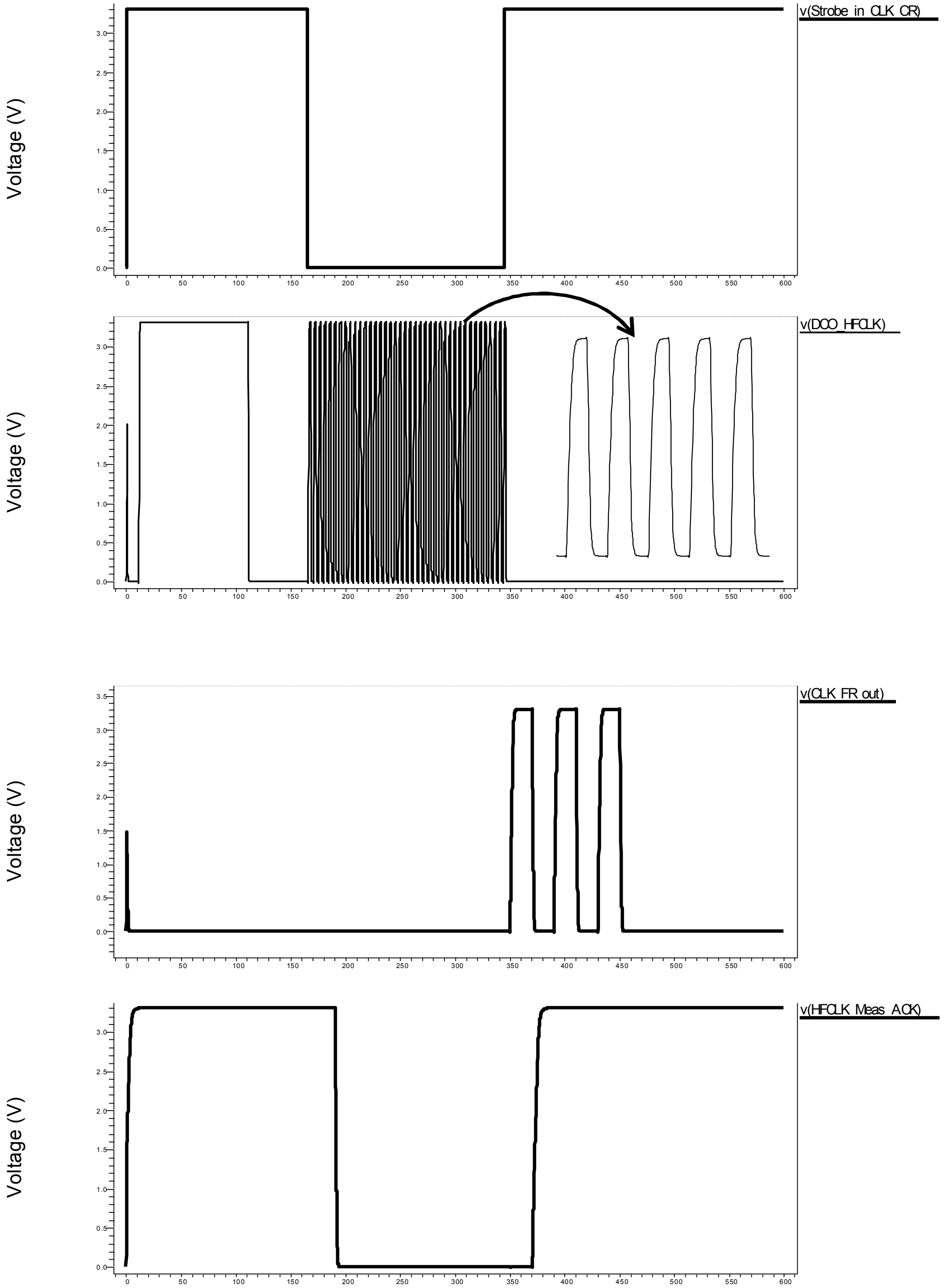


Figure 43 CCG signals (minimum frequency); post-layout simulation

Figure 44 shows the core integrated with other designs of the fabricated chip. From this chip, only ten I/O PADs were allocated for the CCG and the difference between this chip and the one in Figure 38 is that the chip in Figure 41 was used to simulate the CCG only and this chip was the fabricated one.

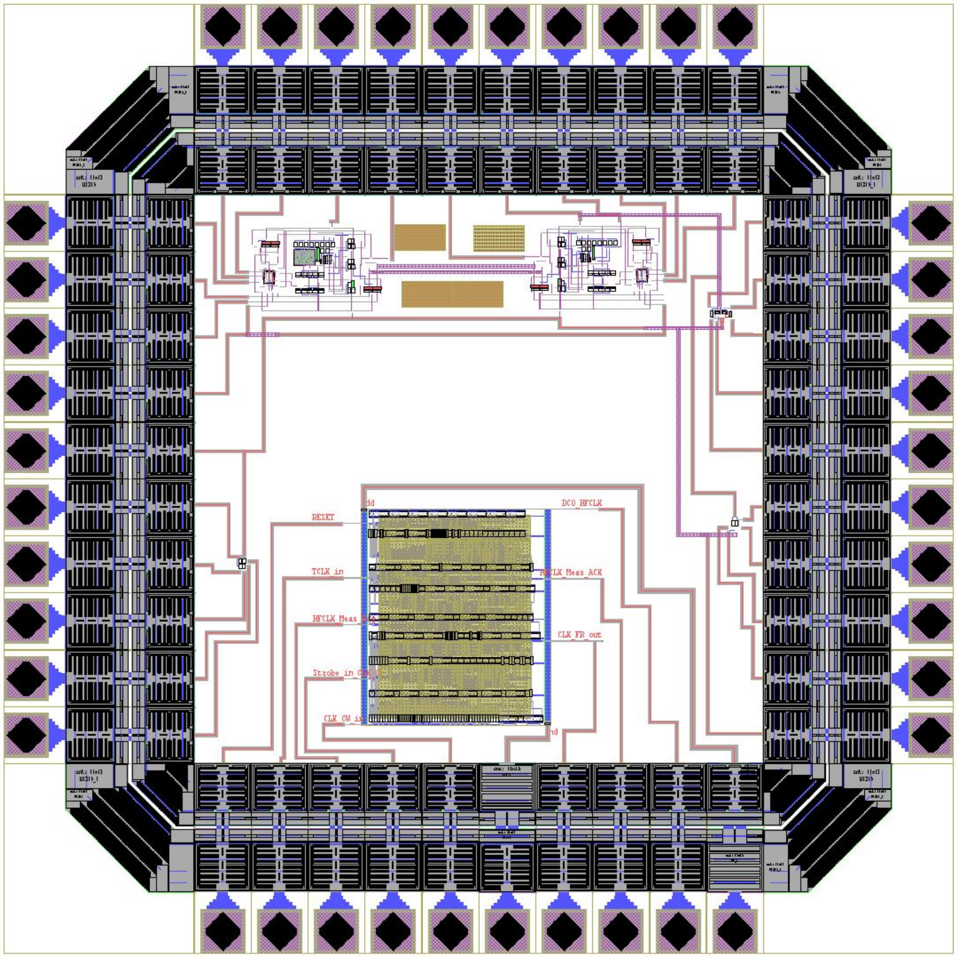


Figure 44 The fabricated chip; the core at the lower half of it

5.2 Implementation of the Complete Test Support Circuitry Using LFoundry 150nm Technology

To verify the operation of the CCG, a complete implementation of the TSC has been carried out using LF150nm technology. A Standard ASIC design methodology was used to implement the supported circuitry; the scripts that were used are reported in Appendix D. The DCO was implemented as custom cell and the two designs were integrated using Custom designer tool and simulated using HSPICE. The design flow, Figure 45, starts by reading and analyzing the Verilog netlist in the DC compiler. Then, constraints are specified for the design such as the clock that is to be used during synthesis and optimization. Thirdly, synthesis and optimization are performed to the design and the output from this step is the gate level netlist of the design and the constraint file that are used during place & route process. Place & route were done in IC compiler and the output from it is the netlist of the design, GDS file of the design used in Custom designer and parasitics file for post-layout simulation. The clock tree is optimized in IC compiler and reports are generated to check wither the required performance of the circuit is met or not. The design then is integrated with the DCO, which is implemented as mentioned before as a custom cell, in Custom designer to produce the complete core of the chip, Figure 46.

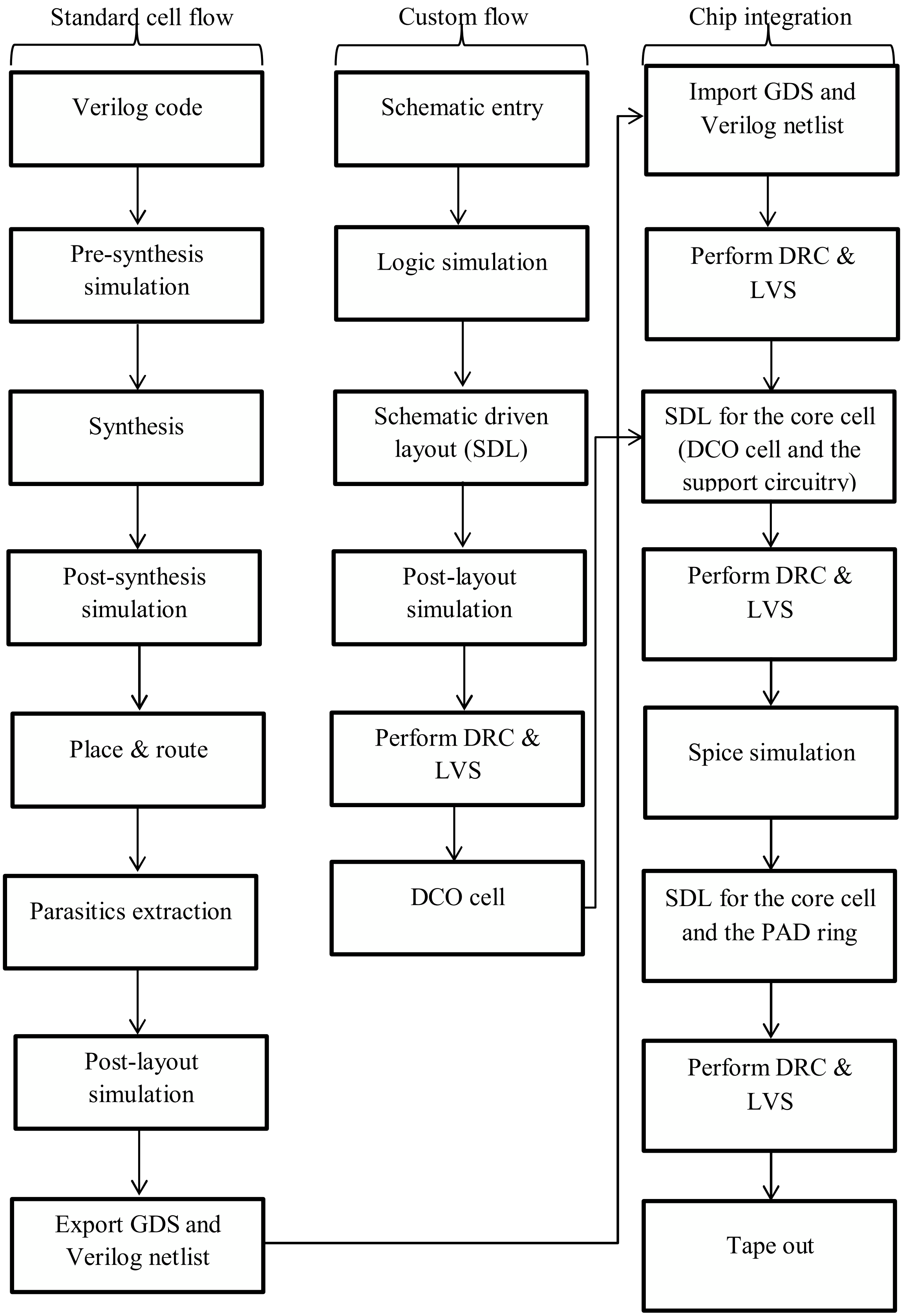


Figure 45 Design flow of the support circuitry

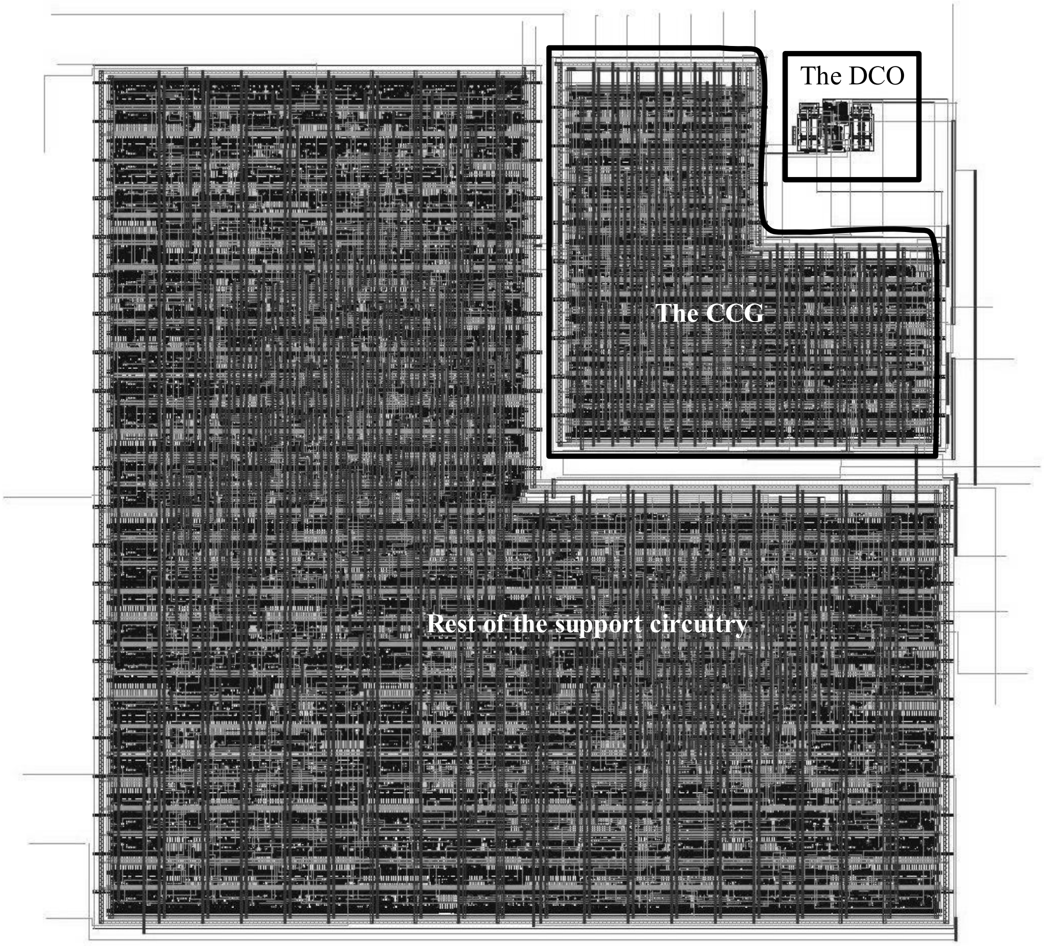
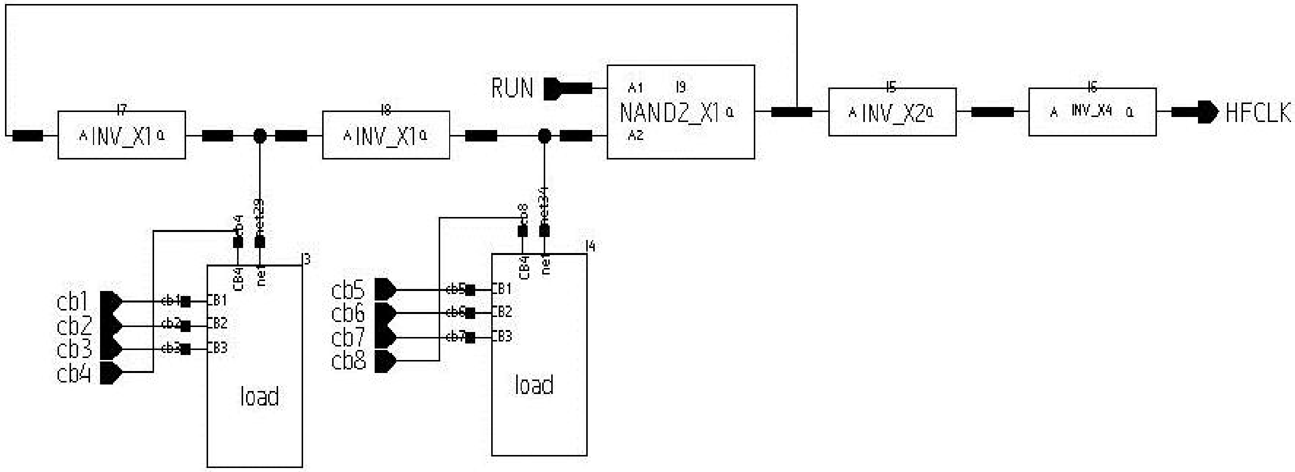


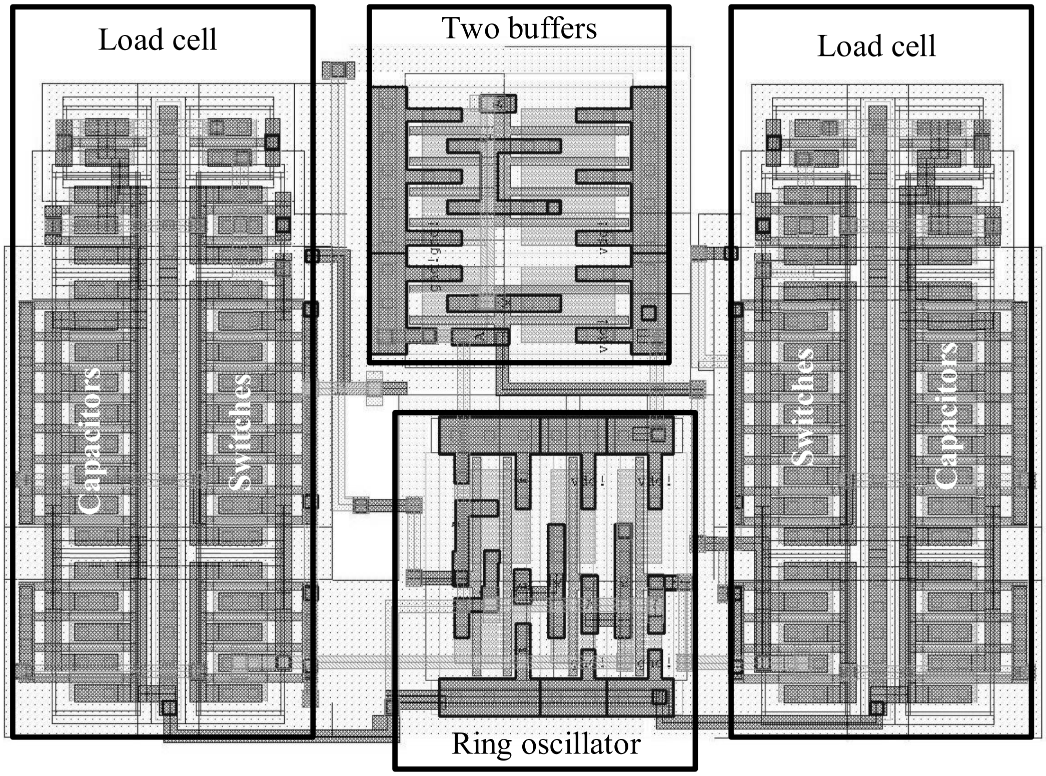
Figure 46 Layout of the support circuitry

There are four IUTs used in this fabricated support circuitry. Two of them are the S820s benchmark. In addition to the S820s IUTs, 4-bit adder was used and also 8-bit pipelined adder. The area dimensions of the support circuitry without the four IUTs are 107 X 107 micron while they are 203 X 203 micron which means that the support circuitry area is small compared to the IUTs it was used to test. The S820s IUTs flip-flops are all scan chain flip-flops which allow doing scan chain testing to them. Simulations were done and the data were scanned in and out to the S820s IUTs as in Appendix C using the test bench in Appendix E.

The DCO, Figure 47, consists of the basic ring oscillator i.e. two inverters and one two input NAND gate, two load cells which are basically eight NMOS switches and eight binary weighted NMOS capacitors, and two buffers connected to the output of the DCO. The final chip that sent for fabrication is shown in Figure 48.



(a)



(b)

Figure 47 (a) Schematic of the DCO using Synopsys tools; (b) Layout of the DCO

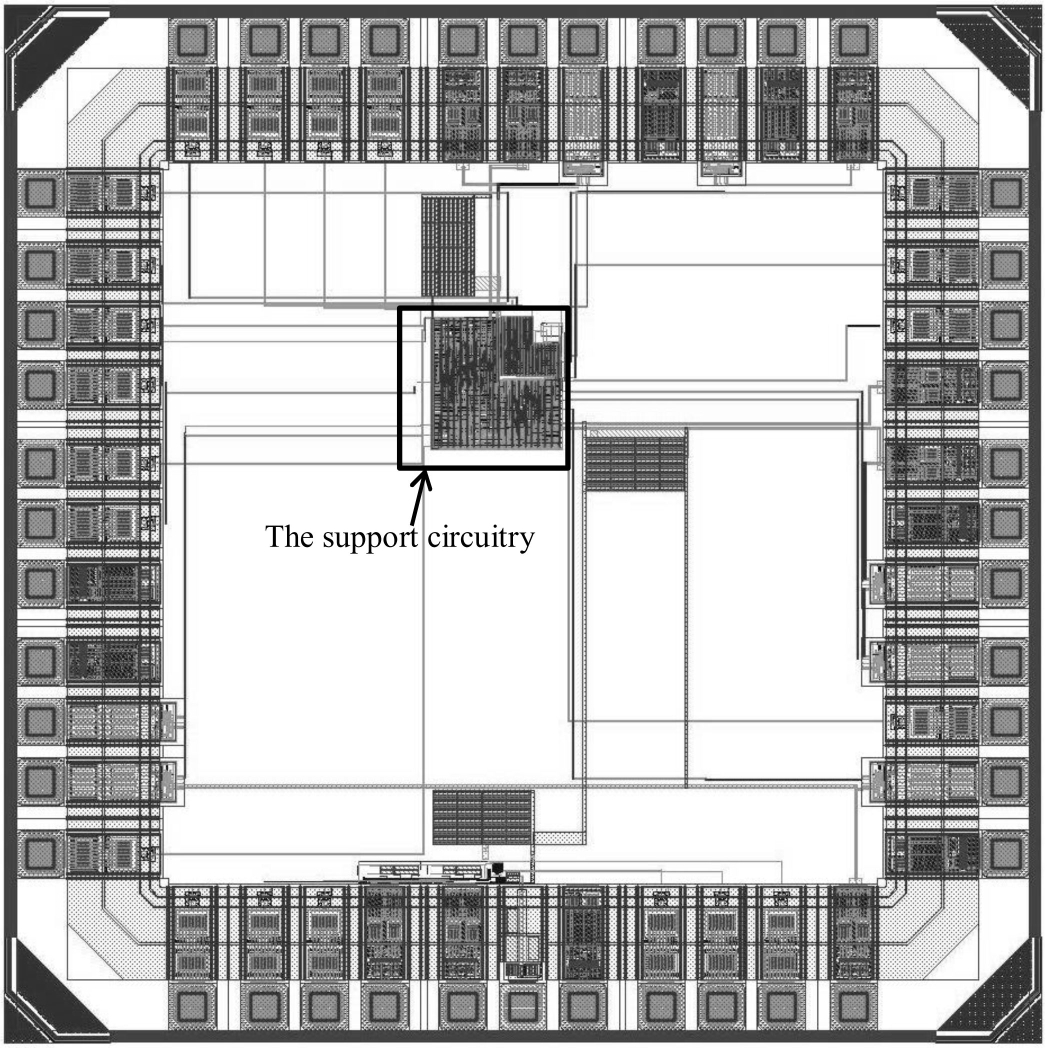
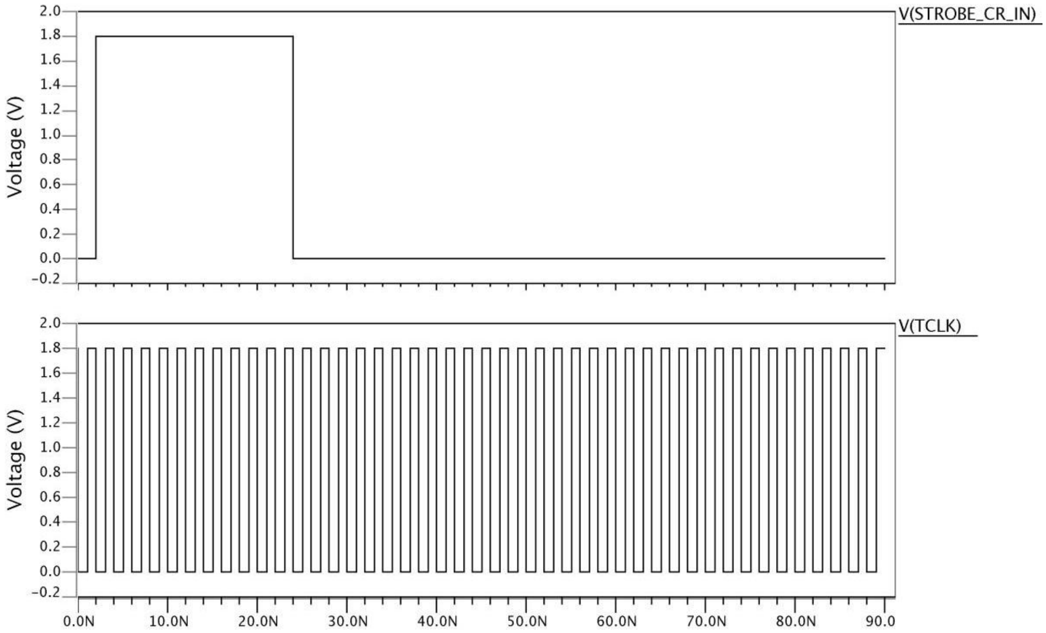


Figure 48 Final chip sent for fabrication

Spice simulation was done to the support circuitry after integrating all of its components and the 4-bit adder IUT was used in this simulation to verify the operation of the support circuitry. To illustrate the division of the DCOs high frequency clock, Figure 49 shows the result of dividing the output of the DCO (OSC-CLK) by four (HFCLK). The DCO period is almost 325ps, zoomed in Figure 49, and the result of division by four is about 1.2ns. To choose the frequency required for testing, which is four in this case, the control word input (CLK_CW_IN) should be. The control word input (CLK_CW_IN) is used for shifting the five control bits of the DCO and also three bits for the frequencies selections i.e. the high frequency clock, the high frequency clock divided by 2, the high frequency clock divided by 4, the high frequency clock divided by 8, and the high frequency clock divided by 16. 000 means divide by 2, 001 means divide by 4 (as in Figure 49), 010 means divide by 8, 011 means divide by 16, and 1xx means no division and the high frequency clock itself will be selected.



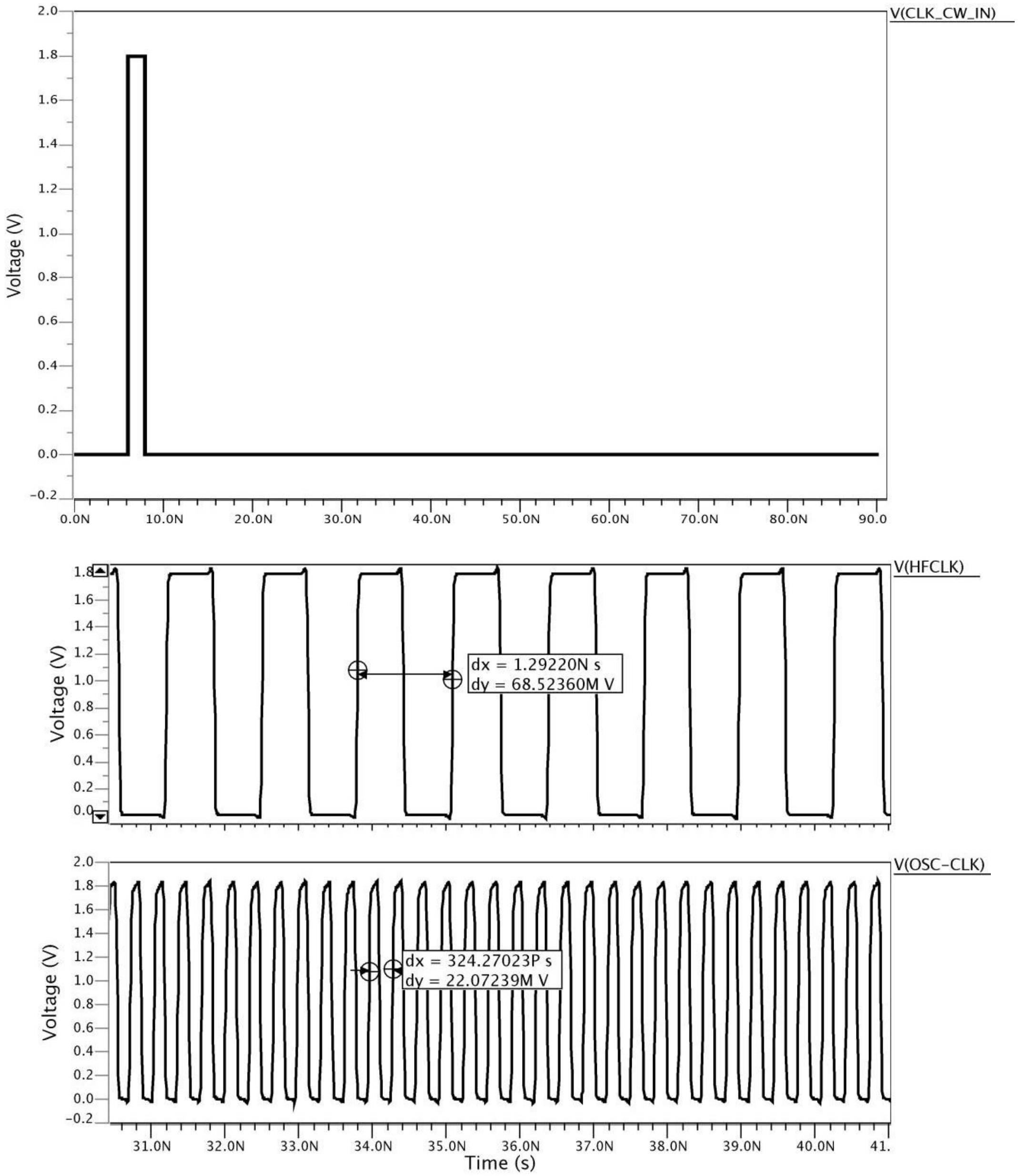


Figure 49 Result of dividing the DCO output by four

To verify the correct operation of the CCG, an example is given in Figure 50. The number of DCO cycles is 1101 (CLK_FR_OUT signal) for 15 DCO cycles (10 TCLK cycles) when measurement request is high (HFCLK_MEAS_REQ signal). Division of the high frequency clock by four was used in this example. The result of counting is shifted out once the acknowledgment signals goes high (HFCLK_MEAS_ACK signal) and the strobe signal is activated (STROBE_FR_OUT signal). The error in counting can be reduced significantly by increasing the measurement request period which increase number of counted DCO cycles according to the following measured frequency formula:

$$\text{Measured Frequency} = \frac{\text{No. of counted DCO cycles } \times \text{TCLK}}{\text{Measurment period}}$$

$$\text{Measurment period} = \frac{\text{No. of TCLK cycles}}{\text{TCLK frequency}}$$

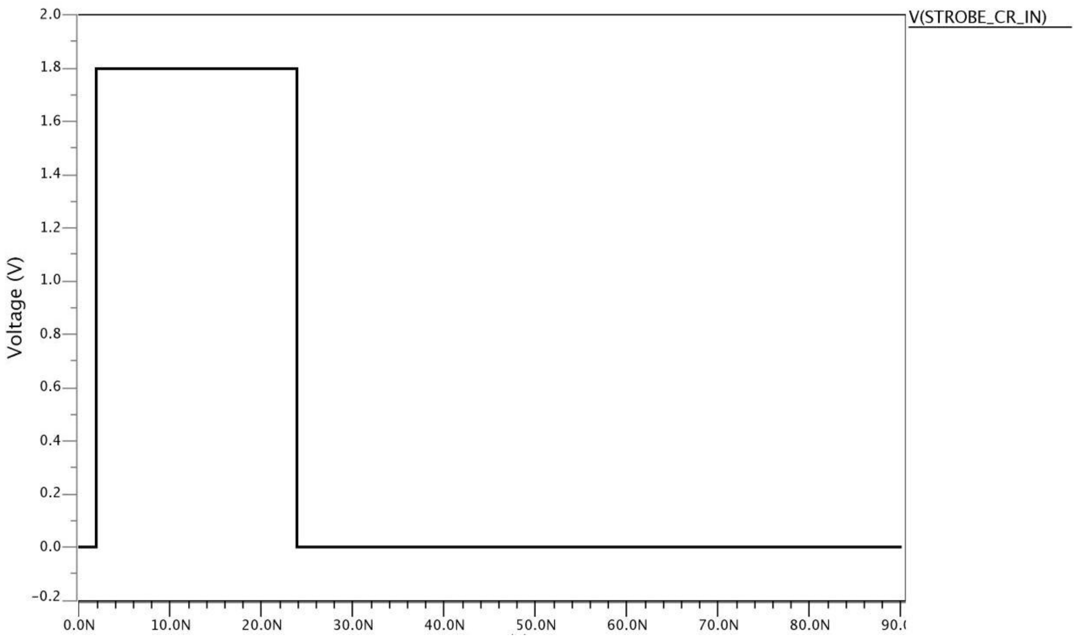
$$\text{Error} = \pm\frac{1}{\text{one DCO cycles}}$$

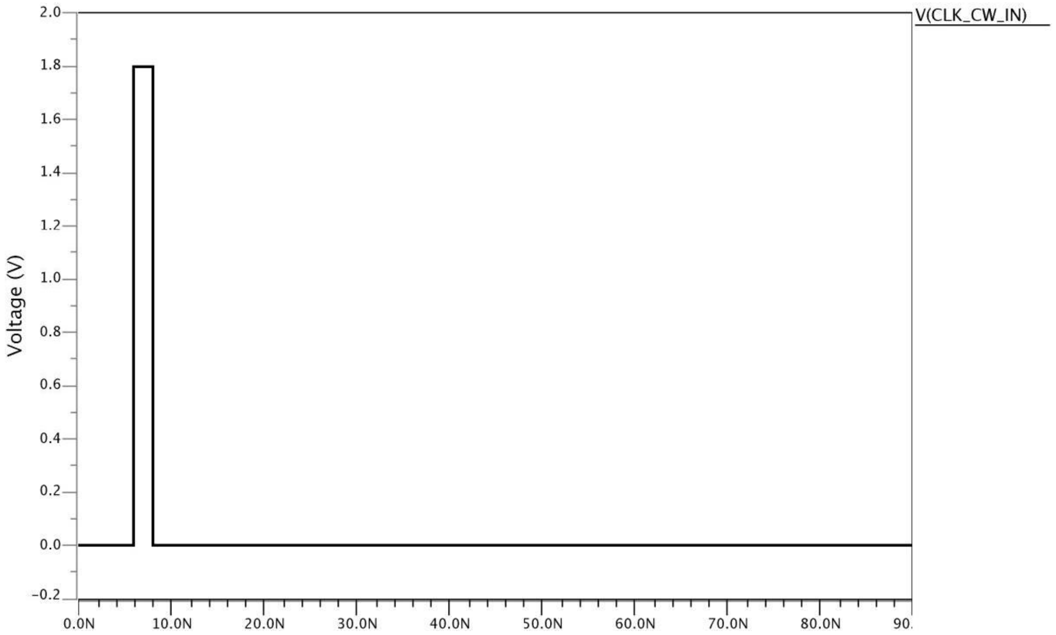
$$\text{Measurment period} = \frac{\text{No. of TCLK cycles}}{\text{TCLK frequency}} \pm \frac{1}{\text{one DCO cycles}}$$

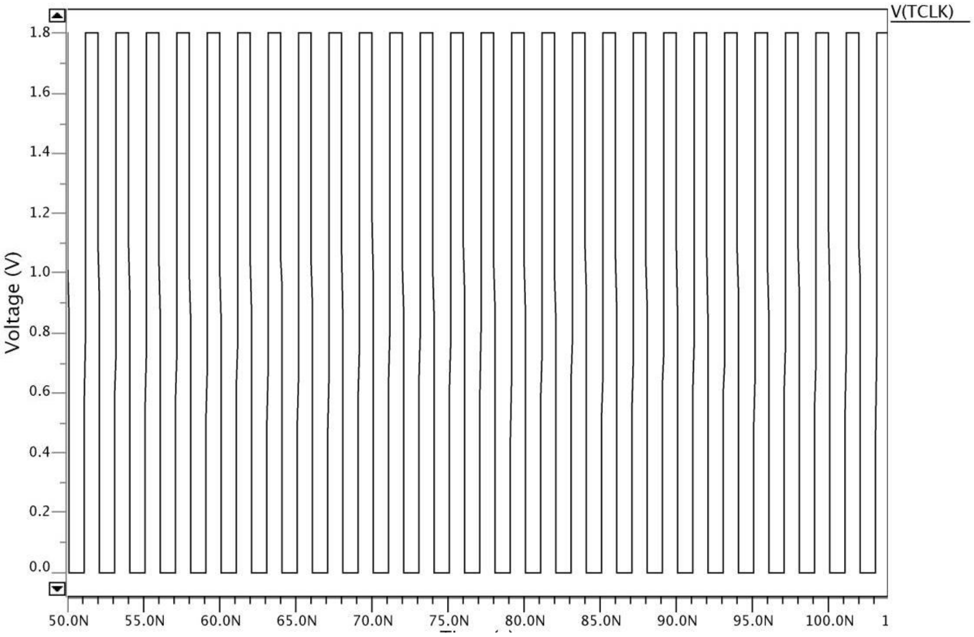
$$\text{Measurment period} = \frac{\text{No. of TCLK cycles} * \text{one DCO cycles } \pm \text{ TCLK frequency}}{\text{TCLK frequency} * \text{one DCO cycles}}$$

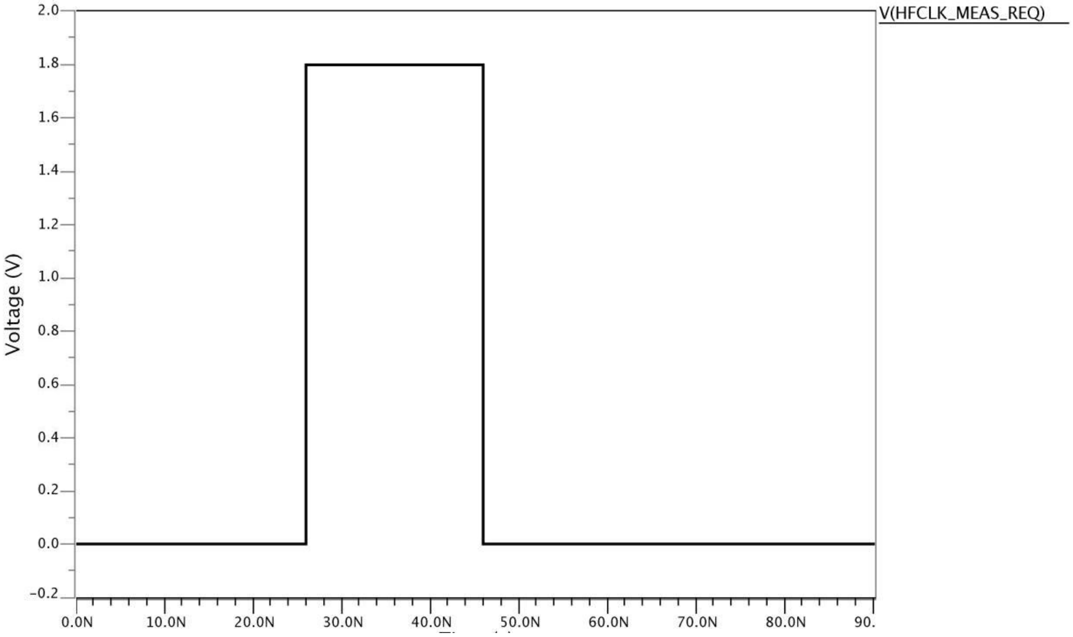
$$\text{Measurment period} = \frac{\text{No. of counted DCO cycles} * \text{TCLK frequency} * \text{one DCO cycles}}{\text{No. of TCLK cycles} * \text{one DCO cycles } \pm \text{ TCLK frequency}}$$

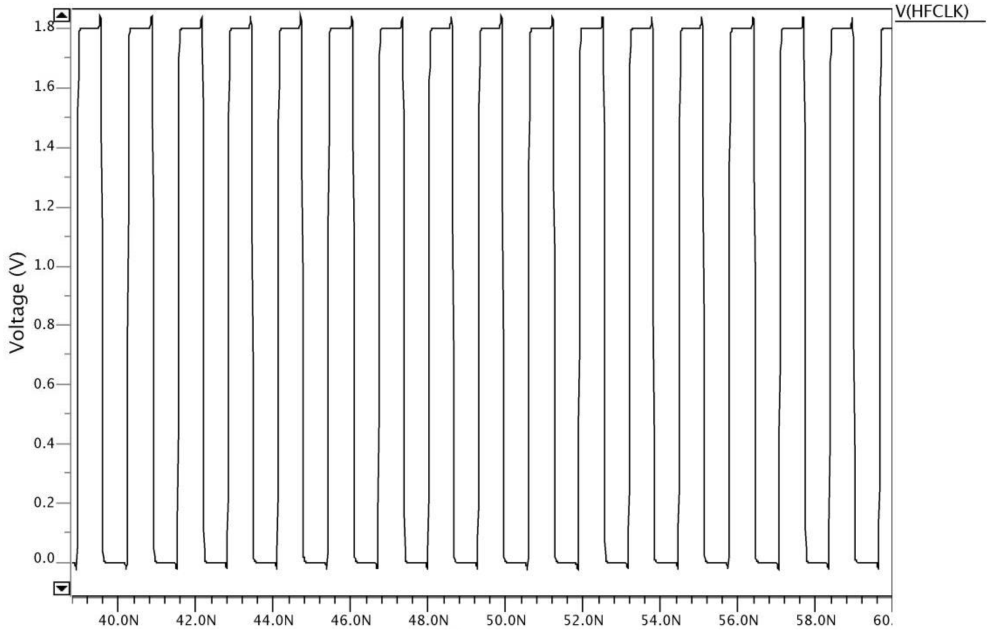
In our spice example, the error was about 13% and if number of cycles is increased to 1000, for instance, the error will be about 1%. This, however, will take a very long simulation time.

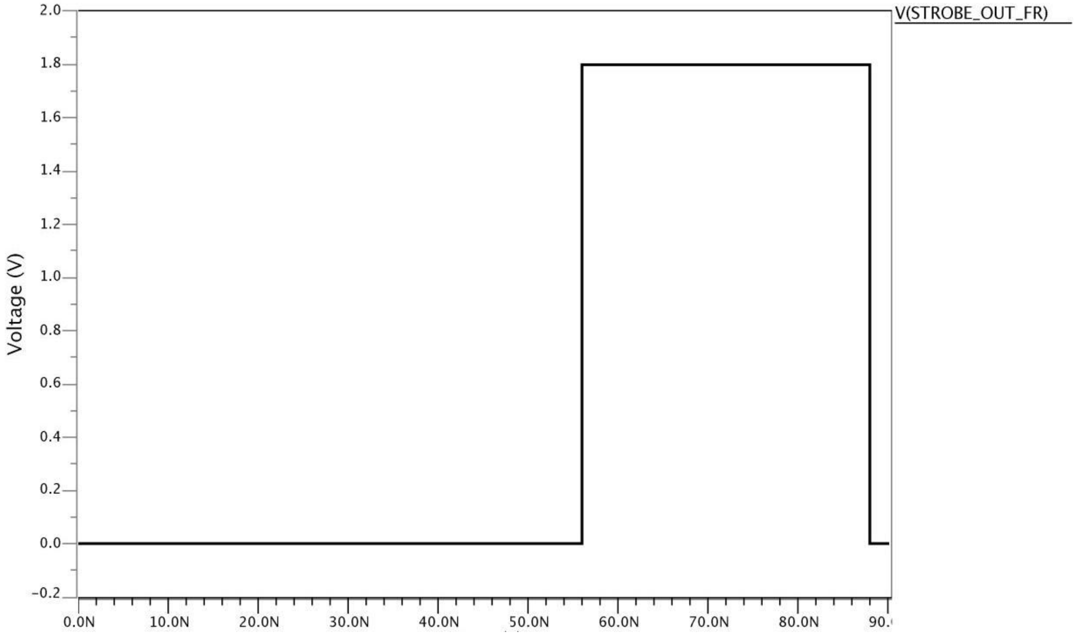

V(STROBE_CR_IN)
Voltage (V)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0N
10.0N
20.0N
30.0N
40.0N
50.0N
60.0N
70.0N
80.0N
90.


V(CLK_CW_IN)
Voltage (V)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0N
10.0N
20.0N
30.0N
40.0N
50.0N
60.0N
70.0N
80.0N
90.


V(TCLK)
Voltage (V)
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
50.0N
55.0N
60.0N
65.0N
70.0N
75.0N
80.0N
85.0N
90.0N
95.0N
100.0N
1


V(HFCLK_MEAS_REQ)
Voltage (V)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0.0N
10.0N
20.0N
30.0N
40.0N
50.0N
60.0N
70.0N
80.0N
90.


V(HFCLK)
Voltage (V)
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
40.0N
42.0N
44.0N
46.0N
48.0N
50.0N
52.0N
54.0N
56.0N
58.0N
60.


V(STROBE_OUT_FR)
Voltage (V)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0N
10.0N
20.0N
30.0N
40.0N
50.0N
60.0N
70.0N
80.0N
90.

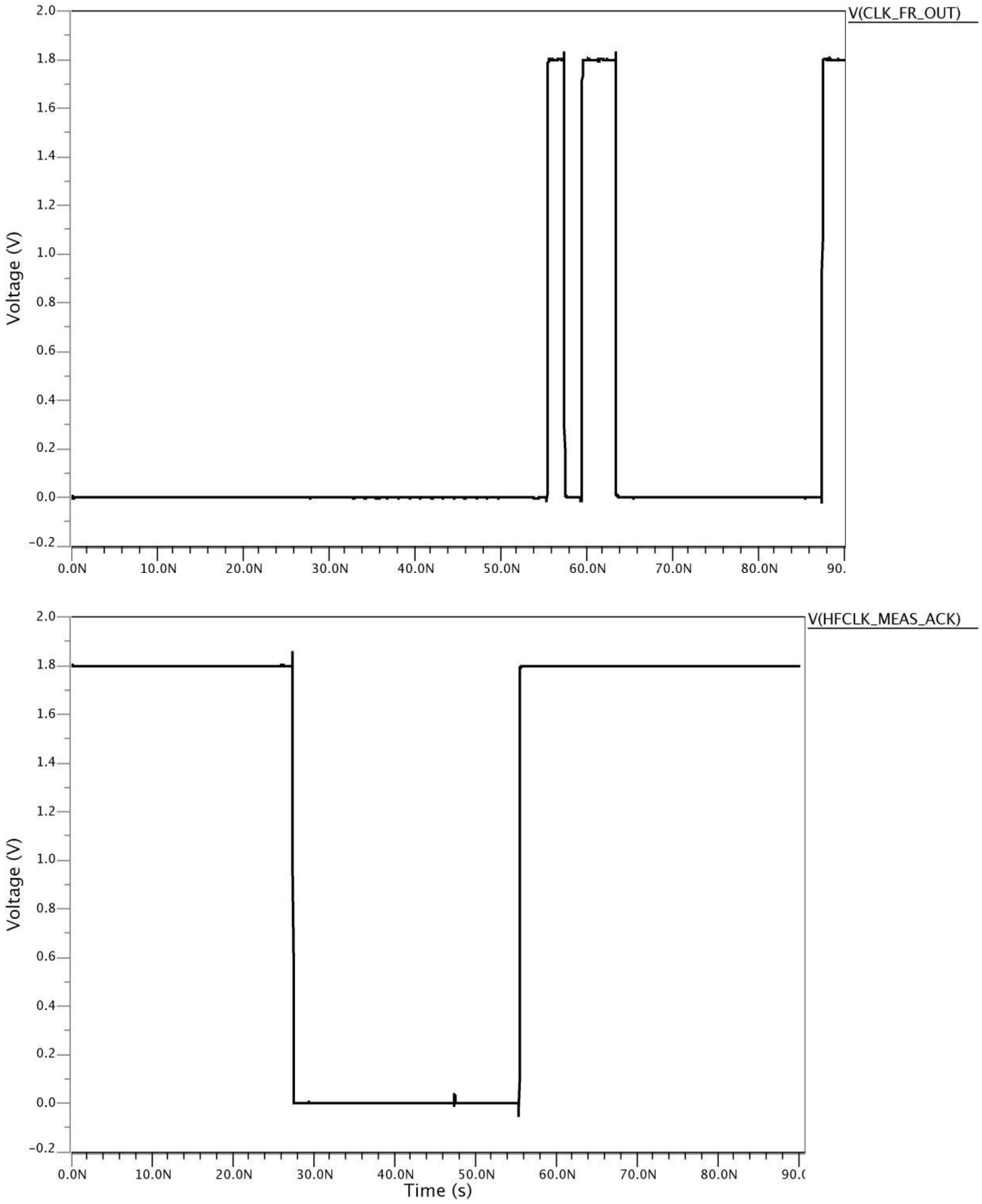
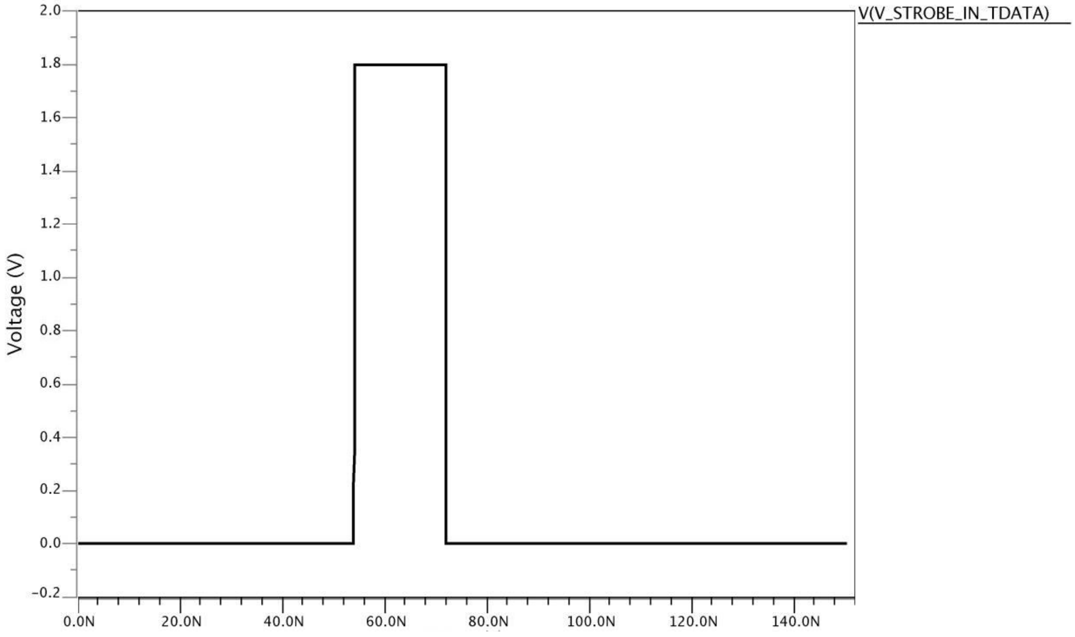
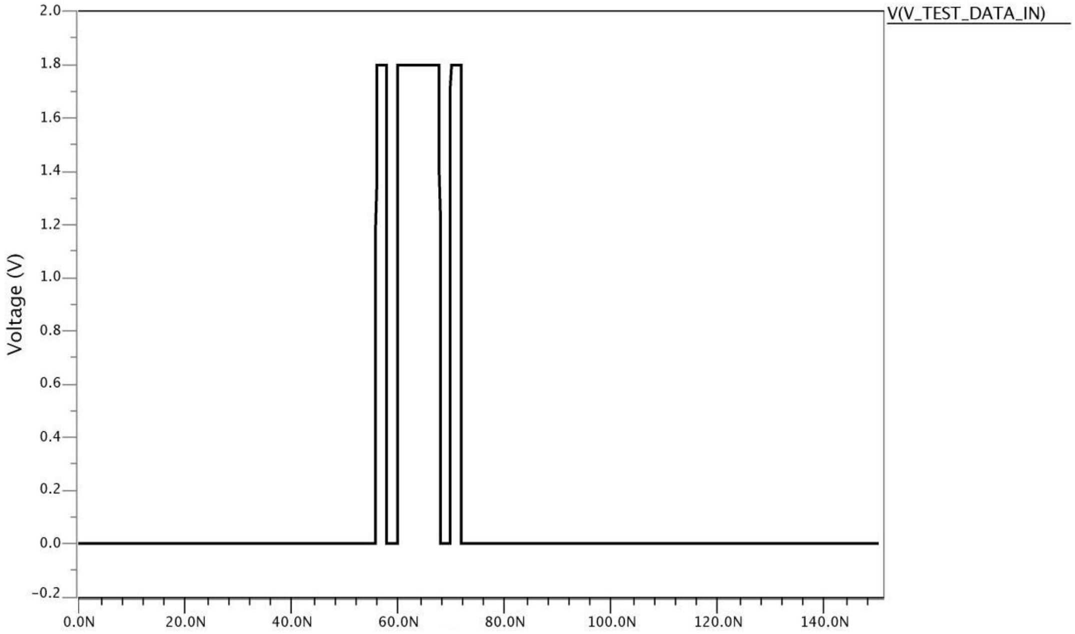


Figure 50 CCG spice simulation

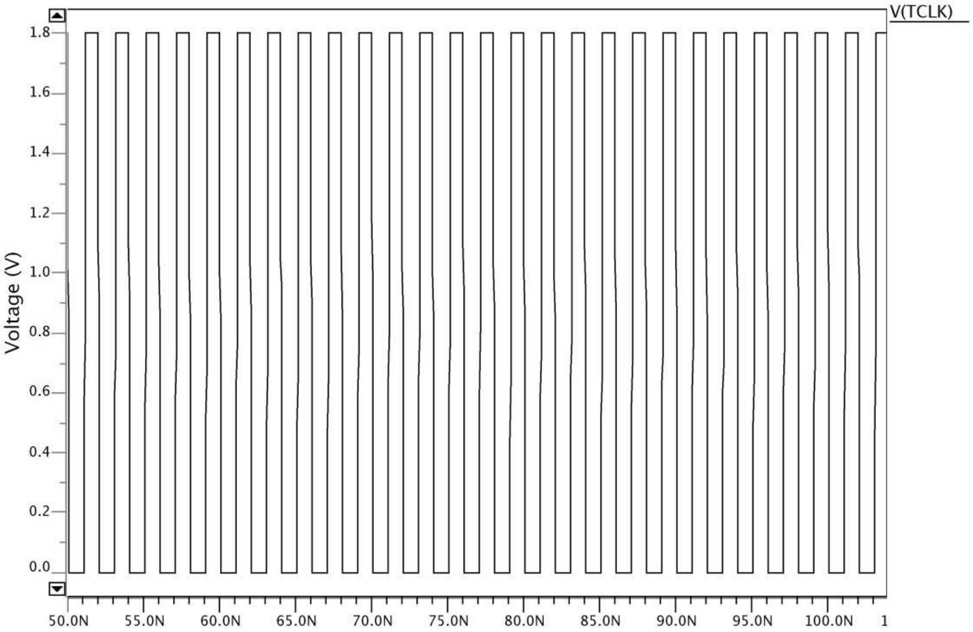
The rest of support circuitry was also simulated and the 4-bit adder IUT was used to verify the support circuitry operation, Figure 51. The input test data (V_TEST_DATA_IN signal) is shifted serially when its strobe (V_STROBE_IN_TDATA signal) is activated. Similarly, the results data

(TRESULT_OUT signal) is shifted out serially when its strobe is high (V_STROBE_OUT_TR signal). In the example of Figure 51, the input is the two numbers to be added (1010+0111) and the carry in which is one in this example and the output as expected is 10010 (18 in decimal). Figure 52 shows the two pulses produced when the AaC signal is activated from the selected clock.





V(V_TEST_DATA_IN)
Voltage (V)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0N
20.0N
40.0N
60.0N
80.0N
100.0N
120.0N
140.0N



V(TCLK)
Voltage (V)
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
50.0N
55.0N
60.0N
65.0N
70.0N
75.0N
80.0N
85.0N
90.0N
95.0N
100.0N
1

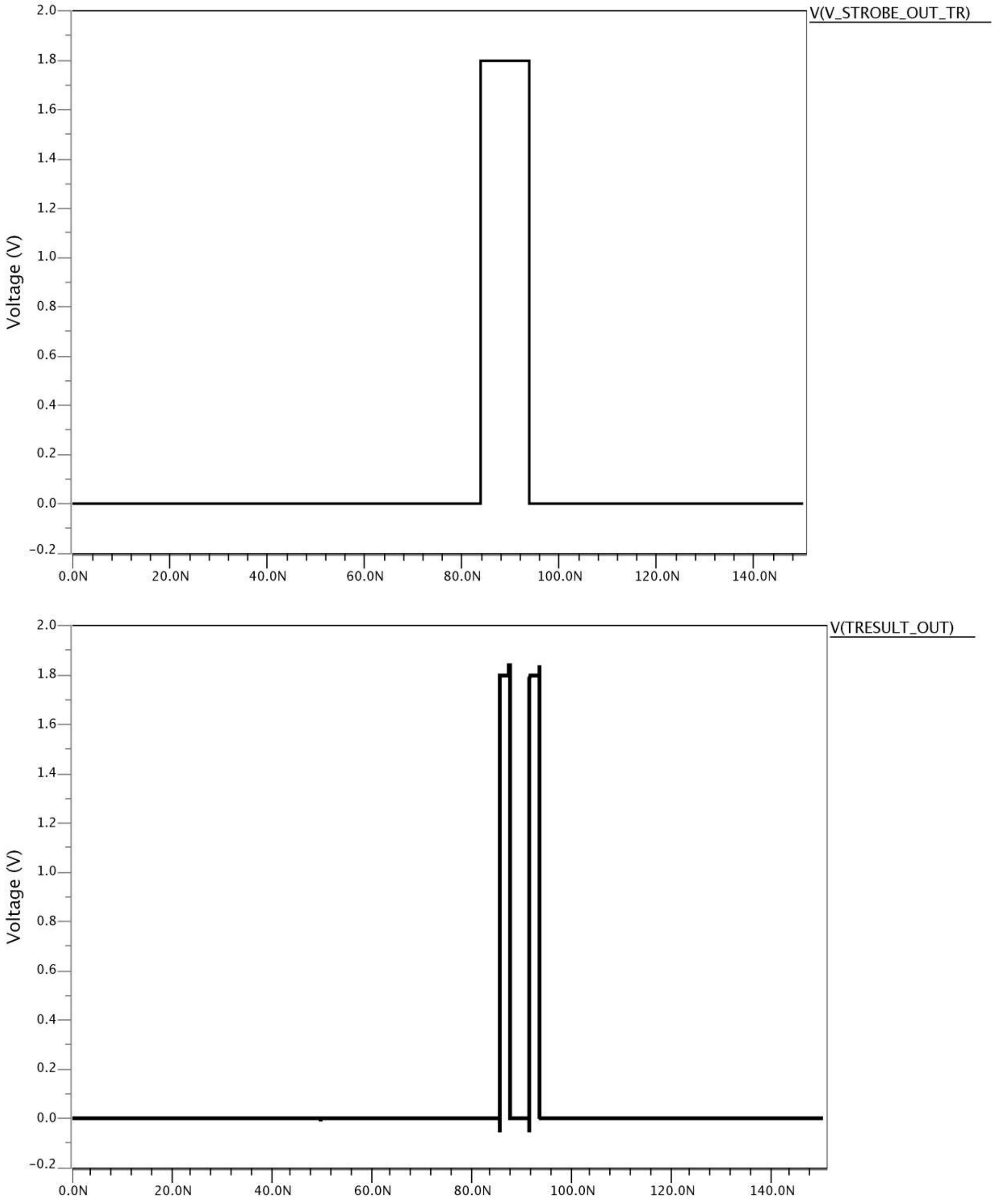


Figure 51 Support circuitry spice simulation; 4-bit adder is used as an IUT

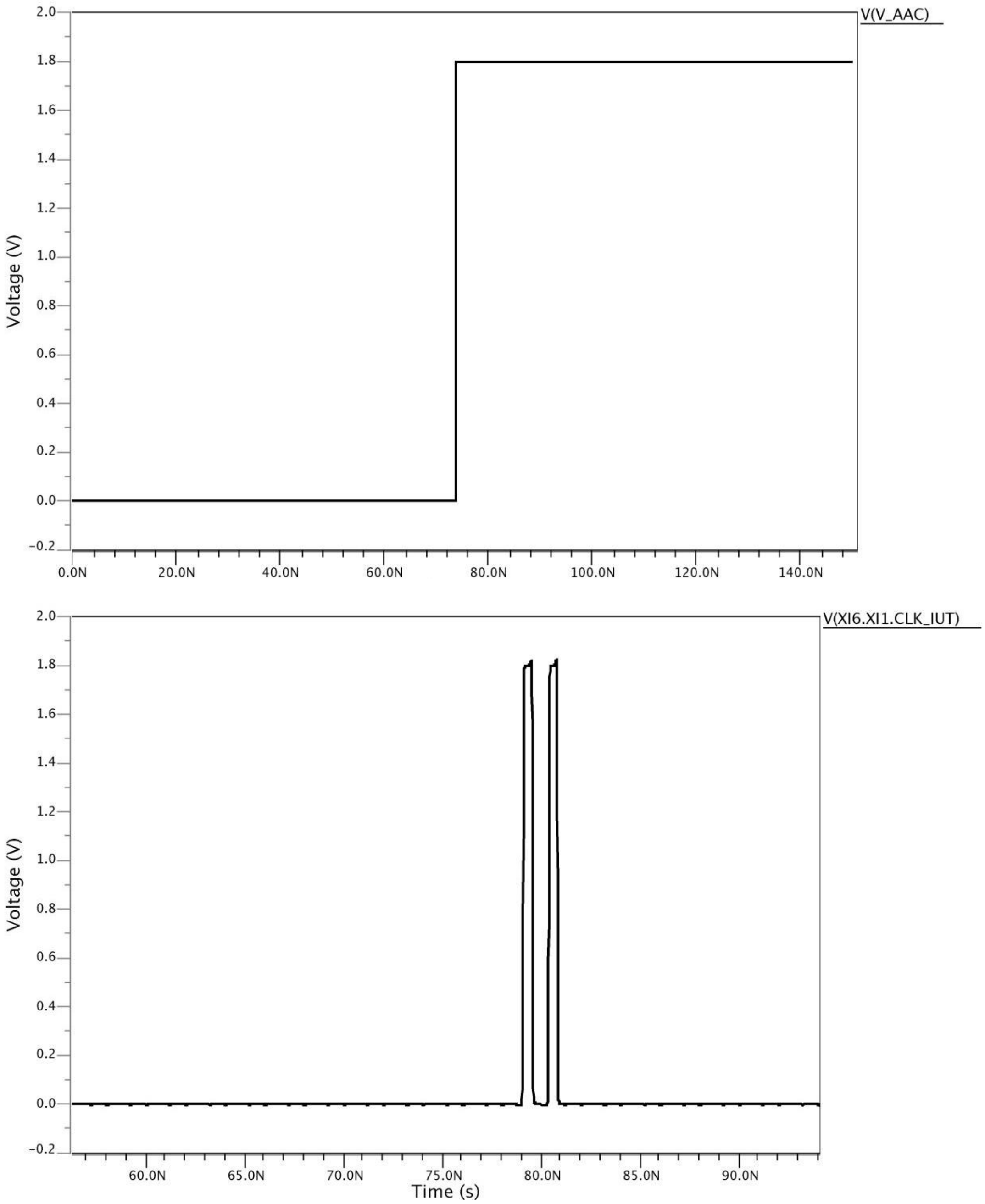


Figure 52 Two pulses of the selected clock produce when the AaC signal is high

The results shown in this chapter is strong evidence of the correct operation of the support circuitry that was designed and fabricated along with the CCG. The simulations that were done to verify the CCG that was fabricated in TSMC 0.35U technology indicate that the CCG is fully working with no errors. However, the chip needs further testing and troubleshooting to identify what are the causes that made the chip not working. The simulations that were also done to the whole support circuitry clearly indicate the correctness of the chip that was sent for fabrication using LF150n technology. To avoid any problems that may be caused by the DCO, an input signal was made so that external high frequency clock can be used as a high frequency clock for at speed-testing.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this work, a configurable all-digital clock generator, for speed characterization of ASICs' purposes, was developed, simulated, and successfully taped-out using TSMC 0.35U technology. The used DCO was sufficiently analyzed and designed to give linear monotonic steps and can be used in any process to be used in the test and characterization platform. The analysis of the DCO gives a very close estimation to the capacitance required to do at-speed testing using any CMOS technology and it should help in further utilizing the support circuitry.

The configurable clock generator was integrated with a complete support circuitry for IUTs speed characterization and functional testing. The support circuitry was designed and sent for fabricated using LFoundry 150nm technology. The sent support circuitry included four IUTs to be tested and to verify the operation of the support circuitry and its configurable clock generator. The support circuitry was successfully simulated and it will be connected with the TACP to form the complete test and characterization platform upon receiving it.

6.2 Future Work

The platform and the clock generator could be implemented using many other technologies as evidence for their portability and to further prove the concept of operation of them. In addition, different number of IUTs with different functions should be implemented with these technologies. Since the platform is new, a standard cell based oscillator might be a choice although the frequency will be decreased. This will allow the user to easily integrate the support circuitry circuit with his/her IPs without concerning about design issues of the oscillator. Finally, the chip that was sent for fabrication would be tested upon receiving it and would be integrated with the TACP to send and receive the data from the support circuitry to test it four IUTs.

References

[1] Andreani, P.; Bigongiari, F.; Roncella, R.; Saletti, R.; Terreni, P.; , " A Digitally Controlled Shunt Capacitor CMOS Delay Line," *Analog Integrated Circuits and Signal Processing* , vol.18, no.1, pp. 89- 96, Jan. 1999

[2] Olsson, T.; Nilsson, P.; Meincke, T.; Hemam, A.; Torkelson, M.; , "A digitally controlled low-power clock multiplier for globally asynchronous locally synchronous designs," *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on* , vol.3, no., pp.13-16 vol.3, 2000

[3] Olsson, T.; Torkelsson, M.; Nilsson, P.; Hemani, A.; Meincke, T.; , "A digitally controlled on-chip clock multiplier for globally asynchronous locally synchronous systems," *Circuits and Systems, 1999. 42nd Midwest Symposium on* , vol.1, no., pp.84-87 vol. 1, 1999

[4] Maymandi-Nejad, M.; Sachdev, M.; , "A digitally programmable delay element: design and analysis," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.11, no.5, pp.871-878, Oct. 2003

[5] Baronti, F.; Lunardini, D.; Roncella, R.; Saletti, R.; , "A self-calibrating delay-locked delay line with shunt-capacitor circuit scheme," *Solid-State Circuits, IEEE Journal of*, vol.39, no.2, pp. 384- 387, Feb. 2004

[6] Jovanović, G.; Mile, K.; Stojčev, T.; , "Linear Current Starved Delay Element," *ICEST 2005*, vol.1, no., pp. 59-62, Jun. 2005

[7] Xiaoqing Ge; Arcak, M.; Salama, K.N.; , "Nonlinear analysis of ring oscillator circuits," *American Control Conference (ACC), 2010* , vol., no., pp.1772-1776, June 30 2010-July 2 2010

[8] Tomar, A.; Pokharel, R.K.; Nizhnik, O.; Kanaya, H.; Yoshida, K.; , "Design of 1.1 GHz Highly Linear Digitally-Controlled Ring Oscillator with Wide Tuning Range," *Radio-Frequency Integration Technology, 2007. RFIT 007. IEEE International Workshop on* , vol., no., pp.82-85, 9-11 Dec. 2007

[9] Frankie King-Sun Cheng; Cheong-Fat Chen; Oliver Chiu-Sing Choy; , "A 1.0 µm CMOS all-digital clock multiplier," *Circuits and Systems, 1997. Proceedings of the 40th Midwest Symposium on* , vol.1, no., pp.460-462 vol.1, 3-6 Aug 1997

[10] Rezayee, A.; Martin, K.; , "A coupled two-stage ring oscillator," *Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium on* , vol.2, no., pp.878-881 vol.2, 2001

[11] Toh, Y.; McNeill, J.A.; , "Single-ended to differential converter for multiple-stage single-ended ring oscillators," *Solid-State Circuits, IEEE Journal of*, vol.38, no.1, pp. 141- 145, Jan 2003

[12] Pasha, M.T.; Vesterbacka, M.; , "Frequency control schemes for single-ended ring oscillators," *Circuit Theory and Design (ECCTD), 2011 20th European Conference on* , vol., no., pp.361-364, 29-31 Aug. 2011

[13] Baronti, F.; Fanucci, L.; Lunardini, D.; Roncella, R.; Saletti, R.; , "On-line calibration for non-linearity reduction of delay-locked delay-lines," *Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on* , vol.2, no., pp.1001-1005 vol.2, 2001

[14] Grozing, M.; Berroth, M.; , "Derivation of single-ended CMOS inverter ring oscillator close-in phase noise from basic circuit and device properties," *Radio Frequency Integrated Circuits (RFIC) Symposium, 2004. Digest of Papers. 2004 IEEE* , vol., no., pp. 277- 280, 6-8 June 2004

[15] Chengxin Liu; McNeill, J.A.; , "Jitter in deep submicron CMOS single-ended ring oscillators," *ASIC, 2003. Proceedings. 5th International Conference on* , vol.2, no., pp. 715- 718 Vol.2, 21-24 Oct. 2003

[16] Parvizi, M.; Khodabakhsh, A.; Nabavi, A.; , "Low-power high-tuning range CMOS ring oscillator VCOs," *Semiconductor Electronics, 2008. ICSE 2008. IEEE International Conference on* , vol., no., pp.40-44, 25-27 Nov. 2008

[17] Wei-Jie Zhu; Jian-Guo Ma; , "Investigating the effects of the number of stages on phase noise in CMOS ring oscillators," *Integrated Circuits, ISIC '09. Proceedings of the 2009 12th International Symposium on* , vol., no., pp.612-615, 14-16 Dec. 2009

[18] Sai, A.; Yamaji, T.; Itakura, T.; , "A low-jitter clock generator based on ring oscillator with 1/f noise reduction technique for next-generation mobile wireless terminals," *Solid-State Circuits Conference, 2008. A-SSCC '08. IEEE Asian* , vol., no., pp.425-428, 3-5 Nov. 2008

[19] Virtual Prototyping, Mentor Graphics Inc.; http://www.mentor.com/products/esl/virtual-prototyping

[20] Virtual Platforms, Synopsys Corp.; http://www.synopsys.com

[21] Prototyping Services, The MOSIS Service; http://www.mosis.com/

[22] Dong Lin; Shiyuan Yang; , "An Implementation of Rapid Prototyping Platform of Embedded Systems," *Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on* , vol., no., pp.1-4, 2006

[23] Vahid, V.; Lysecky R.; Zhang, C.; Stitt G.; , "Highly Configurable Platforms for Embedded Computing Systems," *Microelectronics Journal 34(2003)* 1025-1029, 2003

[24] Or-Bach, Z.; , "(When) Will FPGAs Kill ASICs?," 38th Design Automation Conference, pp. 321-322, 2001

[25] Zhang C.; Vahid F, Najjar W.; , "A highly configurable cache architecture for embedded systems," *Proceedings of the 30th International Symposium on Computer Architecture*, June 2003

[26] Triscend Corp.; http://www.triscend.com, 2006

[27] Atmel Corp.; http://www.atmel.com, 2006

[28] Altera Corp.; http://www.altera.com, 2006

[29] Xilinx, Inc.; http://www.xilinx.com, 2006

[30] Gemignani, V.; Faita, F.; Giannoni, M.; Benassi, A.; , "A DSP-based platform for rapid prototyping of real time image processing systems," *Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the 3rd International Symposium on* , vol.2, no., pp. 936- 939 Vol.2, 18-20 Sept. 2003

[31] Galke, C.; Pflanz, M.; Vierhaus, H.T.; , "A test processor concept for systems-on-a-chip," *Computer Design: VLSI in Computers and Processors, 2002. Proceedings. 2002 IEEE International Conference on* , vol., no., pp. 210- 212, 2002

[32] Bahl S.; Singh B.; , "On-Chip and At-Speed Tester for Testing and Characterization of Different Types of Memories," US Patent # 7,353,442 B2, April 1st, 2008

[33] Franco P.; Farwell R.; McCluskey E.; , "An Experimental Chip to Evaluate Test Techniques: Chip and Experiment Design," *Proceedings of the IEEE International Test Conference on Driving Down the Cost of Test*, pp. 653 – 662, 1995

[34] Maggioni S.; Veggetti A.; Bogliolo A.; Croce L.; , "Random Sampling for On-Chip Characterization of Standard-Cell Propagation Delay," *Proc. of the 4th International Symposium on Quality Electronic Design*, pp. 41-45, 2003

[35] Mosterdini L.; , "FPGA-based low-cost automatic test equipment for digital integrated circuits," *Proc. IEEE workshop IDAACS*, pp. 32-37, 2009

[36] Ciganda L.; , "FPGA-based Low-cost Automatic Test Equipment for Digital Integrated Circuits," *Proc. 12th IEEE Symp. DDECS*, pp. 258-263, 2009

[37] Davis J.; , "An FPGA-based Digital Logic Core for ATE Support and Embedded Test Applications," PhD. Dissertation, Georgia Institute of Technology, 2003

[38] Kaam K.; , "Integrated Circuit (IC) With On-Board Characterization Unit," US Patent 7,475,312 B2, Jan. 6, 2009

[39] Pal Singh, J.; Kumar, A.; Kumar, S.; , "A multiplier generator for Xilinx FPGAs," *VLSI Design, 1996. Proceedings., Ninth International Conference on* , vol., no., pp.322-323, 3-6 Jan 1996

[40] Beck M.; Barondeau O.; Kaibel M.; Poehl F.; Lin X.; Press R.; , "Logic design for on-chip test clock generation - implementation details and impact on delay test quality," *Design, Automation and Test in Europe, 2005. Proceedings*, vol.; no.; pp. 56- 61 Vol. 1, 7-11 March 2005

[41] Setu M.; Raju G.; Weerasekera R.; , "An on-chip low power clock multiplier unit in 0.25 micron technology," *Electrical and Computer Engineering, 2005. Canadian Conference on* , vol.; no.; pp.1735-1738, 1-4 May 2005

[42] Lee S.; Seo Y.; Suh Y.; Park H.; Sim J.; , "A 1GHz ADPLL with a 1.25ps minimum-resolution sub-exponent TDC in 0.18µm CMOS," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International* , vol.; no.; pp.482-483, 7-11 Feb. 2010

[43] Olsson T.; Torkelsson M.; Nilsson P.; Hemani A.; Meincke T.; , "A digitally controlled on-chip clock multiplier for globally asynchronous locally synchronous systems," *Circuits and Systems, 1999. 42nd Midwest Symposium on* , vol.1, no.; pp.84-87 vol. 1, 1999

[44] Combes M.; Dioury K.; Greiner A.; , "A portable clock multiplier generator using digital CMOS standard cells," *Solid-State Circuits, IEEE Journal of*, vol.31, no.7, pp.958-965, Jul 1996

[45] Liu S.L.; Mourad S.; Krishnan S.; , "At-speed testing of data communications transceivers," *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on* , vol.4, no.; pp.9-12 vol. 4, 6-9 May 2001

[46] Swanson B.; , "At-speed testing made easy," August 6, 2004 from: http://eetimes.com/electronics-news/4154986/At-speed-testing-made-easy

[47] Sato Y.; Sato M.; Tsutsumida K.; Kawashima M.; Hatayama K.; Nomoto K.; , "DFT timing design methodology for at-speed BIST," *Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003. Asia and South Pacific* , vol.; no.; pp. 763- 768, 21-24, Jan. 2003

[48] Hyunbean Yi; Jaehoon Song; Sungju Park; , "Low-Cost Scan Test for IEEE-1500-Based SoC," *Instrumentation and Measurement, IEEE Transactions on* , vol.57, no.5, pp.1071-1078, May 2008

[49] Elrabaa M.; , "Method for Digital Integrated Circuits Testing and Characterization," US Patent application number 13/471346, Filed on May 14, 2012

APPENDICES

APPENDIX "A"

CCG Schematics

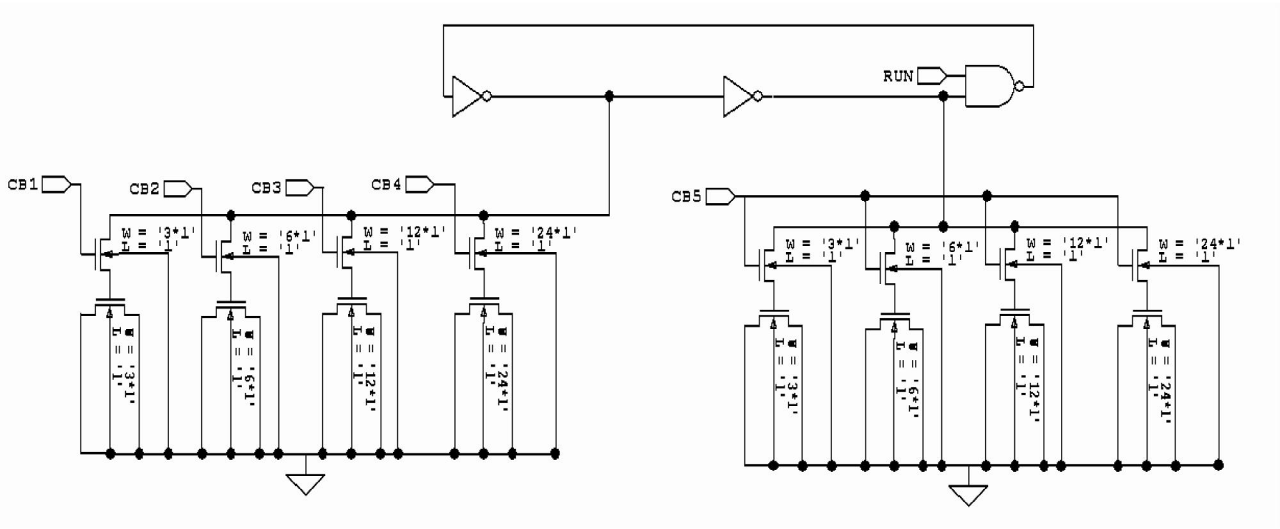


Figure 53 Schematic of the DCO

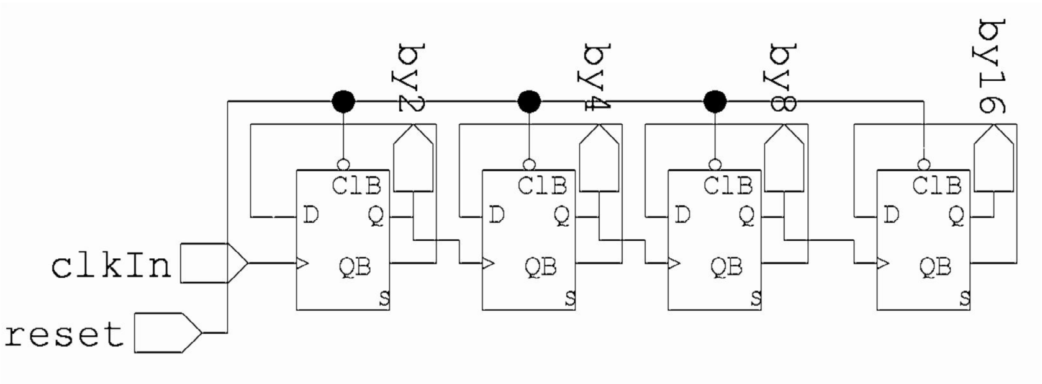


Figure 54 Schematic of the frequency divider

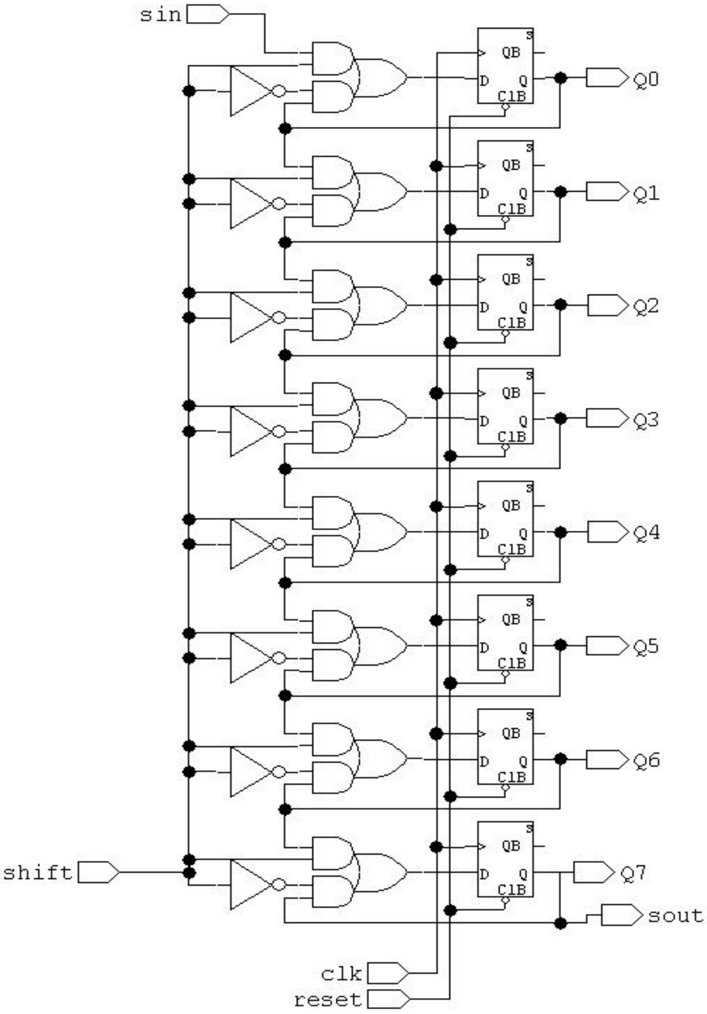


Figure 55 Schematic of the control word register

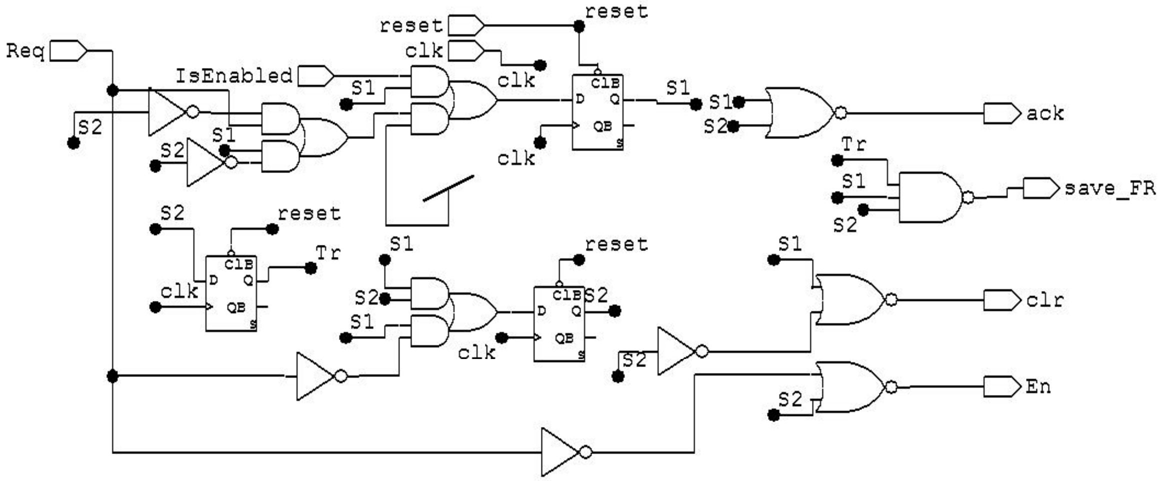


Figure 56 Schematic of the FSM

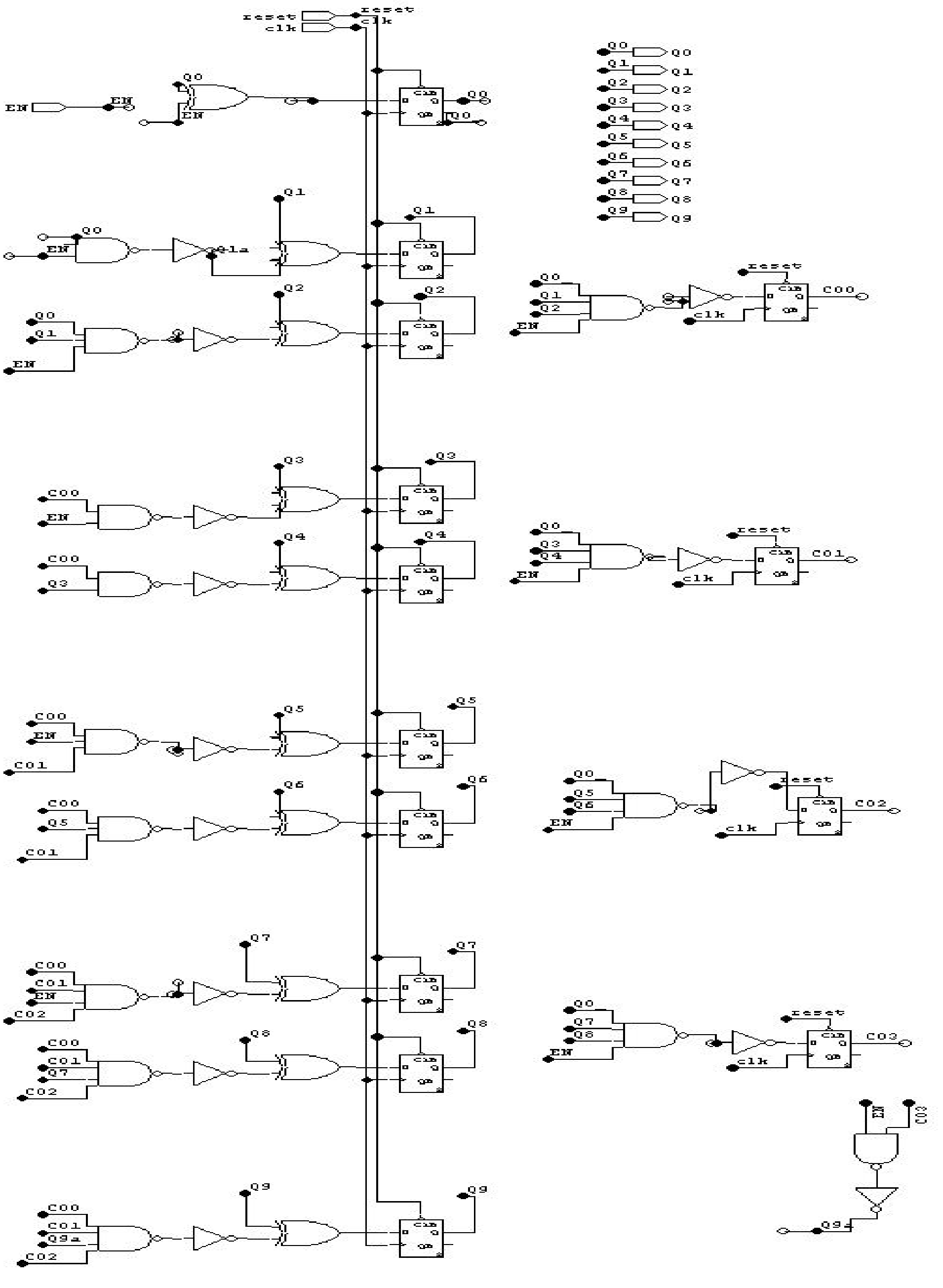


Figure 57 Schematic of the high frequency counter

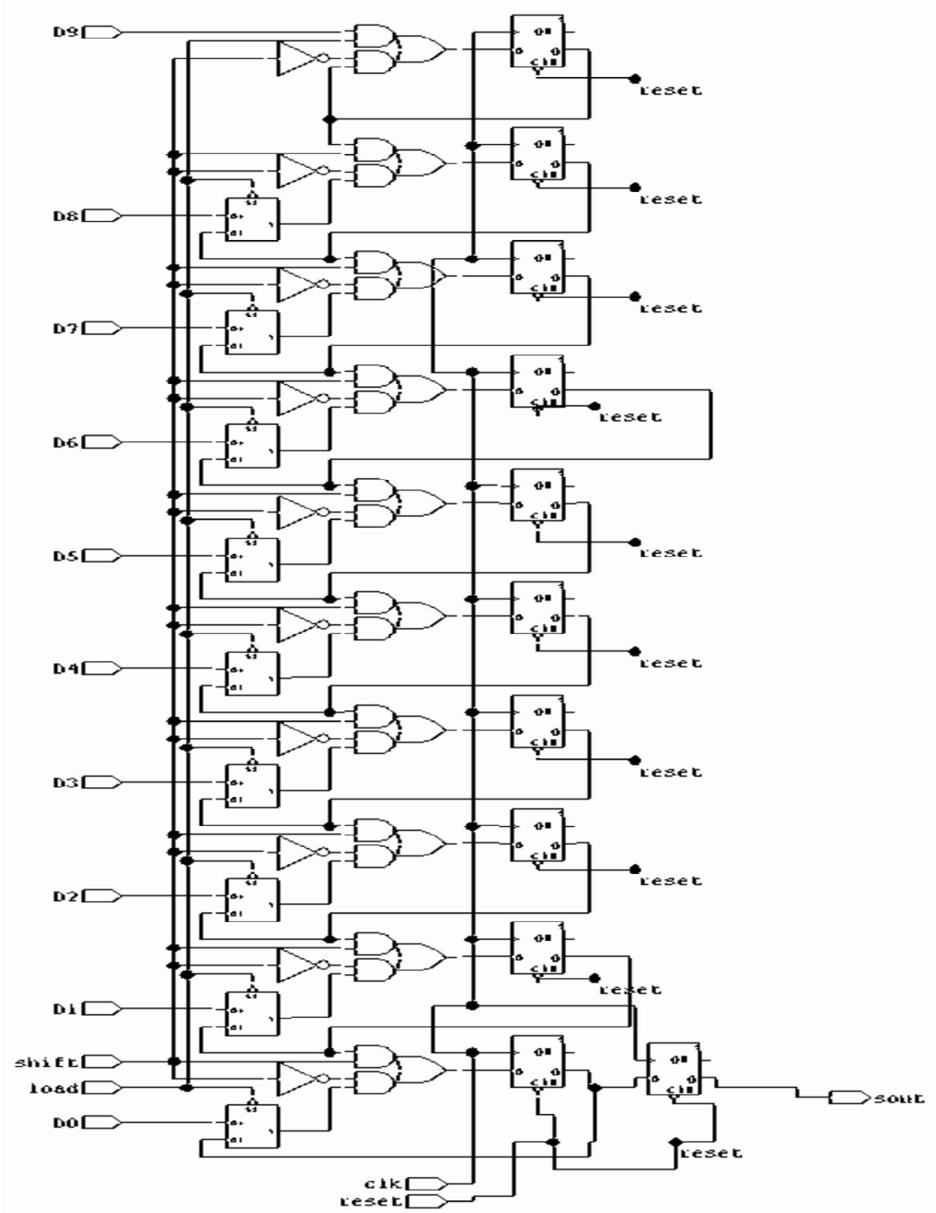
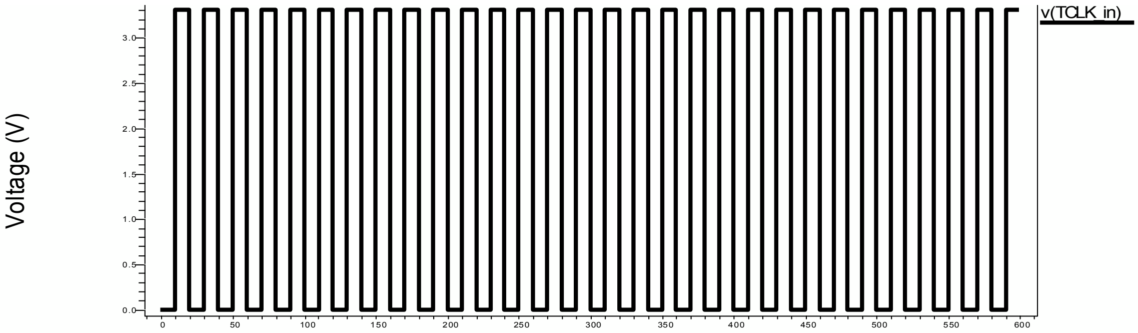
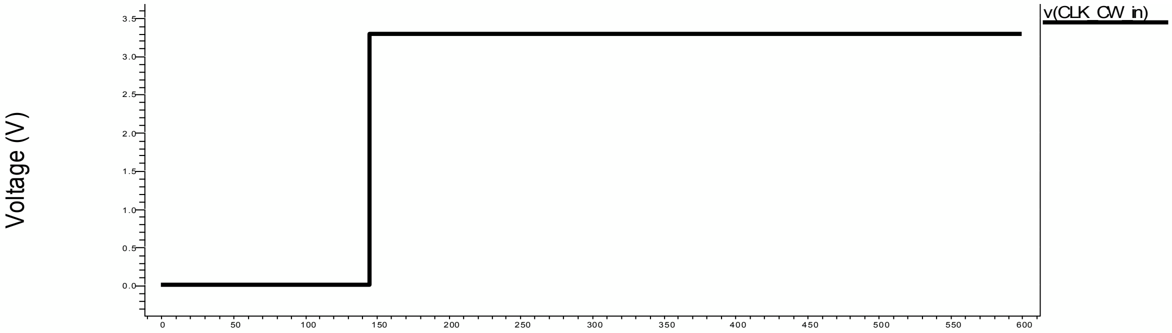


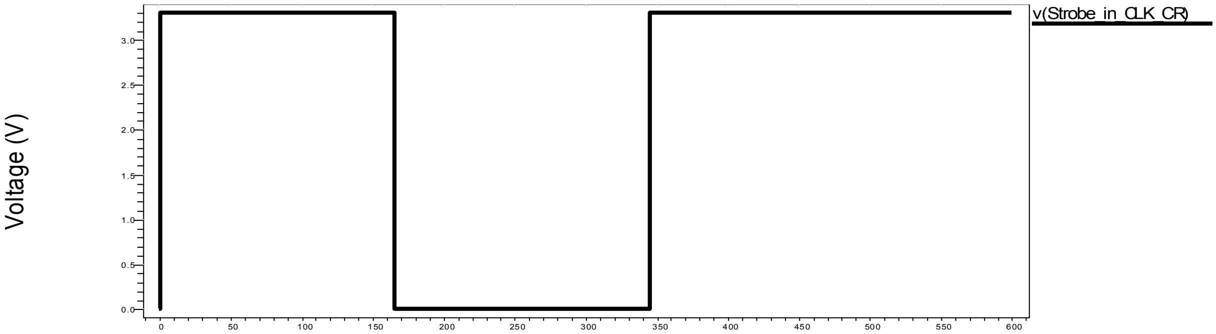
Figure 58 Schematic of the frequency register

APPENDIX "B"

CCG Simulations







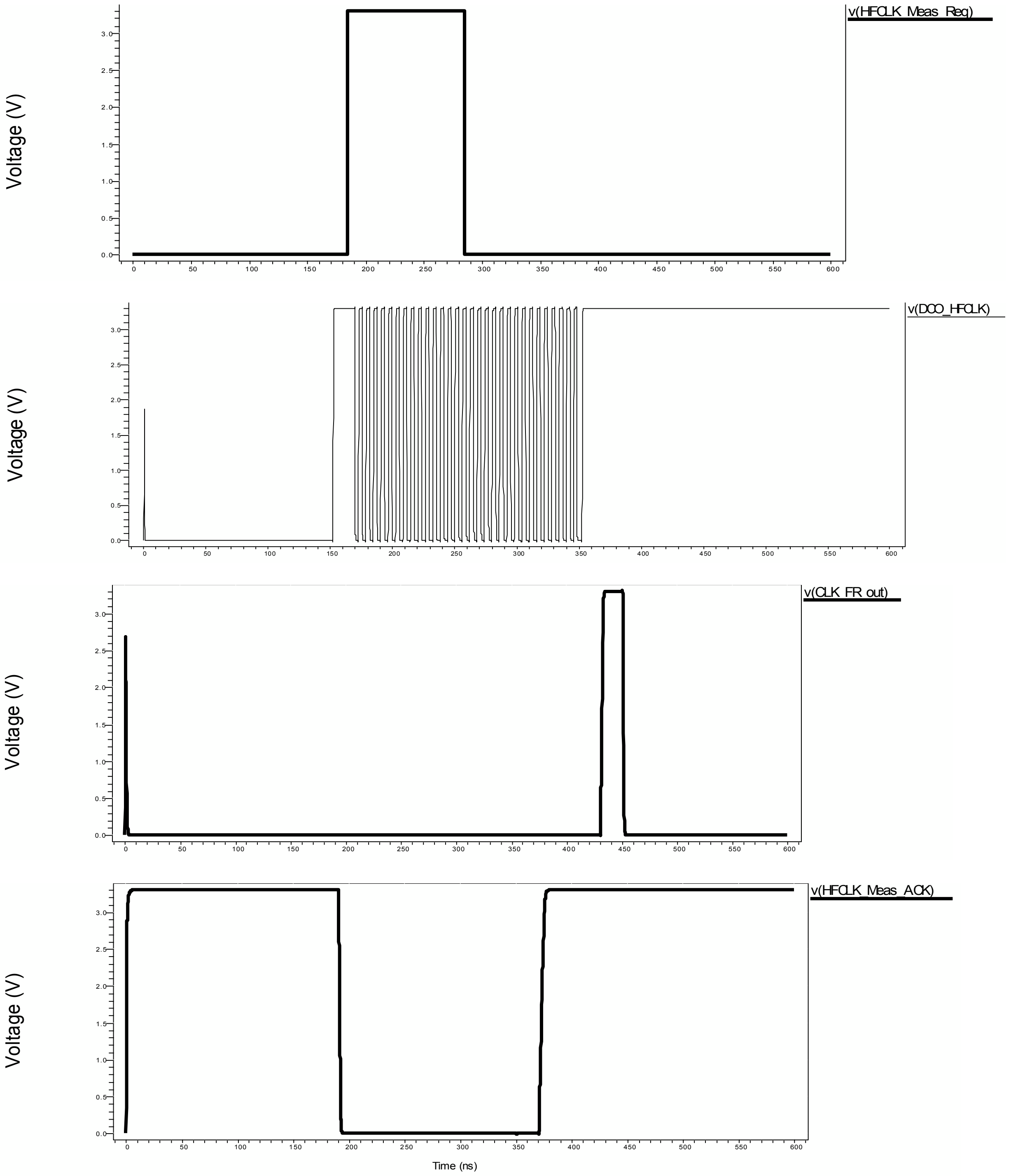
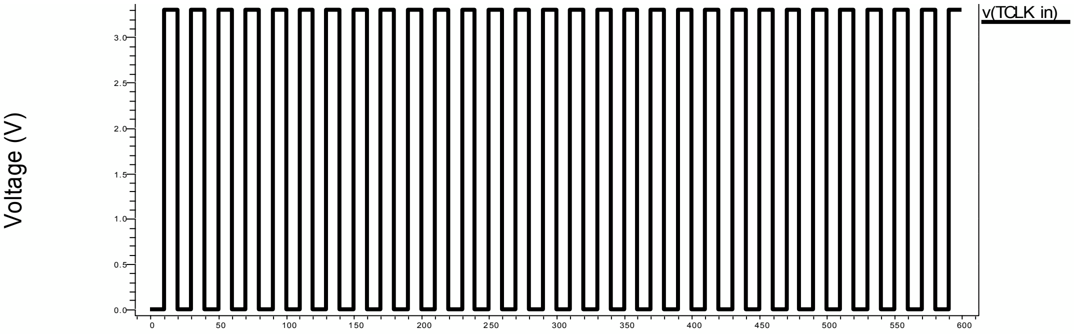
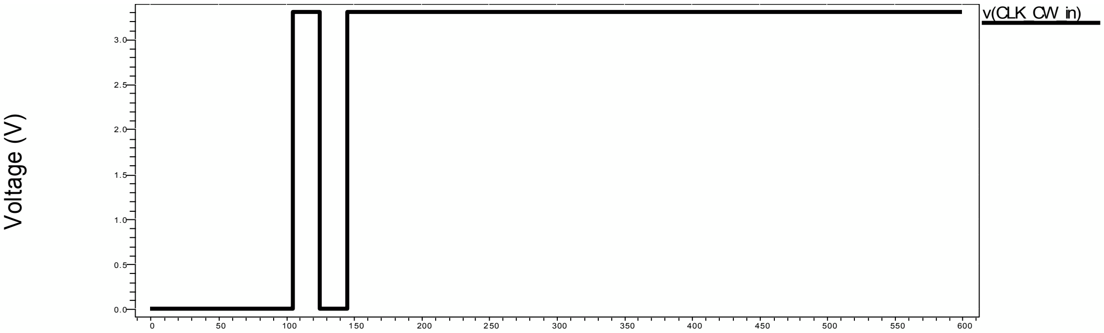
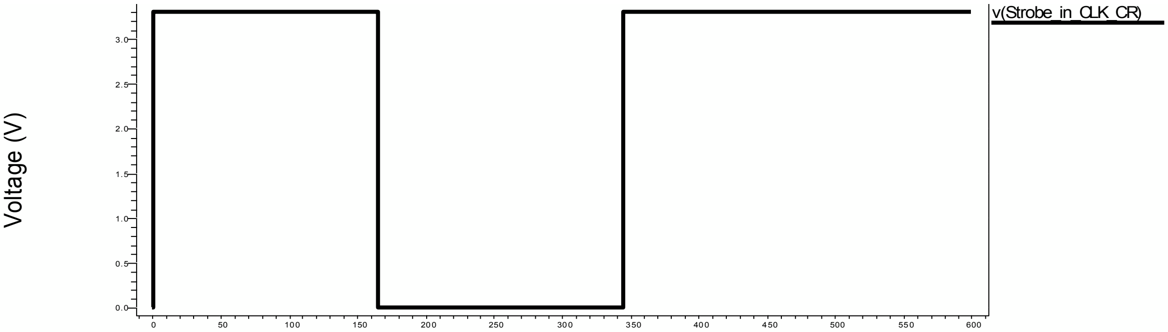
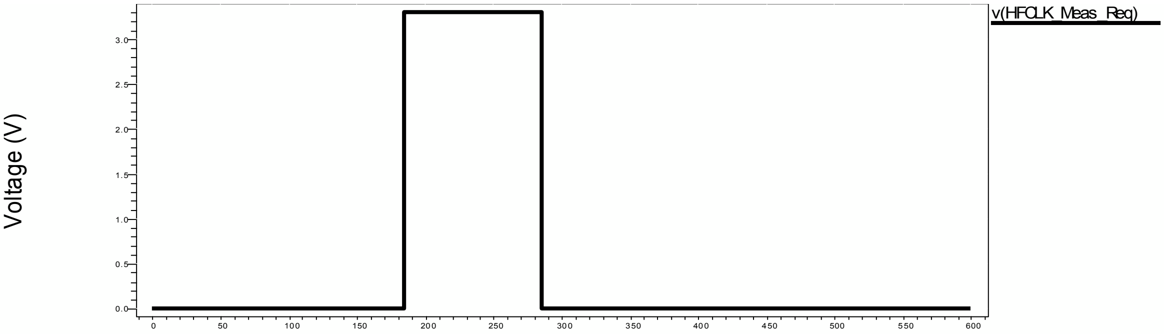


Figure 59 Post-layout simulation of the CCG using DCO frequency divided by 2


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(TCLK_in)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(CLK_CW_in)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(Strobe_in_CLK_CR)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(HFCLK_Meas_Req)

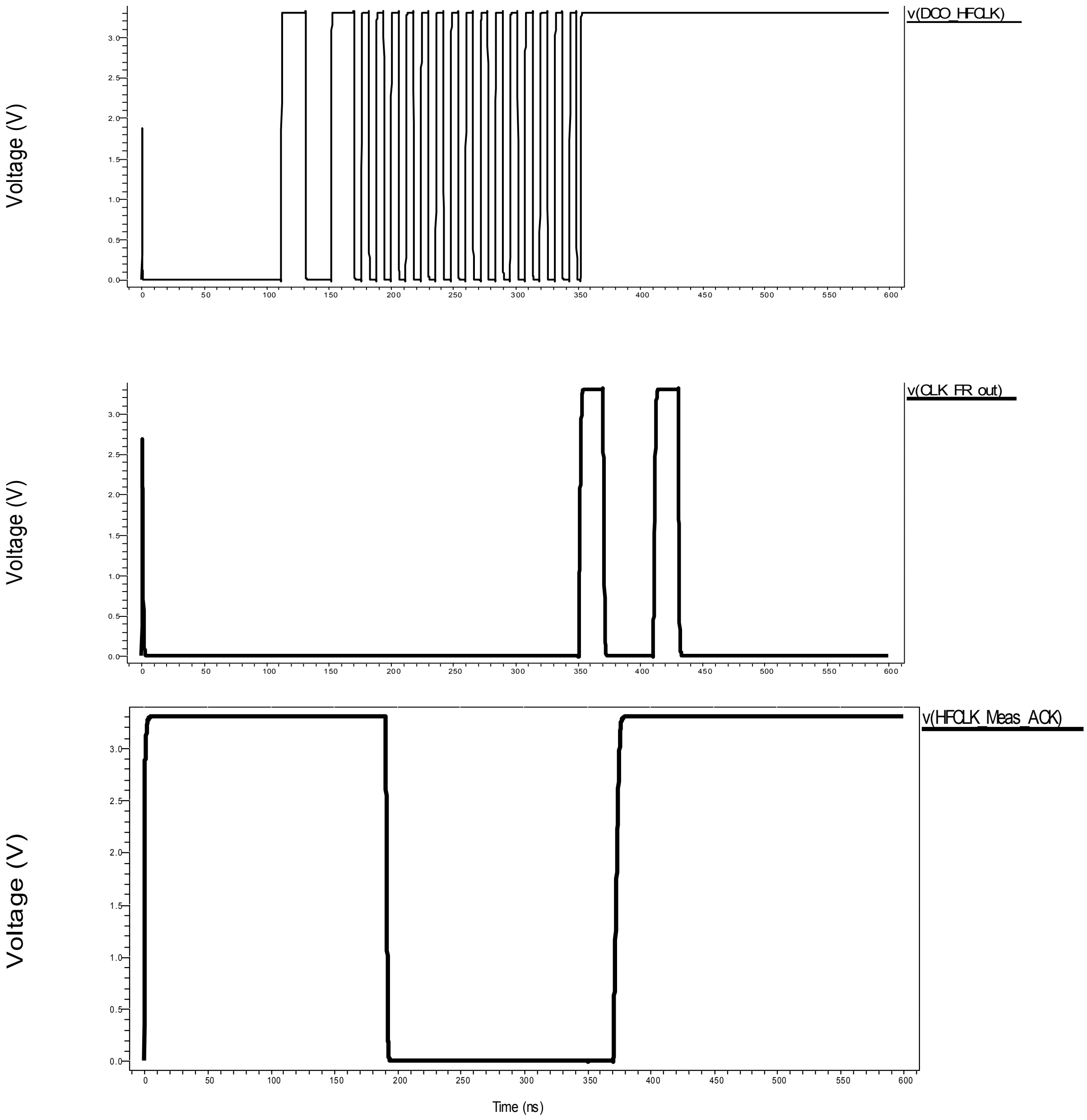
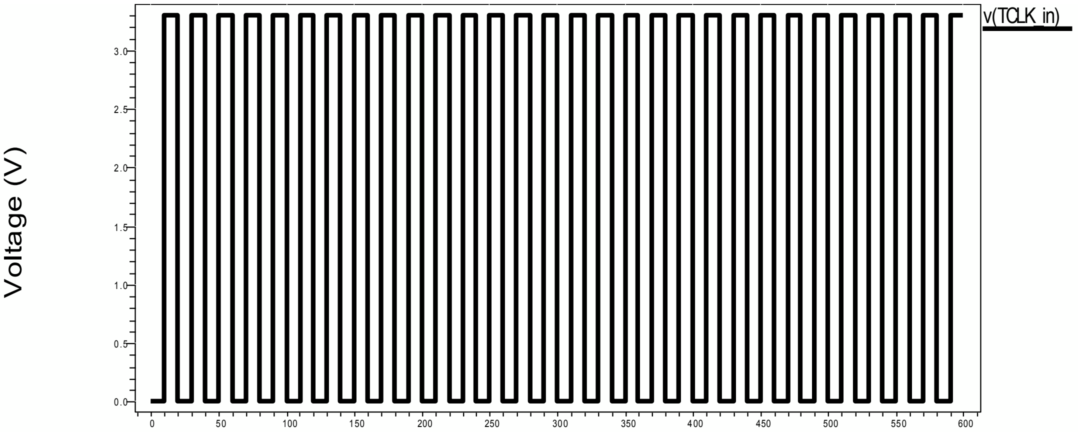
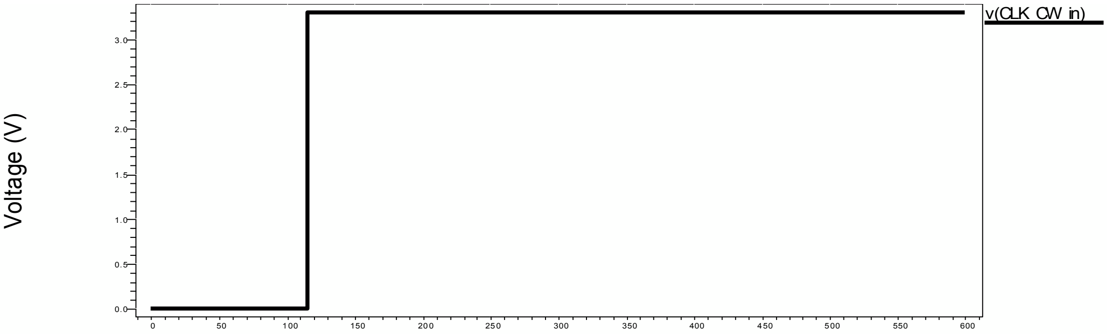
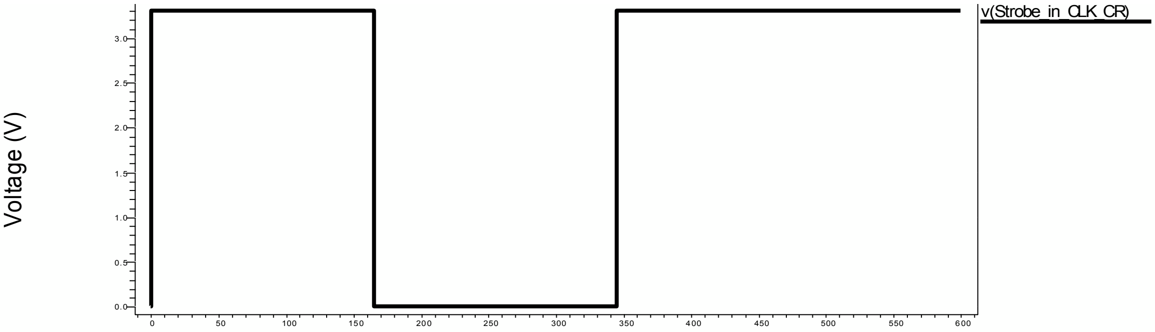
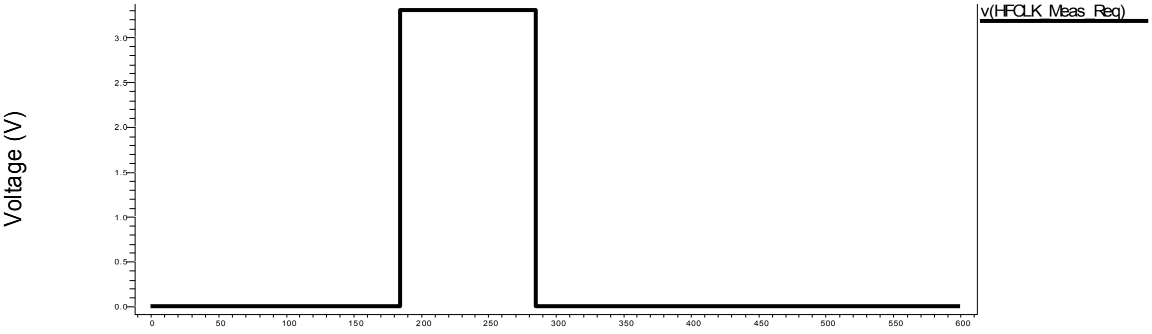


Figure 60 Post-layout simulation of the CCG using DCO frequency divided by 4


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(TCLK in)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(CLK CW in)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(Strobe in CLK CR)


Voltage (V)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
450
500
550
600
v(HFCLK Meas Req)

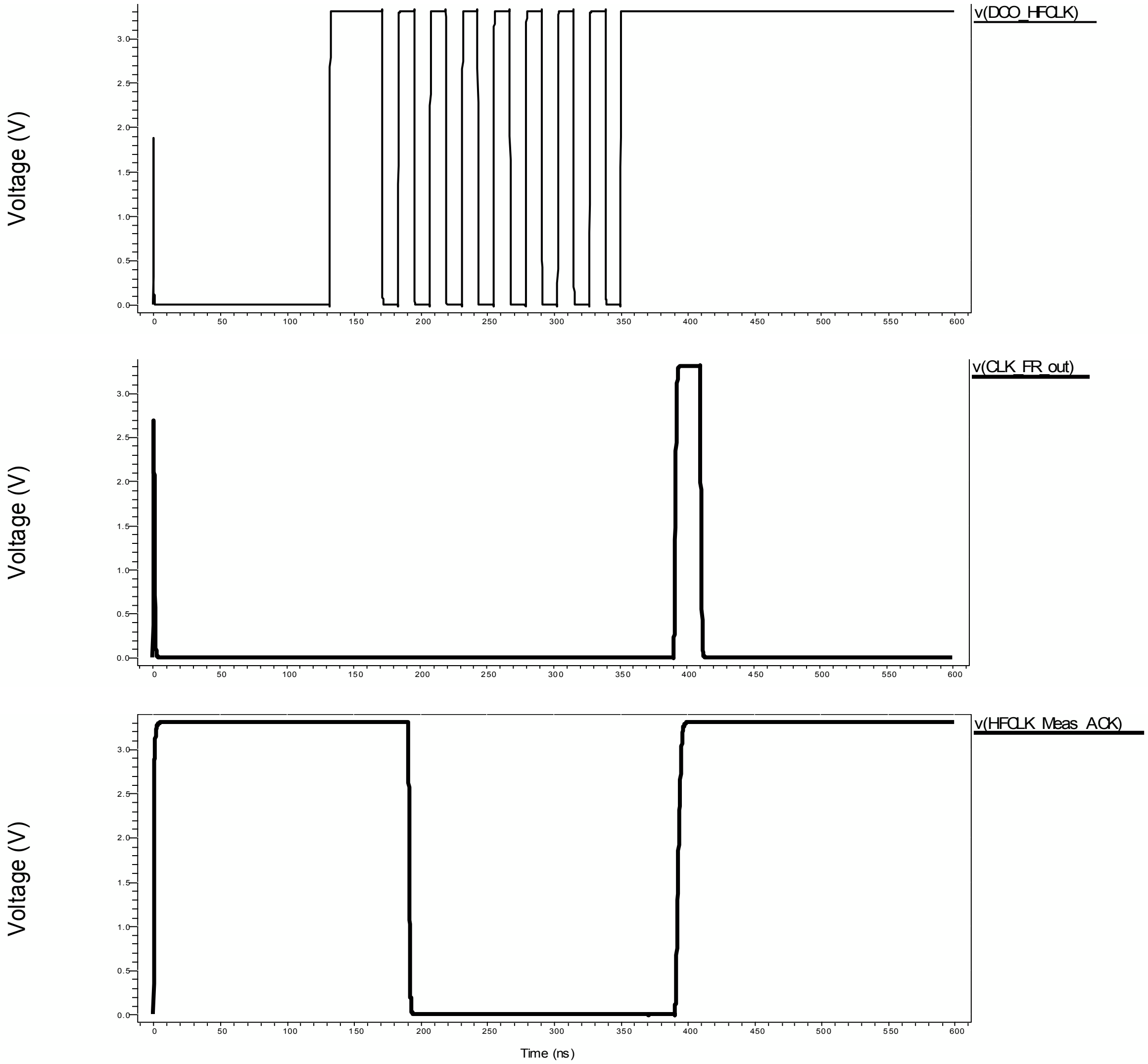
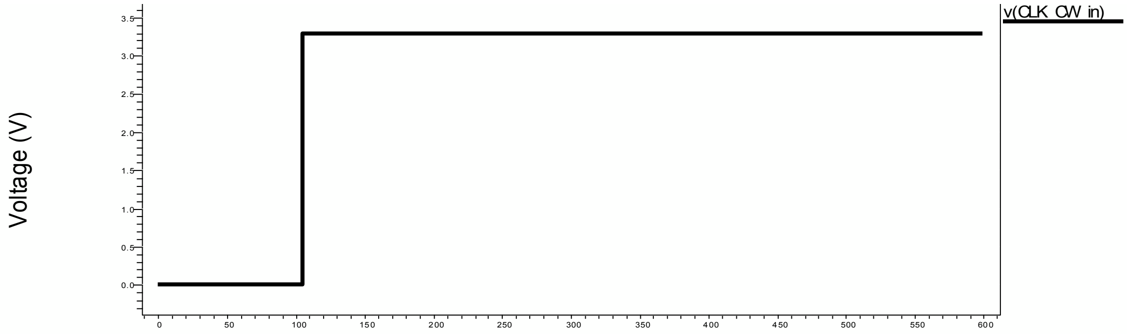
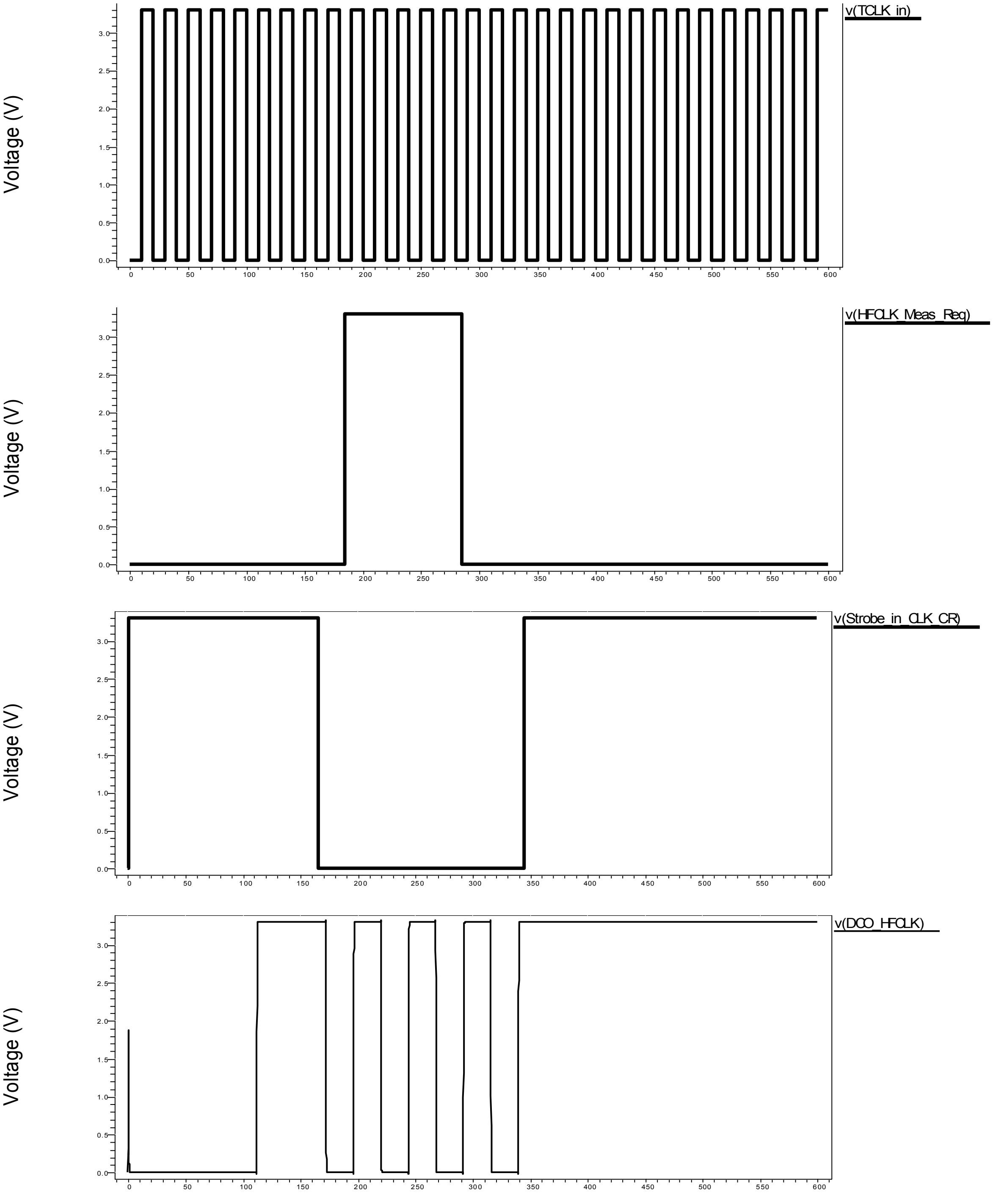


Figure 61 Post-layout simulation of the CCG using DCO frequency divided by 8




v(TCLK_in)
v(HFCLK_Meas_Req)
v(Strobe_in_CLK_CR)
v(DCO_HFCLK)
Voltage (V)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
50
100
150
200
250
300
350
400
450
500
550
600

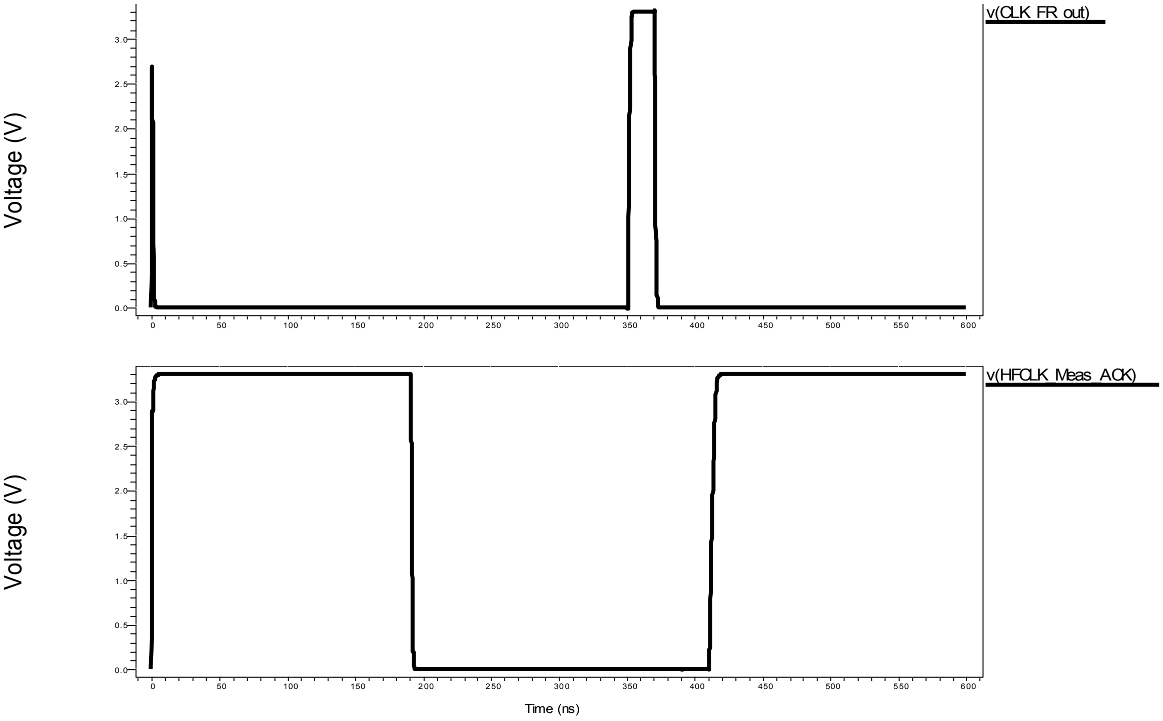


Figure 62 Post-layout simulation of the CCG using DCO frequency divided by 16

APPENDIX "C"

Support Circuitry Simulations using S820s Benchmark

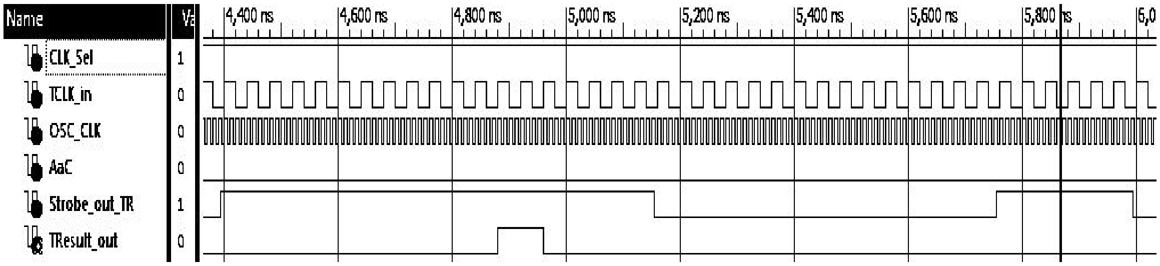


Figure 63 Support Circuitry simulation; test vector is 00011010001000110111111 and expected result is 000000000000000000110000

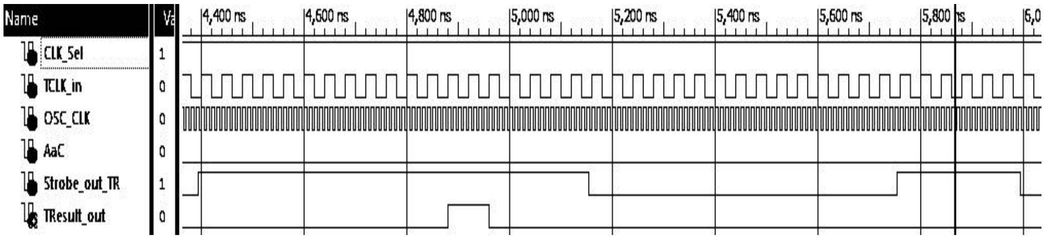


Figure 64 Support Circuitry simulation; test vector is 00001011001100011101001and expected result is 000000000000000000110000

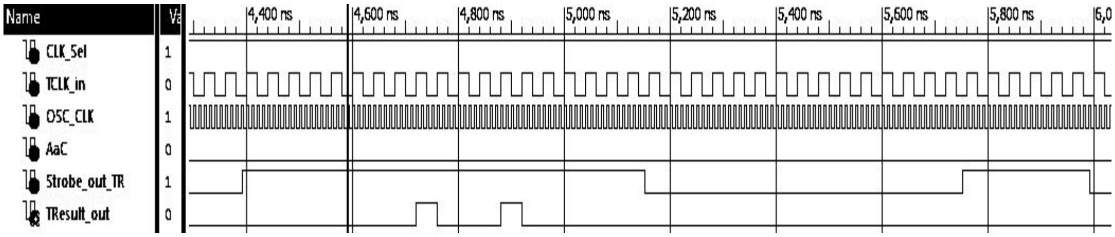


Figure 65 Support Circuitry simulation; test vector is 11100101101010111011001 expected result is 0000000000000001000100000

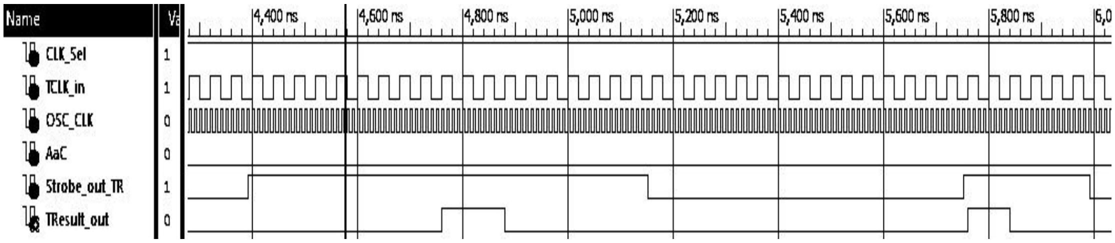


Figure 66 Support Circuitry simulation; test vector is 01001110100111000011100 expected result is 00011000000000111000000

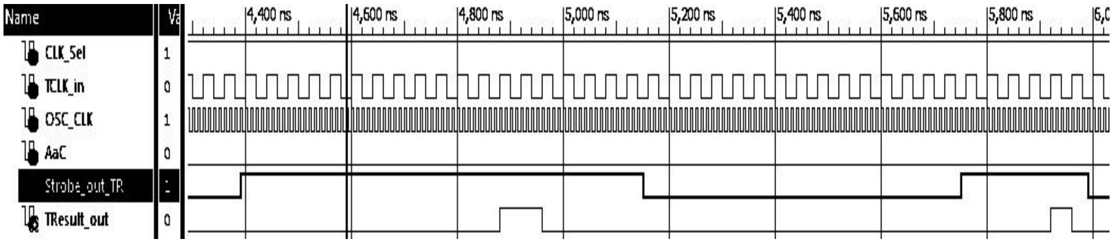


Figure 67 Support Circuitry simulation; test vector is 11110010111011100100000 expected result is 10000000000000000110000

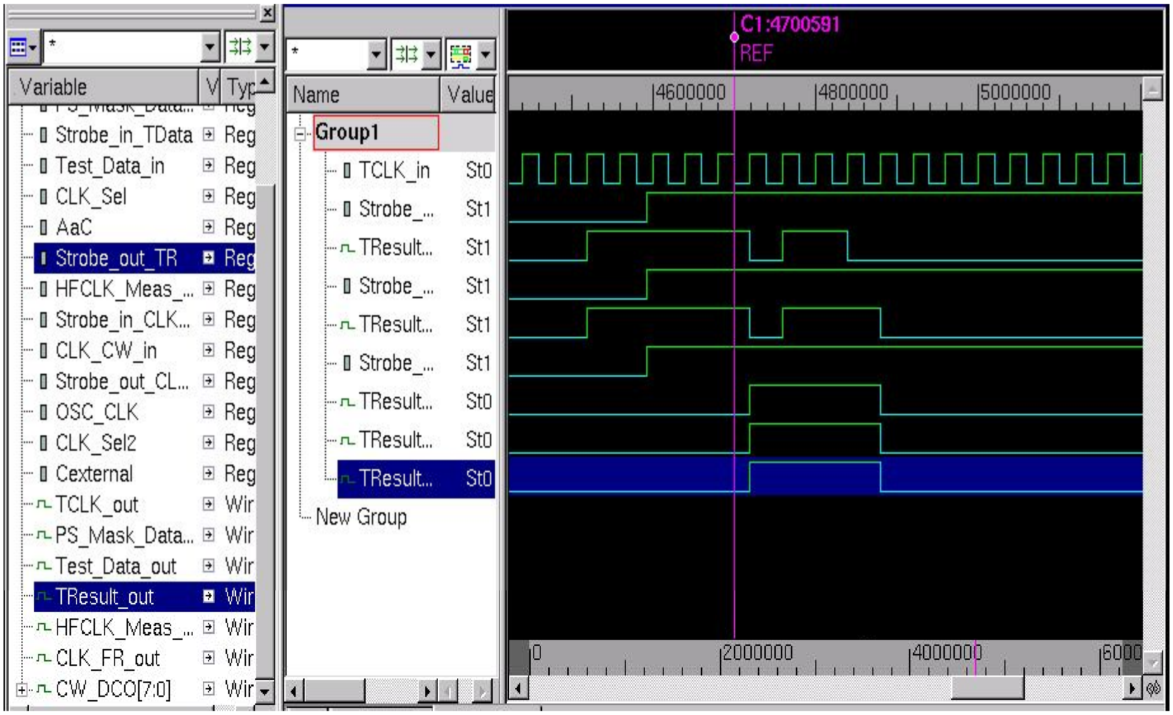


Figure 68 Pre-synthesis, post-synthesis and post-layout simulation of the support circuitry using 8 bit pipelined adder IUT; 00010101+11011010+1=11110000; the last same three signals are the pre-synthesis, post-synthesis and post-layout results

APPENDIX "D"

Synopsys Scripts

DC Compiler:

```
read_file -format verilog {SupportCircuitry.v}

uniquify

analyze -format verilog {SupportCircuitry.v}

link

create_clock -name "clk" -period 20 -waveform { 0 10  }  {
clk  }

set_max_area 0

compile -exact_map

write -hierarchy -format verilog -output
SupportCircuitry_synthesized.v

write_sdc SupportCircuitry_constraints.sdc

read_file -format verilog {CCG.v}

uniquify

analyze -format verilog {CCG.v}

link

create_clock -name "clk" -period 20 -waveform { 0 10  }  {
clk  }

create_clock -name "OSCclk" -period 1 -waveform { 0 0.5  }
{ osc_clk  }
```

```
set_max_area 0
compile -exact_map
write -hierarchy -format verilog -output CCG_synthesized.v
write_sdc CCG_constraints.sdc
```

IC Compiler

```
open_mw_lib lf150dhs9s
set_tlu_plus_files -max_tluplus lf150_best.tluplus -
min_tluplus lf150_worst.tluplus -tech2itf_map  itf_mapping
set stdcells_home 4metal/
set_app_var search_path "$stdcells_home"
set_app_var target_library
/4metal/lf150dhs9s/LM/lf150dhs9s_best_best.db
set_app_var link_library "* $target_library"
read_verilog { SupportCircuitry_synthesized.v}
read_sdc  -version Latest "
SupportCircuitry_constraints.sdc"
 initialize_floorplan -control_type "aspect_ratio" -
core_aspect_ratio "1" -core_utilization "0.7" -
row_core_ratio "1" -left_io2core "30" -bottom_io2core "30"
-right_io2core "30" -top_io2core "30" -start_first_row
create_fp_placement
set mw_logic0_net "gnd!"
set mw_logic1_net "vdd!"
```

```
derive_pg_connection -power_net {vdd!}  -ground_net {gnd!}
create_rectilinear_rings -nets {vdd! gnd!} -offset {1 1} -
space {3 3}
preroute_standard_cells -nets {vdd! gnd!} -connect
horizontal -extend_to_boundaries_and_generate_pins
synthesize_fp_rail -power_budget "1000" -voltage_supply
"1.8" -target_voltage_drop "90" -output_dir "./pna_output"
-nets {vdd! gnd!} -create_virtual_rails "METAL1" -
synthesize_power_plan -synthesize_power_pads -
use_strap_ends_as_pads
commit_fp_rail
clock_opt -only_cts -no_clock_route
route_zrt_group -all_clock_nets -
reuse_existing_global_route true
route_opt -effort medium
insert_stdcell_filler \
-cell_without_metal "FILLCELL_X1 FILLCELL_X2 FILLCELL_X4
FILLCELL_X8 FILLCELL_X16 FILLCELL_X32 FILLCELL_X64" \
-connect_to_power vdd! -connect_to_ground gnd!
set_write_stream_options -output_pin {text geometry} -
keep_data_type
write_stream  -format gds " SupportCircuitry.gds"

read_verilog { SupportCircuitry_synthesized.v}
```

```
read_sdc  -version Latest " CCG_constraints.sdc"
 initialize_floorplan -control_type "aspect_ratio" -
core_aspect_ratio "1" -core_utilization "0.7" -
row_core_ratio "1" -left_io2core "30" -bottom_io2core "30"
-right_io2core "30" -top_io2core "30" -start_first_row
create_fp_placement
set mw_logic0_net "gnd!"
set mw_logic1_net "vdd!"
derive_pg_connection -power_net {vdd!}  -ground_net {gnd!}
create_rectilinear_rings -nets {vdd! gnd!} -offset {1 1} -
space {3 3}
preroute_standard_cells -nets {vdd! gnd!} -connect
horizontal -extend_to_boundaries_and_generate_pins
synthesize_fp_rail -power_budget "1000" -voltage_supply
"1.8" -target_voltage_drop "90" -output_dir "./pna_output"
-nets {vdd! gnd!} -create_virtual_rails "METAL1" -
synthesize_power_plan -synthesize_power_pads -
use_strap_ends_as_pads
commit_fp_rail
clock_opt -only_cts -no_clock_route
route_zrt_group -all_clock_nets -
reuse_existing_global_route true
route_opt -effort medium
insert_stdcell_filler \
-cell_without_metal "FILLCELL_X1 FILLCELL_X2 FILLCELL_X4

FILLCELL_X8 FILLCELL_X16 FILLCELL_X32 FILLCELL_X64" \

-connect_to_power vdd! -connect_to_ground gnd!

set_write_stream_options -output_pin {text geometry} -

keep_data_type

write_stream  -format gds " CCG.gds"
```

APPENDIX "E"

Test Bench of the Support Circuitry

```
`timescale 1ns / 1ps
module sim;

	// Inputs
	reg reset;
	reg TCLK_in;
	reg Strobe_in_PMask;
	reg PS_Mask_Data_in;
	reg Strobe_in_TData;
	reg Test_Data_in;
	reg CLK_Sel;
	reg AaC;
	reg Strobe_out_TR;
	reg HFCLK_Meas_Req;
	reg Strobe_in_CLK_CR;
	reg CLK_CW_in;
	reg Strobe_out_CLK_FR;
	reg OSC_CLK;
	reg CLK_Sel2;
	reg Cexternal;

	// Outputs
	wire TCLK_out;
	wire PS_Mask_Data_out;
	wire Test_Data_out;
	wire TResult_out;
```

```
     wire HFCLK_Meas_ACK;
     wire CLK_FR_out;
     wire [7:0] CW_DCO;

     // Instantiate the Unit Under Test (UUT)
     SupportCircuitry uut (
          .reset(reset),
          .TCLK_in(TCLK_in),
          .TCLK_out(TCLK_out),
          .Strobe_in_PMask(Strobe_in_PMask),
          .PS_Mask_Data_in(PS_Mask_Data_in),
          .PS_Mask_Data_out(PS_Mask_Data_out),
          .Strobe_in_TData(Strobe_in_TData),
          .Test_Data_in(Test_Data_in),
          .Test_Data_out(Test_Data_out),
          .CLK_Sel(CLK_Sel),
          .AaC(AaC),
          .Strobe_out_TR(Strobe_out_TR),
          .TResult_out(TResult_out),
          .HFCLK_Meas_Req(HFCLK_Meas_Req),
          .HFCLK_Meas_ACK(HFCLK_Meas_ACK),
          .Strobe_in_CLK_CR(Strobe_in_CLK_CR),
          .CLK_CW_in(CLK_CW_in),
          .Strobe_out_CLK_FR(Strobe_out_CLK_FR),
          .CLK_FR_out(CLK_FR_out),
          .CW_DCO(CW_DCO),
          .OSC_CLK(OSC_CLK),
          .CLK_Sel2(CLK_Sel2),
          .Cexternal(Cexternal)
     );

//test    1: 00011010001000110111111
000000000000000000001000   49 faults detected
//test    2: 00001011001100011101001
000000000000000000000000   17 faults detected
//test    3: 11100101101010111011001
000000000000000000001000   20 faults detected
//test    4: 01001110100111000011100
000100100000000000001001   32 faults detected
//test    5: 01010010001011000010001
000000000000000000001001    6 faults detected
//test    6: 01000011111010000110011
000000000000000000001001   12 faults detected
//test    7: 11110010111011100100000
000000000000000000000000   24 faults detected
//test    8: 00000111110011001110100
100000000000000000110000   25 faults detected
```

```
//test    9: 101100001101001010100001
000000000000000000000000    4 faults detected
//test   10: 011000110100101100011111
000000001010000000000000   14 faults detected
//test   11: 100111010110111010011111
000000000000000000000000    2 faults detected

//test vectors and expected results after applying two
clocks
//000110100010001101111111 000000000000000000110000
//000010110011000111010001 000000000000000000110000
//111001011010101110110001 000000000000001000100000
//010011101001110000111000 000110000000000111000000
//010100100010110000100001 000000000000000000110000
//010000111110100001100111 000000000000000000110000
//111100101110111001000000 100000000000000000110000
//000001111100110011101000 000000000000000000000000
//101100001101001010100001 000000000000001000100000
//011000110100101100011111 000000000000000000110000
//100111010110111010011111 000000000000000000100001
//010101110110000010111111 000000000000000000110000
//001000000110100010001100 000010000000000000000000
//110001111100000011001110 000010000000000000000000
//001110100111100101101010 000000000000000000110000

//// test vector 1
//tv = 000110100010001101111111, expected =
000000000000000000110000
     //parameter [17:0] tv = 18'b111111011000100010;
     //localparam [4:0] scan_tv = 5'b00011;

//// test vector 2
//tv = 000010110011000111010001 , expected =
000000000000000000110000
//   parameter [17:0] tv = 18'b100101110001100110;
//   localparam [4:0] scan_tv = 5'b00001;

//   test vector 3
//tv = 111001011010101110110001, expected =
000000000000001000100000
     parameter [17:0] tv = 18'b100110111010101101;
     localparam [4:0] scan_tv = 5'b11100;

//   test vector 4
//tv = 010011101001110000111000, expected =
000110000000000111000000
     //parameter [17:0] tv = 18'b001110000111001011;
     //localparam [4:0] scan_tv = 5'b01001;

//// test vector 5
//tv = 01010010001011000010001, expected =
000000000000000000110000
//   parameter [17:0] tv = 18'b100010000110100010;
//   localparam [4:0] scan_tv = 5'b01010;

//   test vector 6
//tv = 01000011111010000110011, expected =
000000000000000000110000
//   parameter [17:0] tv = 18'b110011000010111110;
//   localparam [4:0] scan_tv = 5'b01000;

//   test vector 7
//tv = 11110010111011100100000, expected =
100000000000000000110000
     //parameter [17:0] tv = 18'b000001001110111010;
     //localparam [4:0] scan_tv = 5'b11110;

     initial begin
          // Initialize Inputs
          reset = 0;
          TCLK_in = 0;
          Strobe_in_PMask = 0;
          PS_Mask_Data_in = 0;
          Strobe_in_TData = 0;
          Test_Data_in = 0;
          CLK_Sel = 1;
          AaC = 0;
          Strobe_out_TR = 0;
          HFCLK_Meas_Req = 0;
          Strobe_in_CLK_CR = 0;
          CLK_CW_in = 0;
          Strobe_out_CLK_FR = 0;
          OSC_CLK = 0;
          CLK_Sel2 = 0;
          Cexternal = 0;

          // Wait 100 ns for global reset to finish
          #100;
          //reset the support circuitry
          #13 reset = 1;
          #80 reset = 0;
          #120;
          //select between HFCLK (1) and the normal clock
50MHz (0)
          #40 CLK_Sel = 0;

          //step (1)
          //select the 1st port in the 1st IUT
          //send inputs test data vector
          //apply the test data to move it to the parallel
register
          SelectPort('b000000000001);   //select port 0
          SendTestVector(tv);
          #80 AaC = 0;
          #80 AaC = 1; // Apply and capture
          #120 AaC = 0;

          //step (2)
          //select the 2nd port in the 1st IUT
          //send the 5-bits vector to the scan chain
          SelectPort('b000000000100);   //select port 2
          SendScanChain(scan_tv);

          //step (3)
          //select the 1st and the 2nd ports simultanously
          //send the 5-bits vector to the scan chain
          //apply and capture the result
          SelectPort('b000000000001);   //select port 0,1
          #40 Strobe_in_PMask = 1;
          PS_Mask_Data_in = 1;
          #40 Strobe_in_PMask = 0;
          #80 AaC = 0;
          #80 AaC = 1; // Apply and capture
//        #120 AaC = 0;
//        #80 AaC = 1; // Apply and capture
          #120 AaC = 0; // Apply and capture

          //step (4)
          //select the 2nd port of the 1st IUT
          //Read result, i.e. read the scan chain contents
          SelectPort('b000000000010);   //select port 1
          Strobe_out_TR = 0;
          #40 Strobe_out_TR = 1;
          #760 Strobe_out_TR = 0; //two bytes, 19bits*40

          //step (5)
          //select the 3rd port of the 1st IUT
          //Read result, i.e. read the scan chain contents
          SelectPort('b000000000100);   //select port 2
          Strobe_out_TR = 0;
          #40 Strobe_out_TR = 1;
          #240 Strobe_out_TR = 0; //one byte, (1 dump FF +
5bits)*40
          #40 $finish;
     end

     task SelectPort;
     input [11:0] din;
     begin
          #40 Strobe_in_PMask = 0;
          #40 Strobe_in_PMask = 1;
          PS_Mask_Data_in = din[11];
          #40 PS_Mask_Data_in = din[10];
          #40 PS_Mask_Data_in = din[9];
          #40 PS_Mask_Data_in = din[8];
          #40 PS_Mask_Data_in = din[7];
          #40 PS_Mask_Data_in = din[6];
          #40 PS_Mask_Data_in = din[5];
          #40 PS_Mask_Data_in = din[4];
          #40 PS_Mask_Data_in = din[3];
          #40 PS_Mask_Data_in = din[2];
          #40 PS_Mask_Data_in = din[1];
          #40 PS_Mask_Data_in = din[0];
          #40 Strobe_in_PMask = 0;
     end
     endtask

     task SendScanChain;
     input [4:0] din;
     begin
          #40 Strobe_in_TData = 0;
          #40 Strobe_in_TData = 1;
          Test_Data_in = din[0];
          #40 Test_Data_in = din[1];
          #40 Test_Data_in = din[2];
          #40 Test_Data_in = din[3];
          #40 Test_Data_in = din[4];
          #40 Test_Data_in = 0;//DUMP FOR the 1st scan FF
          #40 Strobe_in_TData = 0;
     end
     endtask

     task SendTestVector;
     input [17:0] din;
     begin
          #40 Strobe_in_TData = 0;
          #40 Strobe_in_TData = 1;
          Test_Data_in = din[0];
          #40 Test_Data_in = din[1];
          #40 Test_Data_in = din[2];
          #40 Test_Data_in = din[3];
          #40 Test_Data_in = din[4];
          #40 Test_Data_in = din[5];
          #40 Test_Data_in = din[6];
          #40 Test_Data_in = din[7];
          #40 Test_Data_in = din[8];
          #40 Test_Data_in = din[9];
          #40 Test_Data_in = din[10];
          #40 Test_Data_in = din[11];
          #40 Test_Data_in = din[12];
          #40 Test_Data_in = din[13];
          #40 Test_Data_in = din[14];
          #40 Test_Data_in = din[15];
          #40 Test_Data_in = din[16];
          #40 Test_Data_in = din[17];
          #40 Strobe_in_TData = 0;
     end
     endtask

     always
     begin
          #20 TCLK_in = 0;
          #20 TCLK_in = 1;
     end
     always
     begin
          #5 OSC_CLK = 0;
          #5 OSC_CLK = 1;
     end
     always
     begin
          #20 Cexternal = 0;
          #20 Cexternal = 1;
     end

initial
begin
$dumpfile ("support_sim3.dump");
$dumpvars (0, sim);
end // initial begin
endmodule
```

Vitae

Name :Mohammed Abdulqaher Ahmed Al-Asali

Nationality :Yemeni

Date of Birth :3/29/1986

Email :alasli86@yahoo.com

Present Address :KFUPM, Dhahran, Saudi Arabia

Permanent Address :Sana'a university, Sana'a-Yemen

Phone Number :+966542452803

Academic Background :BSc-KFUPM