HARDWARE IMPLEMENTATION OF COORDINATE ROTATION DIGITAL COMPUTER IN FIELD PROGRAMMABLE GATE ARRAY

MOHD ILYAS SOBIRIN BIN MOHD SAZALI

UNIVERSITI TEKNOLOGI MALAYSIA

HARDWARE IMPLEMENTATION OF COORDINATE ROTATION DIGITAL COMPUTER IN FIELD PROGRAMMABLE GATE ARRAY

MOHD ILYAS SOBIRIN BIN MOHD SAZALI

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical – Microelectronic and Computer System)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JANUARY 2012

To the next batch of students, *Lillaahi Ta'ala*.

ACKNOWLEDGEMENT

All Praise is to Allah through His Mercy and Favour the completion of this project among all good things is accomplished. In the past, people seek for respected scholars in the name of knowledge. Pursuing a Master's degree at Universiti Teknologi Malaysia (UTM) and in particular undertaking this project under the supervision of Professor Dr Mohamed Khalil Hani have been among the greatest decisions of my life. My sincere acknowledgement goes to my supervisor for his productive guidance and constructive criticisms in completing this project, as well as for his invaluable knowledge sharing, brilliant thoughts, wonderful experiences, wise words, and continuous support throughout my postgraduate time in UTM.

My gratitude is extended to Dr Muhammad Nadzir Marsono and Dr Usman Ullah Sheikh Izzat Ullah Sheikh for their valuable inputs as the panels during the project presentation. I am also grateful to my lab mates, Moganeshwaran, Annuar, Andy, Pei Chee, and Dr Rabia Bakhteri for their technical advices during the course of this project. Special thanks to Dr Hau Yuan Wen and Vishnu P. Nambiar for their technical tutorials that have aided to the success of this project, as well as to Sia, Shafiza, Sathy and Yee Hui for their assistance.

My genuine appreciation is dedicated to my parents, Dr Haji Mohd Sazali and Hajjah Shamsinar, along with my siblings for their incessant encouragement and prayers. Most of all, I am deeply indebted to my wife, Siti Nazmin, without whose kind understandings and tolerances neither this project nor my study would be successful. To our beloved sons, Iman Shahdan and Irshad Shahidin, for having to spend a part of their childhood time without their father, I owe them so much.

I also wish to express my gratitude to Universiti Sains Malaysia (USM) and Ministry of Higher Education (MoHE) for funding my study. Even though far from being perfect, it is hoped that this project report will be beneficial in one way or another to the reader.

ABSTRACT

Trigonometry is of great importance in mathematics as well as in physics, engineering, and chemistry. Astronomy, geography, navigation, study of optics and acoustics, oceanography, architecture, calculus, etc. are just several examples where trigonometry is significantly practiced. Historical figures like Pythagoras and Columbus used trigonometric tables in their careers. The birth of software has empowered relatively faster trigonometric functions performed by processors. In real-time applications though, such as trajectory calculations in military or space exploration, or in biomedical authentication system for fast access or rejection decision, trigonometric computation by software is a considerably time-consuming process. Coordinate Rotation Digital Computer (CORDIC) is an algorithm developed for hardware implementation as a real-time solution to trigonometric computation. This report presents a design approach to realize the CORDIC algorithm, prototyped as an embedded system in an Altera Field Programmable Gate Array (FPGA) development board running at 100 MHz clock frequency. The design flow applies the systematic Register Transfer Level (RTL) methodology, partitioning the design into a Datapath Unit (DU) for computation tasks, and a Control Unit (CU) for controlling the operation flow. Experimental results show that a high accuracy was obtained, with mean computation errors between 0.0014% and 0.0023% with respect to a software implementation on the same platform. The speed up in the execution time is about 89 times for the computation of cosine and sine functions, and 69 times for the arctangent. The work demonstrates the power of the CORDIC algorithm, and presents a methodology for an efficient complex hardware design.

ABSTRAK

Trigonometri amat penting dalam matematik serta fizik, kejuruteraan dan kimia. Astronomi, geografi, navigasi, kajian optik dan akustik, oseanografi, seni bina, kalkulus, dan lain-lain hanyalah beberapa contoh di mana trigonometri dipraktikkan dengan mendalam. Tokoh-tokoh sejarah seperti Pythagoras dan Columbus menggunakan jadual trigonometri dalam kerjaya mereka. Kelahiran perisian telah mempercepat pengiraan fungsi trigonometri oleh pemproses. Namun dalam aplikasi masa benar, seperti pengiraan trajektori dalam ketenteraan atau penerokaan angkasa lepas, atau dalam sistem pengesahan biometrik untuk akses atau penafian yang cepat, pengiraan trigonometri oleh perisian adalah suatu proses yang memakan masa terlalu lama. Komputer Putaran Koordinat Digital (CORDIC) adalah suatu algoritma dibangunkan khusus untuk implementasi perkakasan sebagai penyelesaian kepada pengiraan trigonometri dalam masa benar. Laporan ini membentangkan suatu pendekatan reka bentuk dalam merealisasikan algoritma CORDIC, diprototaipkan sebagai sebuah sistem terbenam dalam papan pembangunan *Field Programmable Gate Array* (FPGA) yang berfungsi pada frekuensi 100 MHz. Pendekatan ini menggunakan kaedah sistematik *Register Transfer Level* (RTL) dengan membahagikan reka bentuk kepada sebuah Unit Laluan Data (DU) untuk tugasan pengiraan, dan sebuah Unit Kawalan (CU) bagi mengawal perjalanan operasi. Keputusan uji kaji menunjukkan bahawa ketepatan tinggi telah diperolehi, dengan min ralat pengiraan antara 0.0014% dan 0.0023% berbanding dengan implementasi perisian dalam platform yang sama. Masa pelaksanaan adalah kira-kira 89 kali lebih pantas untuk pengiraan fungsi sinus dan kosinus, dan 69 kali untuk lengkuk tangen. Kerja ini menunjukkan kelebihan algoritma CORDIC, dan membentangkan suatu kaedah ke arah reka bentuk perkakasan kompleks yang efisien.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ALU	-	Arithmetic Logic Unit
ASIC	-	Application Specific Integrated Circuit
ASM	-	Algorithmic State Machine
CB	-	Computation Block
CC	-	Clock Cycles
CLB	-	Configurable Logic Block
CM	-	Concatenate Module
CMOS	-	Complementary MOSFET
CORDIC	-	Coordinate Rotation Digital Computer
CPU	-	Central Processing Unit
CS	-	Chip Select
CU	-	Control Unit
CV	-	Control Vectors
DFG	-	Data Flow Graph
DU	-	Datapath Unit
FBD	-	Functional Block Diagram
FPGA	-	Field Programmable Gate Array
FSM	-	Finite State Machine
HDL	-	Hardware Description Language
HLL	-	High Level Programming Language
HW	-	Hardware
IC	-	Integrated Circuit
IDE	-	Integrated Design Environment

IEEE	-	Institute of Electrical and Electronics Engineers
I/O	-	Input Output
IOB	-	Input Output Block
IOBD	-	Input Output Block Diagram
LCD	-	Liquid Crystal Display
LE	-	Logic Element
LHS	-	Left Hand Shift
LPM	-	Library of Parameterized Modules
LUT	-	Look Up Table
MCU	-	Microcontroller Unit
MHz	-	Mega Hertz
MOSFET	-	Metal Oxide Semiconductor Field Effect Transistor
MSB	-	Most Significant Bit
MUX	-	Multiplexer
NS	-	Next State
OS	-	Operating System
PCA	-	Pipelined CORDIC Array
PFM	-	Pipelined Filter Module
PLL	-	Phase Locked Loop
PS	-	Present State
RHS	-	Right Hand Shift
RTL	-	Register Transfer Level
RTL-CS	-	RTL Control Sequence
SRAM	-	Static Random Access Memory
SDRAM	-	Synchronous Dynamic Random Access Memory
SoC	-	System-on-Chip
SW	-	Software
USB	-	Universal Serial Bus
UTM	-	University of Technology Malaysia
VLSI	-	Very Large Scale Integration
VHDL	-	Very High Scale Integrated Circuit HDL

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This project report documents a prototype hardware design of Coordinate Rotation Digital Computer (CORDIC) algorithm implemented as part of a System-on-Chip (SoC) implemented in Field Programmable Gate Array (FPGA). This chapter provides an overview of the design abstraction level and the FPGA technology, followed by an introduction to CORDIC, problem statement, project objectives, scope of work, project contribution, and finally the thesis organization.

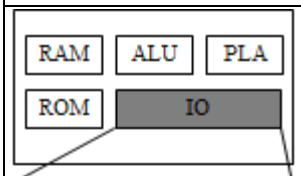
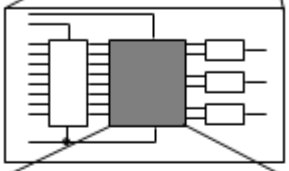
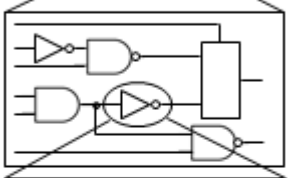
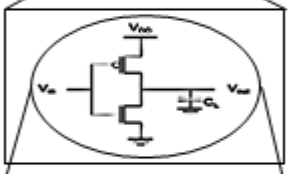
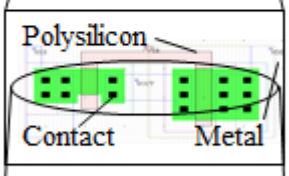
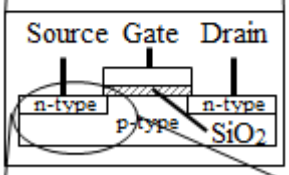
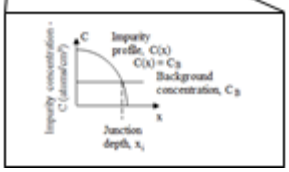
1.1 Design Abstraction Level

In order to handle different complexities in designing and fabricating an integrated circuit (IC), one of the techniques that electronic designers adopted is design abstraction [1]. A typical design abstraction can be arranged in a decreasing abstraction order as illustrated in Table 1.1, i.e. starting from the system or architectural level, moving down to the register transfer level (RTL), logic level, circuit level, layout level, device level, and finally the technology level. In a typical semiconductor company, each design level is managed by different engineering teams which may be situated at different parts of the world. At each level, the in-charged team models the design with a

black box view suitable for the complexity of that particular level, without having to worry for the internal details beyond their responsibilities or expertise. The information contained in the model is however adequate for the specialized team at the lower level of the design hierarchy.

One of the criteria which determine the design complexity is the number of gate counts. This project applies the RTL design level. The first reason is that the complexity of the proposed design is expected not to exceed 100,000 gates. For a larger system with a gate count of up to 500,000 gates, the system level would have been adopted. Else if the design is even simpler that it is expected to use 10,000 gates or lower, the logic or circuit level could be considered. Secondly, the RTL level is suitable for fast prototyping digital circuits into a Field Programmable Gate Array (FPGA) platform prior to transferring to more costly design stages. When a design has been successfully prototyped with relevant analysis carried out, then only it can be assigned to the succeeding levels, which are targeted for fabrication in an Application Specific IC (ASIC). As additional information, Table 1.1 also includes the courses offered at Masters Level by the body of knowledge, University of Technology Malaysia (UTM) under the Faculty of Electrical for each design abstraction level.

Table 1.1 : Design Abstraction Level [1]

Graphical view	Level	Primitive units	Concerned parameters	Courses offered in UTM
	System / Architectural Level	Behavioral modules	Silicon area	Adv. Computer Architecture (MEL1183)
	Register Transfer Level (RTL)	Functional modules	Timing	Adv. Digital System Design (MEL1173)
	Logic Level	Gates, Bits	Delays (propagation / transition)	Integrated Circuit Testing (MEL1133)
	Circuit Level	Transistors	Voltage, Currents	Analog CMOS Design (MEL1193)
	Layout / Physical Level	Layout layers	Topology, Dimensions	VLSI Circuits & Design (MEL1163)
	Device Level	MOSFET models	Current-Voltage Characteristics	Nanoelectronic Devices (MEL1113)
	Technology Level	Process models	Impurity Profiles	-

1.2 Field Programmable Gate Array (FPGA)

The Field Programmable Gate Array (FPGA) technology was pioneered by Xilinx in 1985 [2] as a prototyping platform for ICs. The function of an FPGA is user-configurable, i.e. defined by a user's program rather than the device manufacturer. An FPGA in principle is made up of three major configurable elements which are Configurable Logic Blocks (CLBs) or also called as Logic Elements (LEs), Input/Output Blocks (IOBs), and interconnections as illustrated in Figure 1.1. The CLBs supply the functional element for constructing a user's logic. The interface between external package pins and internal signal lines is provided by the IOBs. The programmable interconnection links the CLBs and IOBs into the relevant network. Among programmable switching technologies driving an FPGA are SRAM-driven pass transistors [2, 3], anti-fuses [4], and EPROM-driven pass transistors [5, 6]. The first generation of FPGA implements 4-input Boolean functions and has a single storage element [2]. Successive generations enabled wider Boolean functions and incorporated additional storage elements, with more powerful and flexible CLBs, as well as improved IOBs and interconnections, allowing the FPGA technology to dramatically reduce the design turn-around time and manufacturing costs [7].

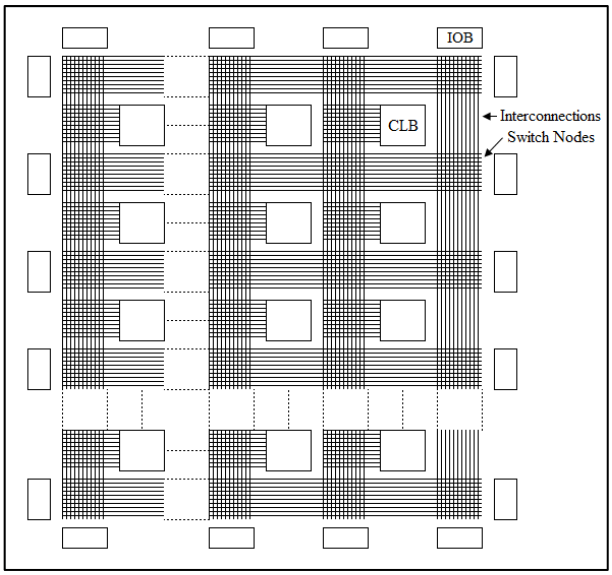


Figure 1.1 The architecture of a Field Programmable Gate Array [8]

1.3 Coordinate Rotation Digital Computer (CORDIC)

Trigonometry is one of the fundamental mathematical functions that are vigorously taught at high schools. Students initially learnt to use trigonometry book which contains look-up tables for all trigonometric functions. This method was later replaced by scientific calculators which university students and engineers are obligatory to own one, be it in the physical form or in the software form. Most people have no knowledge or little appreciation on how these calculators are capable to compute such mathematical functions in just a blink of an eye, or to what extent can the accuracy be. The underlying secret of this capability may lie in an algorithm named as Coordinate Rotation Digital Computer (CORDIC).

CORDIC is an iterative algorithm involving only additions, subtractions, simple bit shifts, and constants look-ups, developed for fast computation of trigonometric functions such as cos, sin, and $\tan^{-1}$, their hyperbolic counterparts i.e. cosh, sinh, and $\tanh^{-1}$, as well as elementary operations like square root, division, and multiplication. Even though it is an approximation approach, the results of a CORDIC operator do not compromise accuracy, with a higher number of iterations provides a higher precision with respect to the actual calculation. The design of a CORDIC module involves division by powers of two, which can be exploited in a hardware implementation by applying right-hand shift (RHS) operation. In a compute-intensive application, CORDIC is one of the preferred alternatives to compute the previously mentioned mathematical functions.

The CORDIC algorithm is credited to Jack E. Volder in 1959 [9], who worked at Convair, an American company which started as an aircraft manufacturer and later expanded to produce rockets and space crafts. Compared to their other products, the CORDIC algorithm was just a small contribution but with significant effects in application areas such as navigational systems for calculating real-time trajectories; biometrics in the image processing module for fingerprint minutiae matching [10];

telecommunications such as in the design of digital down converters [11] and radar signal processors [12]; the HP-35 calculator and many other examples [13]. The original algorithm was meant for the computation of trigonometric functions, multiplication, and division operations. It was John Walther who has actually generalized this algorithm in 1971 [14] for hyperbolic, logarithm, and exponential computations [15].

1.4 Problem Statement

Software solutions adopted by microprocessors to perform trigonometric functions are compute intensive, time-consuming and not suitable for direct hardware implementation [13]. CORDIC is one of the hardware algorithms developed to solve trigonometric, hyperbolic, and linear functions due to its simplicity and speed efficiency [16]. Designing a CORDIC module has been set as the purpose of this project.

1.5 Objectives

The project aims to develop a CORDIC software (SW)/hardware (HW) coprocessor to be implemented in FPGA technology applying the RTL design methodology. The sub-objectives are:

i. To design and simulate a CORDIC HW core consisting of a Control Unit (CU) and a Datapath Unit (DU).
ii. To design and simulate a system bus interface module for the control and data transfer between the Central Processing Unit (CPU) and the CORDIC core.
iii. To integrate the interface module and the CORDIC core into a top-level module, CORDIC coprocessor. This coprocessor is then implemented in a FPGA.

iv. To develop an embedded SW in order to activate the functionality of the CORDIC coprocessor. This SW is then integrated with the coprocessor and running in the FPGA.

v. To compare the performance of the CORDIC coprocessor design with SW library functions as well as previous works, in terms of accuracy, execution time, and resource utilization.

1.6 Scope of Work

The scopes of this thesis are:

i. The CORDIC coprocessor is modeled, synthesized, and simulated in Quartus II Version 8.1 Build 163 10/28/2008 SJ Web Edition running on Windows 7 Operating System (OS), as well as Quartus II Version 9.0 Build 132 02/25/2009 SJ Full Version running on Linux Ubuntu OS. The reason for employing two different versions is that the former one is for home and initial design usage while the latter is for laboratory work purposes of this project.

ii. Nios II Integrated Design Environment (IDE) Version 9.0 Build 132 2003 is utilized for the development of the embedded SW and execution of the CORDIC coprocessor design by the Nios II CPU.

iii. Verilog Hardware Description Language (HDL) is applied for the HW design and synthesis, and C High Level Programming Language (HLL) is used to program the embedded SW.

iv. The CORDIC coprocessor is designed to be the first working prototype to implement the trigonometric functions of cos (θ), sin (θ), and $\tan^{-1}$ (y/x) only. The implementation of other functions such as cosh (θ), sinh (θ), square root, etc. are recommended for future work.

v. This working prototype is executed in Altera Cyclone II EP2C35F672C6 FPGA DE2 development board. The board is equipped with a maximum clock

frequency of 50 Mega Hertz (MHz), but with an inclusion of a Phase Locked Loop (PLL), the coprocessor design is executed at 100MHz.

1.7 Project Contributions

i. A prototype of a trigonometric computer is developed in a FPGA-based platform.
ii. A systematic digital design technique to realize a SW-HW coprocessor in FPGA is presented.

1.8 Report Organization

This report is organized into six chapters. Chapter 1 introduces the preliminary information of the project, the problem statement, project objectives, scope of work, and project contributions to the body of knowledge.

Chapter 2 is the background and literature review chapter. This chapter provides an insight to the CORDIC algorithm and reviews some previous related work on its implementation in FPGA platform.

Chapter 3 presents the methodology employed in the project. It begins with the overall project flow and continues with the RTL design flow applied in the CORDIC HW development. The chapter ends with the development flow of the embedded SW.

Chapter 4 is dedicated for the HW design of the CORDIC coprocessor. This includes the explanation of the involved HW cores and the system bus interface module.

Chapter 5 is the results and discussion chapter. Snapshots of Nios II console window displaying the outputs of trigonometric functions executed are presented in this chapter. The results of the CORDIC coprocessor are compared with C trigonometric library functions and a couple of previous works in terms of accuracy, execution time, and resource utilization involved.

Chapter 6 concludes the project outcome and provides recommendations for future work to further improve the functional prototype of the CORDIC coprocessor.

REFERENCES

1. Mohamed Khalil Hani. *Starter's Guide to Digital Systems VHDL & Verilog Design*. 2nd Edition (Revised Edition 2.4). Prentice Hall. 2011.
2. Hsieh, H. C., Carter, W. S., Ja, J., Cheung, E., Schreifels, S., Erickon, C., Freidin, P., Tinkey, L., Kanuzawa, R. Third-Generation Architecture Boosts Speed And Density Of Field-Programmable Gate Arrays. *Custom Integrated Circuits Conference*. IEEE. 1990. 31.2/1 – 31.2/7.
3. Hsieh, H. C., Dong, K., Ja, J. Y., Kanazawa, R., Ngo, L. T., Tinkey, L. G., Carter, W. S., Freeman, R.H. A 9000-gate User-Programmable Gate Array. *Custom Integrated Circuits Conference*. IEEE. 1988. 15.3/1 – 15.3/7.
4. El Gamal, A., Greene, J., Reyneri, J., Rogoyski, E., El-Ayat, K.A., Mohsen, A. An Architecture for Electrically Configurable Gate Arrays. *IEEE Journal of Solid-State Circuits*. IEEE. 1989. 24(2): 394 – 398.
5. Gupta, A., Aggarwal, V., Patel, R. Chalasani, P., Chu, D., Seeni, P., Liu, P. W. Wu, J. Kaat, G. A User Configurable Gate Array using CMOS-EPROM Technology. *Custom Integrated Circuits Conference*. IEEE. 1990. 31.7/1 – 31.7/4.
6. El-Ayat, K. A., El Gamal, A., Guo, R., Chang, J., Mak, R. K. H., Chiu, F., Hamdy, E. Z., McCollum, J., Mohsen, A. A CMOS Electrically Configurable Gate Array. *IEEE Journal of Solid-State Circuits*. IEEE. 1989. 24(3): 752 – 762.
7. Lai, Y. T. and Wang P. T. Hierarchical Interconnection Structures for Field Programmable Gate Arrays. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. IEEE. 1997. 5(2): 186 – 196.

8. S. Brown and J. Rose. Architecture of FPGAs and CPLDs: A Tutorial. *IEEE Design and Test of Computers*. 1996. 13(2): 42 – 55.
9. Volder, J. E. The CORDIC Trigonometric Computing Technique. *IRE Trans. Electronic Computers*. 1959. 8(3): 330 – 334.
10. Fons, F., Fons, M., Canto, E., and Lopez, M. (2006). Trigonometric Computing Embedded in a Dynamically Reconfigurable CORDIC System-on-Chip. In Bertels, K., Cardoso, J. M. P., and Vassiliadis, S. (Eds): ARC 2006, LCNS 3985 (pp. 122 – 127). Berlin/Heidelberg: Springer-Verlag.
11. Löhning, M., Hentschel, T., and Fettweis, G. Digital Down Conversion in Software Radio Terminals. *EUSIPCO*. 2000. 4 – 7.
12. Andraka, R. Building a High Performance Bit-Serial Processor in an FPGA. *Proc. Of Design SuperCon '96*. 1996. 5.1 – 5.21.
13. Andraka, R. A Survey of CORDIC Algorithms for FPGA based Computers. *Proc. 6th International Symposium on FPGAs.* 1998. 191 – 200.
14. Walther, J. S. A Unified Algorithm for Elementary Functions. *Proc. AFIPS Spring Joint Computer Conference*. 1971. 379 – 385.
15. Neji, N., Boudabous, A., Kharrat, W., Masmoudi, N. Architecture and FPGA Implementation of the CORDIC Algorithm for Fingerprints Recognition Systems. *8th IEEE International Multi-Conference on Systems, Signals and Devices (SSD).* IEEE. 2011. 1 – 5.
16. Vadlamani, S., and Mahmoud, W. Comparison of CORDIC Algorithm Implementations on FPGA families. *IEEE Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory*. IEEE. 2002. 192 – 196.
17. Meher, P. K., Valls, J., Juang, T., Sridharan, K., and Maharatna, K. 50 Years of CORDIC: Algorithms, Architecture, and Applications. Regular Papers. *IEEE Transactions on Circuits and Systems*. IEEE. 2009. 56(9): 1893 – 1907.
18. http://babbage.cs.qc.cuny.edu/IEEE-754.old/
19. Liu, J., Wang, S., Li, Y., Han, J., and Zeng, X. Configurable Pipelined Gabor Filter Implementation for Fingerprint Image Enhancement. *10th IEEE International Conferences on Solid-State and Integrated Circuit Technology (ICSICT)*. IEEE. 2010. 584 – 586.