Arithmetic Operations On Elliptic Curve Defined Over Radix-7 Symmetrical Quaternary Signed -Digit Finite Fields And Their Application In Secure Communication Systems.

DAIKPOR, Michael Naseimo

M. Sc (Ukraine-USSR)

A thesis submitted to the School of Postgraduate Studies , **University of Lagos** in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy

Department of Electrical and Electronics Engineering

University of Lagos, Nigeria

NOVEMBER 2010

SCHOOL OF POSTGRADUATE STUDIES
UNIVERSITY OF LAGOS

CERTIFICATION

This is to certify that the Thesis:

"ARITHMETIC OPERATIONS ON ELLIPTIC CURVE DEFINED OVER RADIX-7 SYMMETRICAL QUATERNARY SIGNED-DIGIT FINITE FIELDS AND THEIR APPLICATION IN SECURED COMMUNICATION SYSTEMS"

Submitted to the
School of Postgraduate Studies
University of Lagos

For the award of the degree of

DOCTOR OF PHILOSOPHY (Ph.D.)

is a record of original research carried out

By:

DAIKPOR, MICHAEL NASEIMO

In the Department of Electrical & Electronics Engineering

DAIKPOR, MICHAEL N.		29/10/10
AUTHOR'S NAME	SIGNATURE	DATE
PROF OLUWOLE ADEGBENRO		29 10 10
1ST SUPERVISOR'S NAME	SIGNATURE	DATE
PROF. IKE MOWETE		29/10/2010
2ND SUPERVISOR'S NAME	SIGNATURE	DATE
3RD SUPERVISOR'S NAME	SIGNATURE	DATE
Dr T.O. Akinbulire		29/10/10
1ST INTERNAL EXAMINER	SIGNATURE	DATE
Prof. J.A. Olaofe		29/10/2010
2ND INTERNAL EXAMINER	SIGNATURE	DATE
DR. O.A. FAKOLUJO		29-10-2010
EXTERNAL EXAMINER	SIGNATURE	DATE
Dr S.A. Okunoye		29-10-10
SPGS REPRESENTATIVE	SIGNATURE	DATE

ACKNOWLEDGEMENT

This work was done at the Department of Electrical and Electronics Engineering University of Lagos, Nigeria. I say a big thank you to the Department of Electrical and Electronics Engineering University of Lagos, Nigeria.

My sincere gratitude goes to my Lead Supervisor Professor Oluwole Adegbenro. Words cannot be invented to express my regard and admirations for his steadfastness, in the face of many odds and even at the most turbulent periods will continue to be an inspiration for me in life. Though it was suppose to be just a supervisor/student relationship but he went the extra mile. He sees no boundary in coming to my aid whenever and wherever it is necessary even when not solicited for. He is certainly a role model for me and my prayer is that I meet his expectations in this career to justify the very robust: academic, social, moral and professional investment he has put into this my training.

I also acknowledge my other supervisors: Professor Alex Ike Mowete and Dr. Philip. B Osofisan for their guidance and encouragement.

My appreciation go to Professors C.A. Awosope and C.C. Okoro for their friendly advice. I cannot fail to mention Professor Frank N. Okafor for all the support given me and Dr. T.O Akinbulire for his diligence in sending my papers to PG School and organizing the seminars. I thank Dr. A Ayorinde and Dr. D.O Osunde.

I gratefully acknowledged the support from my fellow staff Ph.D students: Engr. Isa Mohammed, Engr. M.A.K Adelabu, Engr. (Mrs.) Abdul salam, Engr. (Mrs) Gbenga-Ilori, Engr. (Mrs) F. Olobaniyi, Engr. S. Adetona and Engr. P.O. Oluseyi

There are other numerous friends and colleagues inside and outside the Department, Faculty, the University and my family whose goodwill, prayers and assistance during this work are inestimable. My wife, Mrs Lilian Buluobere Daikpor and my children: Ebimoboere, kememopigha, Tamara denyeifa, Tamarakuro, Tamarameimene, Edeseimi, and Peremoboere who bore the brunt of a husband that never stay at home and a father that can only take them home very late in the night.

Above all I give thanks to God Almighty, The God of Abraham, Jacob and Moses; the El-Shadai. The Lord of Host, from whom all good things come.

Michael Naseimo Daikpor

Lagos, Nigeria

DEDICATION

To GOD Almighty

ABSTRACT

This thesis proposes a Restricted Radix-7 Symmetrical Quaternary Signed Digit Number system (RR7SQSDNS) both as an alternative processing logic threshold system and as a signed digit algebraic structure. The thesis also presents the arithmetic operations on elliptic curves defined over this special type of finite field and develops Restricted Radix-7 Symmetrical Quaternary Signed Digit (RR7SQSD) modular exponentiation scheme that uses a multiply-by-7 addition/subtraction chain to compute the product of very large word length integer operands in data word length constrained compute environments. The multiply-by-7 addition/subtraction chain is also used to develop a one-stop multiply-by-7 point multiplication computation strategy for elliptic curves defined over RR7SQSD finite fields.

A Complementary Pass (CP) gate derived 7-valued symmetrical quaternary signed-digit T-gate realized as a basic building block for implementing the SMVL arithmetic unit VLSI circuit operates on RR7SQSD profile voltage signals. The VLSI functional logic circuits were thus synthesized on this basic building blocks level. Multisim Electronic Work Bench was used to analogue simulate the T-gate's MOSFET circuit. The T-gate's multiplexing ability, the RR7SQSD full adder and quasi-multiplier circuits' accuracy were simulated using qbasic language hardware-descriptive-styled source code program with results that confirmed the efficiency of RR7SQSD arithmetic.

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

ABBREVIATIONS

ABREVATION	MEANING
AES	Advanced Electronic Signature
ALU	Arithmetic and Logic Unit
ANSI	American National Standard Institute
ASIC	Application Specific Integrated Circuit
ATM	Automated Teller Machine
BDCM	Bi-Directional Current Mode
CH	Chapter
CMOS	Complementary Metal-Oxide Semiconductor
CP	Complementary Pass
DBNS	Double Based Number System
DES	Digital Electronics Signature
DFT	Discrete Furrier Transformation
DLP	Discrete Logarithm Problem
DRAM	Double Random Access Memory
DSA	Digital Signature Algorithm
EC	Elliptic curve
ECC	Elliptic Curve Cryptography
ECDHA	Elliptic Curve Diffie-Hellman Algorithm
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECL	Emitter Coupled Logic
EEA	Extended Euclidean Algorithm
FA	Full Adder
FEAM	Field Element Adder-Multiplier
Fig	Figure
FPGA	Field Programmable Gate Array
FPLSC	Field Programmable Level System Circuit
HA	Half Adder
ICTS	Information and Communication Technology Systems
IEEE	Institute of Electrical and Electronics Engineers
IFP	Integer Factorization Problem
ISO	International Standard Organisation
ISP	Internet Service Provider
LSI	Large Scale Integration
LUT	Look Up Table
MIPS	Millions of Instructions Per Second
MOSFET	Metal-Oxide Semiconductor Field Effect Transistor
MVL	Multiple-Valued Logic

ABBREVIATIONS

N/A	Not Available
NAF	Non-Adjacent Form
NIST	National Institute for Standard and Technology
OS	Operating System
P[N] MOS	P [N] Metal Oxide Semiconductor
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PKC	Public Key Cryptography
POS	Palm held Operating System
RFID	Radio Frequency IDentification
RNS	Residue Number System
RR7SQSD	Restricted Radix-7 Symmetrical Quaternary Signed Digit
RRR7SQSDNS	Restricted radix-7 Symmetrical Quaternary Signed-digit Number System
RSA	Rivest, Shamir and Adleman
SD	Signed – Digit
SDFA	Signed Digit Full Adder
SDNS	Signed Digit Number System
SET	Secure Electronic Transaction
SHA	Secure Hash Algorithm
SMVL	Symmetrical Multiple-Valued Logic
SSH	Secure Shell
T_ULM	Tree-type Universal Logic Module
USN	Ubiquitous Sensor Network
VLSI	Very Large Scale Integration
WAP	Wireless Application Protocol
XOR	Exclusive-OR

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 THE BACKGROUND OF STUDY

As today's human society becomes more and more information technology driven the integrity of received information continue to be a very important security issue. Communication among community of information users is said to be secured and trusted if it can be proved that the received intelligence at the authorized receiver's end is original, un-intercepted or re-transmitted and un-adulterated by an un-authorized party. With most communication channels open and accessible to all and sundry, the desired security of transmitted message can only be provided at the sending end while authentication can only be performed at the receiving end. Cryptography is one of the several means of making transmitted plaintext messages secure over open and insecure communication channels and provide facilities for authentication at the receiving end. Cryptography transforms the plain text message into an obscure form which can only be reconstructed by the intended recipient in record time.

1.1.1 Classical cryptography

At the beginning there were just two basic techniques of cryptography: substitution cryptography (where letters of the original plain text are replaced by other letters) and the transposition cryptography (in which letters of the plain text are re-arranged in a different order). Concerned parties agree on a particular technique and exchange single secret keys that would be used for encryption and decryption of plain text messages. Such cryptosystems ciphers were either monoalphabetic (that is only one substitution/transposition) or polyalphabetic (characterized by a combination of several

substitution/transposition) and concatenation techniques were used. These were the symmetric cryptosystems and they require extreme strict key exchange management to ensure the desired security level [1].

According to available records classical cryptography dates back to 400 BC in the city of Menet Khufu in Egypt, in the form of hieroglyphic inscriptions on tombs while Julius Ceaser extensively used substitution ciphers between 100 to 44 BC [2]. Ever since then, cryptography has continued to flourish in governance, diplomacy, wars and in all other human endeavours. Some of the ubiquitous examples in history include: the sensational "Babington plot" of 1586 in which Mary Stuart of Scotland plotted to dethrone and kill Queen Elizabeth I of England; the infamous "1917 Zimmermann Telegram" that contained the top secret plan of Germany to invade the United States of America during World War I (WWI) and the Japanese top secret diplomatic code on the plans to attack Peel harbour during World War II (WWII). Cracking of these ciphers were not only historical but were also academic achievements in their times. History also has it that America's uncertainty concerning the authenticity of the deciphered message and thus her delay in taking counter measures cost her loss of over 3000 sailors and Airmen along with several hundreds of warships and planes in one single day. This underscores the need for fast and secure information delivery systems. Today proper functioning of the Information and Communication Technology Systems' (ICTS), the secret intelligent service, crime detection, e-banking, e-commerce, working wireless, internet shopping, e-payment, pay television broadcasting, satellite image processing and communication,

etc, is based on fast secure and authenticable information delivery systems.

1.1.2 Elliptic curve cryptography: - corner stone and future of modern cryptography

Modern cryptography made popular and open by White Diffie evolved along a completely different direction. It is public-key-oriented and based on the practically-reverse-operation intractability of some mathematical hard problems [3]. These hard problems typically have forward operations that are relatively easy to perform and reverse operations that are computationally infeasible even with all available cutting edge computing systems working in parallel. Such hard problems are the: Integer Factorization Problem (IFP), index calculus or Discrete Logarithm Problem (DLP) and the Elliptic Curve Discrete Logarithm Problem (ECDLP). They formed the basis of today's acclaimed cryptographic protocols such as: IFP in Rivest, Shamir and Adlemans (RSA) cryptosystem protocol, the DLP in the Diffie Hellman's Digital Signatures Algorithm (DSA) and the ECDLP in Elliptic Curve Cryptography (ECC) [4-16]. Communicating parties maintain a pair of keys each: a private key and a public key. The private keys are selected randomly and independent of the other party and used to compute the public keys. The public keys are made public and an intending message sender uses the public key of the intended receiver and his private key to encrypt the plain text message using an agreed hard mathematical algorithm and transmit the cipher over a public domain (an insecure) channel. The message can be deciphered only by the intended recipient. This type of cryptosystem is referred to as asymmetric or Public Key Cryptography (PKC). It ensures data integrity, provide

authentication and does not repudiate hence a secure and trusted communication. Elliptic curves are mathematical constructs with complex addition and point multiplication arithmetic operations. Today there exist a large variety of improved Elliptic Curve (EC) arithmetic techniques for performing integer factorization, primality testing, pseudorandom number generation to mention a few. However, the major arithmetic operation in Elliptic Curve Cryptography (ECC) is EC point multiplication [4 -7]. EC point multiplication is essentially scaling a point $p=(x,y)$ on the elliptic curve with a scalar quantity m, in such manner that the product $W=mP=(x_w,y_w)$ is also a point on the same elliptic curve. Consequently, point multiplication is often referred to as EC scalar multiplication or modular exponentiation since EC point multiplication is equivalent to exponentiation in modular arithmetic [8 - 13]. The procedure for computing EC point multiplication requires two other basic elliptic curve arithmetic operations namely: elliptic curve points addition and elliptic curve point doubling. These basic elliptic curve arithmetic operations are in turn executed using finite field addition, multiplication, squaring and inversion operations. Finding W, from known values of P and m is easy; but finding m from given values of W and P is a mathematically exponentially hard problem. This problem, often referred to as the Elliptic Curve Discrete Logarithm Problem (ECDLP) [9 - 10], is the kernel of Elliptic Curve Cryptography (ECC) and the foundation basis of modern cryptography.

1.1.3 The emerging secure and trusted communication devices and standards

Most emerging secure and trusted communication devices such as: pocket size laptop computers, smart cards, mobile or cell phones, Personal digital Assistants (PDA), Ubiquitous Sensor Networks (USN), pagers, Palm held Operating Systems (POS), digital post markers, internet service providers, radio frequency identification (RFID) tags and environmental sensors, as well as the Automated Teller Machines (ATM) and other critical service delivery systems for all e-banking, e-commerce and e-shopping have become routinely supported by ECC [17-22]. These devices are required to execute millions of computer instructions in a matter of seconds in special environments where there are limitations to the system's computing power, memory space, bandwidth and power supply (battery) [3,19-21]. Many of these devices are capable of handling voice, data and video images (3G) over electronic data communication channels and connect to the internet as terminals. This novel technological advancement further highlights the need for an improved secure and trusted communication system. The traditional cryptosystem algorithms (DES, DLP, RSA, and DSA) have become impractical as they require bigger memory and intensive power hungry big integer computation co-processors to complete calculations in timely manner. RSA comparatively slow but an acclaimed powerhouse of today's e-commerce, e-banking, e-purchase will have its key length doubled before the start of the next decade in order to sustain the present security level. The increase in key size will render RSA crypto-systems unsuitable for constrained environment devices. EC based crypto server security arrangement will further enhance internet security and allow Internet Service Providers (ISP) to utilize fewer crypto servers to provide secure gate way operations. As ECC becomes ICTS

only security tool globally, e-transactions traffic will be on the rise and servers will be have to perform signature generation, signature verification and key exchange operations using standard elliptic curves. Although present ECC supported communication standards include ANSI X9.63, IEEE 1363.2000, IETF RFC 3278, ISO 15946-3 and NIST SP 80056 [3], none-the-less there will equally be users with non-standard curves but meet prescribed parameters. For example elliptic curves defined over non-conventional number system or algebraic structures such as the RR7SQSDNS

element finite field. Certificating authorities must have to tolerate and servers must have to handle them.

1.2 STATEMENT OF PROBLEM

Despite the prominent role cryptography has come to play in transforming today's secure communication gadgets- 21st century wonders, they are still beset with conceptual, technological and operational challenges that need urgent attention. The problem statement of this study is formulated by considering these challenges and certain other foci that help to define the boundary conditions.

1.2.1 The challenges of secure communication gadgets

1.2.1.1 The conceptual challenge

This challenge is as a result of the variations in the degree of hardness of the mathematical problems and the vanishing gap between forward and reverse computation times. The hardness of the mathematical problems is a function of the length of the operands or the key size (number of decimal digits or bits) in the

mathematical operations; thus the longer the key size the better the security. Also key sizes for minimum security requirements vary with the type of cryptosystem protocol but because ECDLP hardness is exponential with key size unlike the sub exponential hardness for non-EC based PKC systems such as RSA and DSA, ECC provides minimum key size for the same level of security and confidentiality of the transmitted message. Table 1.1 gives the minimum security level per key size and period to break a corresponding cipher text for EC and non-EC cryptosystems. It can be seen that while a 210 bits key size in ECC requires 10^{20} MIPS-years to break, ten times this number of bits is required by non-EC cryptosystems for the same code breaking period.

Furthermore, under normal circumstances bigger processing power is required to calculate the inverse function of a PKC hard mathematical problem. However, as microprocessors increasingly become faster and of bigger memories yet smaller and cheaper, the time to solve the reverse problem reduces and approaches that of the time to solve the forward problem. This in effect reduces the security level of the cryptosystem.

Today's sufficient and adequate minimum security level key sizes may be inadequate tomorrow. To continually maintain the same acceptable minimum security level, key sizes will have to creep upward but so are clock speeds of processors increasing daily. It is thus, a "race between guns and armour" [17] that may not end. Notwithstanding, even if microprocessor and DRAM clock speeds will continue to increase as human

activities become more and more ICTS driven, it is necessary to find some means of widening the gap between forward and reverse operation computation times.

Table 1.1 Comparative code breaking period

Time to break cipher(MIPS- years)	Minimum key size (bits)		
	ECC	Non-ECC	Ratio
10^4	106	512	1:6
10^8	132	768	1:6
10^{11}	163	1024	1:7
10^{20}	210	2048	1:10
N/A	256	3072	1:12
N/A	384	7680	1:20
N/A	512	15360	1:30
10^{78}	600	21000	1:35

1.2.1.2 The Technological challenge – a binary processing logic threshold problem

This challenge is due to semiconductor device leakage current and in-circuit heat generation as a result of technological procedure involved in the manufacture of binary logic VLSI circuits. The binary logic system is a throw-back of Aristotle's law of excluded middle ("0" for the past, "1" for the present). This law though made popular by George Boole in 1854 had nothing for the 'future' but, in 1965 Jan Lukasiewcz proposed the Multiple-Valued Logic (MVL) to address the deficiency in Aristotle's

work called "Art of discourse''. MVL is higher radix r $(r > 2)$ **processing logic threshold none-the-less only binary logic Very Large Scale Integrated (VLSI) circuits continued to dominate the mind of electronics system designers up to the early part of this decade. This late interest can be attributed to the ease with which the semiconductor device used in building binary logic can be biased to operate on the two levels of "0"and "1". Today the field of semiconductor devices is one of the fastest developing areas as microelectronics records tremendous development with the ever-increasing frequency of occurance continuously enhancing the performance of integrated circuits. New materials, innovative processes, novel device structure, better modelling and simulation techniques are announced virtually on daily basis. The construction of powerful microprocessors is made possible by the astonishing break through in Complementary Metal Oxide Semiconductor (CMOS) VLSI circuit technology. In order to maintain consistency with Moore's law [23 - 27] the microelectronics industry cramps billions of transistors into a small area by physical down scaling of the transistor size.**

However, the increasing demand for higher performances in the emerging secure and trusted communication system puts a heavy burden on semiconductor device technology. Down scaling reduces wire capacitances at the sub micron level, below sub-micron level system inter wire capacitances increases greatly and becomes problematic both for power consumption and signal propagation delay [26 - 28]. By continued down scaling of the transistor, the gate-oxide region of Metal Oxide Field Effect Transistors

(MOSFET) shrinks and thus, causing gate leakage current flow. The presence of gate leakage current causes degradation in drain-source current quality which in turn adversely affects circuit performances. Also with several millions of transistors crammed into a small area, the bulk of individual semiconductor devices vanishes and the heat generated as a result of the numerous high-speed switching devices need to be quickly and effectively evacuated. Other problems with binary logic systems include: difficulty in realizing high speed array multipliers, complex device layout and large interconnection area [27]. Binary logic systems are also known for low information content per line, relatively low data processing capability [28], reduced reliability, and long carry propagation chain during arithmetic operations due to the small number of processing logic thresholds "0" **and** "1" **[29].**

In practical terms, these problems in binary logic VLSI circuits transform into bigger surface chip area, longer waiting or turn-around time at the ATM, longer delay time in making and receiving calls with the mobile phone and higher connection error rate. The implication is that the binary CMOS VLSI circuit at very high level of semiconductor devices integration is approaching a limit of performance and there is now a scramble for MOS VLSI circuit performance boosters in the semiconductor industry [21]. It has been predicted [24] that by the year 2016 the present binary logic CMOS VLSI circuit would be extinct. One such performance boosters is the MVL system which has been proved to provide a better insight into digital logic processing by displaying phenomena that are never possible in binary logic system. Working in binary is almost similar to

working with black and white colour. Whereas working with the full spectrum of white light improves performance if one uses optical analogy.

Consequently, for a long term solution the international scientific community is now directing its efforts on three major areas namely, i) to develop new materials for VLSI circuits, ii) to find alternative system architecture and iii) to formulate a new processing logic level threshold values [24 - 25]. The last of these three areas is also the focus of this study. Finding alternative processing threshold logic with high enough information per line capacity that will render semiconductor devices continued down scaling a non-issue to tomorrow's secure and error-free communication system's VLSI circuit designer.

1.2.1.3 The operational challenges

A properly implemented cryptosystem is a good security mechanism for governance, defence, commerce and industry. However, absolutely secure and trusted communication in practical terms is utopian. Information can only be theoretically secure and the cryptosystems are just hard to crack or are based on some computational notion of security. That is, cryptography-based service delivery systems like any other critical service delivery system also fails, though seldom publicized, such failures, however minutiae, may involve either loss of human lives or loss of very large sum of money. Studies of bank ATMs [30] showed that ATM failures, to a very large percentage, are as a result of the software implementation of the cryptosystem. Examples of failures in the banking and other customer-driven sectors include: i) banks issuing PIN codes through their branches and bank employees gaining access to

customers PIN number, ii) Fake ATM or service Engineers modifying ATM with a view to recording customers account numbers and PIN iii) programming errors resulting in the issuance of the same PIN numbers to several customers. The conclusion of the study of bank ATM is another grey area in this study. A hardware-implementable cryptosystem protocol algorithm will no doubt reduce frauds in ATM and the unthinkable damages incurred when secure communication and other critical service delivery systems fail.

1.2.2 Foci of this work

Arising from section (1.2.1) are the following three foci of the study.

i) Already there is a deluge of publications on various theoretical studies and practical design implementations of PKC arithmetic operations but they are all based on binary CMOS VLSI circuit techonolgy. Hence, the MVL system is considered as an appropriate alternative processing threshold logic.

ii) The Signed Digit Number System (SDNS) and the Residue Number Systems (RNS) both supporting efficient arithmetic, are two MVL processing threshold number systems. The RNS, though presents absolute carry-free arithmetic but because of the exigencies of complex conversion procedure had to be dropped for the SDNS which limits carry propagation one position to the left.

iii) ECC which provides an acceptable security level and confidentiality of transmitted message for a comparatively small key size was taken as the appropriate cryptosystem for this study.

The choice of SDNS and the ECC require the need for an appropriate algebraic structure in which the related signed digit arithmetic can be executed. Hence, the statement of the problem of this thesis is to investigate the arithmetic operations on EC defined over RR7SQSD finite fields and examine implementation issues of an ECC arithmetic unit VLSI circuit that will sustain the present human society increasing reliance on secure communication and information technology system gadgets. These foci facilitated the formulation of the following aims and objectives.

1.3 AIMS AND OBJECTIVES

1.3.1 The aim

The aim of this work is to develop a Symmetrical Multiple Valued processing Logic (SMVL) based elliptic curve arithmetic unit VLSI circuit for secure communication systems.

1.3.2 The Objectives

The objectives of this study are to:

i) **Formulate an appropriate strategy for performing Restricted Radix-7 Symmetrical Quaternary Signed Digit (RR7SQSD) based big integer operands parallel addition and multiplication operations.**

ii) **Establish the necessary basis of efficient arithmetic operations on EC defined over RR7SQSD element finite fields suitable for secure communication systems and prove that RR7SQSD element field based ECC is message-degradation and repudiation free using a particular cryptosystem protocol as a case study.**

iii) **Establish the architectural organization of the RR7SQSD based ECC arithmetic VLSI circuit.**

iv) **Design the VLSI circuit's basic building block and derive the exact LSI circuit diagram of the main functional logic units using the basic building block.**

1.4 SCOPE OF STUDY

The research study covers:

i) **The principle of the RR7SQSD arithmetic and methods of developing appropriate schemes for performing big integer arithmetic operations in compute constrained environments.**

ii) **Verification of the RR7SQSD number system as an algebraic structure or field suitable for cryptosystem application and suggestion of appropriate strategies for the realization of the relevant cryptosystem arithmetic operations.**

iii) **Encryption and decryption procedures of plain text messages on tested cryptosystem protocols on EC defined over RR7SQSD finite fields.**

iv) **Experiments /simulation procedures for testing the VLSI basic building block's circuit and the functional logic unit's conformity with design specifications i.e. the electrical parameters, correct function and operational accuracy.**

1.5 SIGNIFICANCE OF STUDY

The practical significance of this study is that, it will provide a basis for the design and implementation of MVL cryptosystems based secure and trusted state-of-the-art communication gadgets. In addition, the hitherto only software based cryptosystem protocols can be converted to hardware implementable algorithms thereby reducing failures and frauds in cryptographic based customer service delivery systems such as the ATM.

1.6 RESEARCH QUESTIONS

The pertinent research questions are:

i) **Which type of strategy can be employed to ensure fast and error-free RR7SQSD arithmetic on integer operands with the cryptographic significant length of about 160 to 512 decimal digits?**
ii) **What should be the description of an EC defined over RR7SQSD finite field that will guarantee message degradation and repudiation-free encryption and decryption operations in an RR7SQSD based ECC?**
iii) **What should be the constituent functional subunits of the RR7SQSD based ECC arithmetic unit VLSI circuit and the design specifications of the basic building block?**

iv) **Which engineering tool(s) should be best employed for the several simulation experiments?**

1.7 DEFINITION OF TERMS

Plaintext: This is the original intelligible message or unencrypted text.

Cipher Text: This is the end product of encrypting a plaintext.

Code: This is an algorithm for transforming an intelligible message into an unintelligible form.

Discrete Logarithm Problem: Given an element of a group, find the power n of the group generator g, such that g to the power of n, is an element of that group.

Elliptic Curve Cryptography: This is a type of cryptography based on group theory that provides a harder to solve discrete logarithm problem.

Transposition ciphers: A type of cipher that hides the message contents by rearranging the order of the letters

CHAPTER 2

LITERATURE REVIEW

In this study a very extensive search for previous works on the problem statement presented in section 1.2 revealed little or nothing. This perhaps, can be attributed to the fact that hitherto, only two categories of finite fields have been in use; the binary Galois fields $GF(2)$ and the prime fields $GF(P)$. The elements of these fields are positive character digits sets thus; ECC arithmetic operations have been only on positive character digit set number systems. In addition, the related VLSI circuits are mainly binary logic CMOS VLSI circuits. There has been no work on RR7SQSD element finite field based ECC as far as the archival literature search of previous works in this study is concerned. However, since the work of this study encompasses MVL/SD arithmetic, finite fields, elliptic curve cryptography and VLSI circuit design; this literature review begins with MVL circuits basic building blocks in section 2.1 and followed with signed digit arithmetic in section 2.2. Survey of finite fields, Elliptic curve basics and elliptic curve cryptosystems are presented in sections 2.3, 2.4 and 2.5 respectively. In section 2.6, secure and trusted communication systems' VLSI circuit organization is presented. Finally section 2.7 presents the summary and conclusion of the literature review.

2.1 MVL circuits basic building block

MVL circuit systems process signal voltage levels represented in higher processing logic thresholds in such manners as to facilitate direct performances of arithmetic operations in higher radices $(r > 2)$. MVL VLSI circuits have already found applications in digital signal processing, arithmetic and logic units design, digital control and robotics, and high-speed

image processing [31 - 35]. The basic building blocks of MVL VLSI circuits are different from the binary logic circuit basic building blocks.

One traditional basic building block of MVL VLSI circuits is the Bi-Direction Current Mode (BDCM) circuit. The current mode nature enables BDCM circuits to realize linear arithmetic operations by simple-internal wiring thereby greatly reducing circuit interconnections and so provides small signal delay time [30, 32 - 33]. However, today's emerging VLSI circuits are fabricated with semiconductor devices that operate in voltage mode thus, rendering BDCM circuits obsolete. One other traditional MVL VLSI circuits' design building block that constructs any arbitrary function of n **variables is the Tree-type Universal Logic Module (T_ULM), popularly referred to as the T-gate. This is naturally voltage mode multiple-valued signals switching multiplexer circuits [35 - 36]. A method of synthesizing T-gate network and the necessary conditions for its use in combinational and sequential logic circuits has been developed. The theoretical work study of [37] on the necessary conditions for a static-hazard-free T-gate eventually resulted in the practical realization of a static-hazard-free T-gate on the level of ternary NAND gates using Emitter Connected Logic (ECL) techniques. T-gate modules have been used to construct arithmetic and memory elements in [38] and quaternary 10-μm NMOSFET integrated circuits for systematic image processing without encoding and decoding and MVL dynamic shift registers for pattern matching cells in [35].**

A typical T-gate circuit comprises a pair of NMOS transistors of different threshold literal voltages, realized by multiple ion implants, as inverters and a pair of pass transistors as analogue switches to multiplex the MVL input signal. T-gate circuits greatly reduce chip interconnections; transistor count and signal delay time but suffer from greater power dissipation relative to the make-and-break circuits of [37 - 38]. On the other hand, Complementary Pass (CP) gates circuits [39-40] are known to exhibit extreme compactness, smaller signal propagation time and very low power dissipation. CP-gate circuits consist of a 'down literal and the traditional analogue switch of N-type Metal Oxide Semiconductor (NMOS) and P-type Metal Oxide Semiconductor (PMOS) transistor circuits.

Present CP-gates and T-gates circuits function in a manner similar to binary logic circuits. They associate positive signal voltage level with positively signed data or variable. Negatively signed data must be represented in some radix compliment form of representation. Consequently, existing MVL VLSI circuits' voltage utilization is restricted to half of the supply voltage rail thereby resulting in supply voltage under utilization. For example, a 5-valued logic system requires 0V,1V,2V,3V and 4V signal voltage levels to correspond to the character digit set of L = {0,1,2,3,4}. In the course of this study, a new MVL circuit's basic building block is developed to utilize both positive and negative signal voltage rails thus maximizing supply voltage utilization.

2.2 SIGNED DIGIT ARITHMETIC

The SDNS is one of a four-member family of number systems referred to in [41 - 42] as 'unconventional fixed-radix number system' and proven to have very high redundancy level and of special application potentials. Other members of this family are the: negative radix number system, sign-logarithm number system and residue number system. The SDNS in particular, presents very efficient carry-free arithmetic as it limits carry propagation chains to only one digit position to the left and enhances parallel system architecture. Some features of SDNS include: no separate sign digit for a negative number, unique representation of '0', each number within the range has two representations while addition is independent of operand length but on two neighbouring digits. Archival search on SDNS [35, 44] revealed two categories namely the unrestricted SDNS and restricted SDNS. This section begins with the general discussion on the natures of SDNS. It will be followed with numerical examples on the SD arithmetic and an epilogue of SD arithmetic applications. It is also noted in literatures that $\bar{i}$ denotes $-i$ however, in this study either of them will be used.

2.2.1 Nature of the SDNS

In a radix-r positional number system the character digits are described by the set $L \in \{0,1,2,.....,r-1\}$ however, in an unrestricted symmetrical SDNS, the character digits are described by $L \in \{(\overline{r-1}),(\overline{r-2}),...,\bar{2},\bar{1},0,1,2,.....,(r-2),(r-1)\}$ with a unique representation for "0" and require no separate sign digit . Thus for a 2-digit number in the range $\bar{9}\bar{9}_{10} \le X \le 99_{10}$ in radix-10 SDNS where $L \in \{-9,-8,...,-1,0,1,....,8,9\}$, there are

199 numbers. If 2 digits represent a radix-10 character digit then each of these character digits is capable of 19 possible representations. This is a total of 361 representations of L and an 81% redundancy level. Each number in the range $\bar{9}\bar{9}_{10} \le X \le 99_{10}$ has two representations e.g. $03 = 1\bar{7} = 3; 01 = 1\bar{9}; 0\bar{2} = 1\bar{8} = -2$ and $-04 = \bar{1}6 = -4$. The attendant problem with high redundancy level is cost intensive hardware implementation. Restricting the character digit set reduces redundancy level and thus reduces cost.

In the restricted radix-r symmetrical SDNS, the character digits are restricted to a value α, such that $\left\lceil \frac{r-1}{2} \right\rceil \le \alpha \le r-1$. A minimum of r different digits are required to represent a number x_i, in such manner that $\bar{\alpha} \le x_i \le \alpha$ that is, $x_i \in \{\bar{a}, \overline{(a-1)}, .., \bar{1}, 0, 1, ..., (a-1), a\}$. For $r = 10$, $5 \le \alpha \le 9$. Let $\alpha = 8$ and for $n = 2$ then, there will be 177 numbers all in the range $\bar{8}\bar{8} \le X \le 88$. With 17 values for each digit, there are a total of 189 representations which represent a 39% redundancy. 1 has one representation .i.e. $01 = 1\bar{9}$. Over reduction of the redundancy level could resurrect the long carry propagation chains.

2.2.2 Numerical examples of SD addition

Generally, the carry-free arithmetic is executed in two phases [33 – 36,42]. Let x, y be some signed digit operands just as u_i and c_i are the interim sum and carry digits respectively in the r-nary symmetrical SDNS. A carry-free SD addition operation of the

type $x \pm y = s \equiv (x_{n-1}, x_{n-2}, ..., x_1, x_0) \pm (y_{n-1}, y_{n-2}, ..., y_1, y_0) = (s_n, s_{n-1}, s_{n-2}, ..., s_1, s_0)$**, can be described by equation (2.1) or (2.2) and exemplified numerically as in Table 2.1.**

$$\left.\begin{aligned} x_i \pm y_i &= rc_i + u_i \\ s_i &= u_i + c_{i-1} \end{aligned}\right\} \qquad (2.1)$$

$$\left.\begin{aligned} &u_i = x_i + y_i - rc_i \ , \\ &c_i = \begin{cases} \bar{1} & ; \text{if } (x_i + y_i) < \bar{a} \\ 1 & ; \text{if } (x_i + y_i) > a \\ 0 & ; \text{otherwise} \end{cases} \end{aligned}\right\} \qquad (2.2)$$

and the final sum, $s_i = u_i + c_{i-1}$

where: $L \in \{-3,-2,-1,0,1,2,3\}$**,** $x_i, y_i, s_i \in L$**,** $c_i \in \{-1,0,1\}$**,** $u_i \in \{-2,-1,0,1,2\}$ **and** $\left\lceil \frac{r-1}{2} \right\rceil \le \alpha \le r-1$**.**

Table 2.1 Numerical examples on archival signed digit addition.

Unrestricted signed digit arithmetic		Restricted signed digit arithmetic	
Radix-10 signed digit (general concept)	Radix-4 signed digit [40]	Radix-10 symmetrical signed digit [38] $(r=10, h=7, a=6)$	
1356+(-580) 1356 = 1356 $-580 = \bar{5}\bar{8}0$ **= 776**	182 + (-83) = 99 X = 182 = 2 3 1 2 +) Y= -83 = -1 -2 3 1 $x_i + y_i = 4c_i + u_i = 0101101\bar{1}$ $s_i = c_{i-1} + u_i = 121\bar{1}$	$x = 3645_{10}$ $y = 1456_{10}$ u_i; $40\bar{1}1$ c_i; $\underline{111}$ s_i; 5101_{10}	$1356_{10} = 3645_7$ $580_{10} = 1456_7$ u_i; $40\bar{1}1$ c_i; $\underline{111}$ s_i; 5101_7

Poster!

$$c_1 = \begin{cases} \bar{1} & ; \text{if} \quad x_i + y_i < -a \\ 1 & ; \text{if} \quad x_i + y_i > a \\ 0 & ; \text{if otherwise} \end{cases}$$
$$u_i = x_i + y_i - rc_i$$
$$s_i = u_i + c_{i-1}$$

2.2.3 Signed digit arithmetic applications

Since the invention of the SDNS [42] signed digit arithmetic has been deployed in various areas of scientific research. For example, the work in [33] employed radix 4 Signed Digit (SD) arithmetic to propose a 4 x 4 signed digit multiplier for digital filters. In another related work [32], an interconnection of radices 2 and 5 SD adders' were used to propose a radix 10 SD parallel adder for use in MVL image processing. This use of multi- or mixed radices requires complex interconnections that bring about reduced system speed, additional

hardware and cost. Hence, most circuit designers settle for the radix-2 application. For example, in [34] radix-2 SD adders are used to implement the arithmetic circuits of a MVL processor for digital control while in [35] radix-5 SD Full Adders (SDFA) are used to realize convolution operations in the design of image processing in robot vision. The high speed compact multiplier on multiple-valued bidirectional current mode circuits in [31] uses a radix-4 SDFA.

However, in all the enumerated applications of SD arithmetic, the operands are coded in the full signed character digit set $L = \{\overline{(r-1)}, \overline{(r-2)}, ..., \bar{2}, \bar{1}, 0, 1, 2, ..., (r-2), (r-1)\}$ **[31,35,41]. For large values of** r**, the arithmetic operations are executed at correspondingly high radices or processing logic thresholds. High processing logic threshold values mean large signal and so supply voltage which in turn needs complex and cost intensive in-circuit generated heat evacuation arrangements. Thus, although the full character digit set coding brings out the high redundancy property necessary to limit the carry propagation however, there is need to perform the associated higher radices arithmetic operations at reduced or restricted processing logic thresholds in practical applications.**

The work in [43] used symmetrical quaternary signed digit number system to design a Symmetrical Quaternary SD Coded Decade Adder (SQSDCDA) similar to that of a binary decade adder. To realize the strategy of decimal addition, each radix-10 SD $x_i \in \{-9,-8,...,-1,0,1,...8,9\}$ **is represented by two components** (u_{i0}, u_{i1}) **where the**

components: $u_{i0} \in \{-2,-1,0,1,2\}$ **and** $u_{i1} \in \{-3,-2,-1,0,1,2,3\}$ **such that the character digits of the radix-10 SD set are as shown in table 2.2**

Table 2 .2 **S**ymmetrical quaternary signed digit equivalent of radix 10 digits

Decimal	-9	-8	- 7	-6	-5	-4	-3	-2	- 1	0	1	2	3	4	5	6	7	8	9
U_{i0}	-2	-2	-1	-1	-1	-1	0	0	0	0	0	0	0	1	1	1	1	2	2
U_{i1}	-1	0	-3	-2	-1	0	-3	-2	-1	0	1	2	3	0	1	2	3	0	1

This is essentially restricting the normal radix-10 signed character digit value. Restricting character digit magnitude though reduces the SDNS redundancy level but is still found useful in certain very special areas. This is evident in the works of [41] and [44] that separately presented different versions of SDNS with reduced character digit set are compatible with equation (2.1). The version in [41] can also be used to convert any radix-10 positional number to a restricted radix-10 symmetrical 7-nary signed digit number and vice visa. It must also be noted that the algorithm there-in presents some implementation difficulties in the binary signed digit domain. Long carry signal propagation chains resurface at the second stage in multi-digit addition operations. Very reliable solutions to the problem have been proffered [41,44 - 46] and several others [47 – 48] on how to avoid carry generation in restricted radix-r signed digit arithmetic. This work will also present the RR7SQSD version of this addition process and give the corresponding solution as part of contribution to knowledge in the next chapter.

2. 3 FINITE FIELDS

The starting point of most combinatorial structures is finite fields hence exegesis of finite field's basic theory can be found in most abstract algebra and vector space texts such as [49 - 54]. Consequently, only certain basic axioms and formulae that can facilitate understanding of the work of this study are reviewed here.

2.3.1 The fundamentals of finite fields

2.3.1.1 Basic Axioms

Axiom 2.1 **A finite field** F_p **, has a definite prime number of elements** p **, called the Galois order of the field with addition and multiplication operations closed.**

That is, if $a,b,c \in F_p$ **then i)** $c = a \pm b \in F_p$ **,** $c = ab \in F_p$ **. ii) The fundamental laws of algebra: Commutative, Associative and Distributive holds true. iii) Identity elements** 0 **and** 1 **exist such that** $a + 0 = a$ **land** $a.1 = a$ **. iv) for any field element** $a \neq 0$ **, there exists a multiplicative inverse** a^{-1} **such that** $a.a^{-1} = 1$**. Furthermore the elements of a finite field form an Abelian group under addition operation. Similarly, all non-zero elements also form an Abelian group under multiplication operation.**

Axiom 2.2 **A Galois order** p **finite field** F_p **, is always designated** $GF(p)$ **and for any** $GF(p)$**, there is a** $GF(q = p^k)$ **, where** $k \geq 1$ **and p a prime and the field elements are isomorphic. For cases of** q **being a prime there exists only one order** $GF(q)$ **that is the elements of such a field are** $0,1,2,3,\ldots\ldots,q-2,q-1$**.**

Axiom 2.3 **All $GF(q)$ have order q which is a power of a prime and every element β in $GF(q)$ has an order that is the smallest positive integer m such that $\beta^m = 1$.**

2.3.1.2 Ground, Extension, Composite and Optimal extension finite fields

These are further classifications of finite fields. If $\psi \in \left\{p, p^k, \left(p^n\right)^m, \left(p^n + \phi\right)^m\right\}$ then, equations (2.3) and (2.3a) define explicitly, the most common types.

$$GF(\Psi) = \begin{cases} \text{a ground field ; if it is of the type } GF(p), \\ \text{an extension field; if it is of the type } GF\left(p^k\right), \\ \\ \text{a composite field; if it is of type } GF\left(\left(p^n\right)^m\right), \\ \text{an optimal extension field; if it is of the type } GF\left(\left(p^n + \phi\right)^m\right) \end{cases} \quad \textbf{(2.3)}$$

where

$$p = \begin{cases} 2 \text{ ; for binary fields} \\ \\ \text{a prime} \geq 3 \text{ ; prime fields} \end{cases} \quad \textbf{(2.3a)}$$

The binary Galois field $GF\left(2^k\right)$ and prime $GF\left(q = p^k\right)$ are larger versions of the corresponding much lower or ground fields $GF(2)$ and $GF(p)$. Such subfields exist for every large field and the number of elements or order of the

$GF(p) < GF\left(p^k\right) \equiv GF\left(\left(p^n\right)^m\right) < GF\left(\left(p^n + \phi\right)^m\right)$**. Field construction as well as arithmetic operations of higher field can be executed using its immediate lower or ground field**

operations. For example, in binary composite fields GF((2^n)^m), where $k = m.n$, GF((2^n)^m) can not only be constructed from $GF(2^n)$ but also the GF((2^n)^m) arithmetic operations can be executed through the $GF(2^n)$ arithmetic operations as shown in the work of [54]. Similarly the optimal extension fields $GF\left(\left(2^n \pm \theta\right)^m\right)$, introduced in [60 - 62] enable field elements to be represented in $k-word$ sequences. This fact is also employed in [54] to study "elliptic curve cryptosystems over optimal extension fields for computationally constrained devices" .

2.3.1.3 Finite field elements representation

The method of representing the elements of a finite field, most often bit or r-nary string pattern, does have an important bearing on the arithmetic of the field. Any finite field element $a \in GF\left(\xi \in \left\{2^k, p^k\right\}\right)$, can be represented as a $radix-\xi$ or a $\xi-nary$ vector $\left(a_{k-1}a_{k-2}a_{k-3}.....a_1a_0\right)$ known as polynomial basis. The terms a_i are the coefficients of the polynomial $a_{k-1}x^{k-1}+a_{k-2}x^{k-2}+a_{k-3}x^{k-3}+....+a_1x+a_0 \in GF(\xi)$ It is also the practice to view the elements of $GF(\xi)$ as a k-dimensional vector space over $GF(\xi)$ so that, there exists a set of k elements $\alpha_0,\alpha_1,\alpha_2,\alpha_3,.....,\alpha_{k-1} \in GF(\xi)$, referred to as the vector basis. however, if $\alpha_i \in GF(\xi)$ are known in advance then, it is sufficient to uniquely express the element a, of any finite field as in equation (2.4)

$$a = \begin{cases} a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + ... + a_1 x + a_0 = \sum_{i=k-1}^{0} a_i x^i \text{ ; if polynomial basis} \\ \\ a_0\alpha_0 + a_1\alpha_1 + ...a_{k-1}\alpha_{k-1} = \sum_{i=0}^{k-1} a_i\alpha_i \text{ ; if vector space basis} \end{cases} \quad \textbf{(2.4)}$$

2.3.1.4 Monic, Irreducible and primitive polynomials

If the element $\gamma \in GF(q)$ then the order of γ is the smallest positive integer m such that $\gamma^m = 1$. An element with order (q-1) in $GF(q)$ is called the primitive element in $GF(q)$ and every $GF(q)$ has at least one primitive element α so that all non-zero elements a_i can be represented as $(q-1)$ consecutive powers of α as in equation (2.5)

$$a_i = \begin{cases} 1, \alpha, \alpha^2, \alpha^3, ..., \alpha^{q-2}, \alpha^{q-1} = 1 \text{ , } \alpha^q = \alpha, ... \text{ ; if } a_i, \alpha \in GF(q) \\ \\ 0, 1, \alpha, \alpha, \alpha, ..., \alpha^{p^{m-2}} \quad \text{; if } a_i, \alpha \in GF(p^k) \end{cases} \quad \textbf{(2.5)}$$

Similarly if $F[X]$ is a $m+1$ degree polynomial then, $f(x) \in F[X]$ can be defined as:

$$f(x) = \begin{cases} \text{a monic polynomial ; if the coefficient of the highest power in } x \text{ is unity.} \\ \text{an irreducible polynomial ;if } (f(x) \in F[X]) \text{ is a polynomial of } m \text{ degree} \\ \text{that cannot be factored further as a product of two polynomials in } F[X]. \\ \text{a primitive polynomial that is, the minimal polynomial of a primitive element } \in GF(p^k). \end{cases} \quad \textbf{(2.6)}$$

Arithmetic operations on elements in polynomial basis representation are reduced to modulo the irreducible polynomial. The polynomial basis representation is suitable for the coordinate-wise addition operation in $GF(q)$, multiplication operation is better off with chosen primitive element. A table of translation between polynomial and primitive is needed to perform both operations.

2.3.1.5 Finite field construction

Constructing a field is essentially reducing a degree *m+1* polynomial F[X] by the irreducible polynomials f(x) i.e. F[X]/f(x). All primitive elements are generators of finite fields, hence the following steps constitute a simple procedure for constructing finite fields [55].

- **i) List all monic polynomials of the field $GF(q)$**
- **ii) Obtain the irreducible polynomials that is, those polynomials with constant terms in the above list.**
- **iii) Obtain the irreducible monic polynomials by taking each in turn and evaluating the polynomial by substituting all field elements for x. A polynomial that evaluates to zero in an element of x is not irreducible.**
- **iv) Determine the primitive element that is a generator of a cyclic group of the highest order, i.e.; order q-1**

The binary Galois field $GF(2^k)$, and the prime field $GF(q = p^k)$ with distinguishing properties were indeed the only two categories of finite fields hitherto have been widely studied. During the course of this research, a third category of finite fields that

combines the two properties was presented in [57]. A synopsis of this work is presented in section (3.1.2).

2.3.2 FINITE FIELDS' BASIC ARITHMETIC OPERATIONS

Finite fields' arithmetic comprises addition, multiplication, squaring and inversion operations. Field arithmetic can be performed either in Polynomial or vector space basis representation of field elements [55 – 64]. However, while the polynomial basis representation of elements in $GF(2^k)$ enhances component-wise modulo 2 additions or bit-wise XOR operation [52-53] non-zero elements of vector space basis are expressed as consecutive powers of the field generator element α. This approach particularly renders polynomials basis presentation suitable for prime fields where addition operation is performed coordinate wise and multiplication/ exponentiation with reduction of the exponent mod $q-1$. When the elements of a field are represented in vector basis, a careful choice of basis type i.e.; normal, binomial or trinomial simplifies squaring and multiplication operations.

2.3.2.1 Computing field element inverse

Field element inverse computation has found indispensable applications in cryptography and code theory in the detection of errors such as Reed Solomon and other BCH error correcting codes [50]. Computing the inverse of the elements of a field is characteristically slow and cost intensive yet the operation is an indispensable component procedure in finite field arithmetic operations. Division operation can only be executed through field inversion precisely, by multiplying the dividend with the field

inverse of the divisor ($c = a/b \equiv a.b^{-1} \in F_p$ **). Several very efficient methods of computing field element inverse have been proposed in the past. In [58] five basic methods are suggested namely: reciprocal ,Extended Euclidean Algorithm (EEA), Lagrange representation, repeated squaring, and Discrete Logarithm Problem (DLP) methods of performing field inversion operation. The comparative speed test conducted in [58] (with a Pentium IV processor), on the various methods using GF(2^{155}) concluded that the EEA method is faster than the Lagrange method. Efficient EEA based inversion algorithm using multiply instructions and factorization of exponents to compute field inversion operations have been presented in [59] and [60] respectively. In [60] only ten multiplication operations are required to compute an inverse in** $GF\left(2^{155}\right)$**.**

2.3.2.2 Computing field: addition, multiplication and squaring

Let a,b **represent two distinct elements and let the members of the set** $\{c,d\}$**, be such that,** c **represents the result of either addition, multiplication or squaring operation and** d **, represents the result of inversion operation , for any** $a \neq 0, d = a^{-1}$ **in any of the fields: binary or prime,** $a,b,c,d \in \left\{GF\left(2^k\right), GF\left(p^k\right)\right\}$**. Now using the polynomial basis representation and establishing the format that [54-55]:**

a) For the binary field:

i) an irreducible polynomial $w(x) \in F_{2^k}$ **of degree** k **,is described by equation (2.7)**

$$w(x) = x^k + w_{k-1}x^{k-1} + w_{k-2}x^{k-2} + w_{k-3}x^{k-3} + ,....w_1x + w_0 : w_i \in \{0,1\} \qquad \textbf{(2.7)}$$

ii) $GF(2^k)$ **consisting of all polynomials of degree less than** k **can be expressed as**

$$GF(2^k)=\vartheta_{k-1}x^{k-1}+\vartheta_{k-2}x^{k-2}+.....+\vartheta_2x^2+\vartheta x+\vartheta_0 : \vartheta_i \in \{0,1\} \quad \textbf{(2.8)}$$

iii) a field element $a,$ **is also a polynomial of degree** $k-1$ **can be denoted by the binary string** $(a_{m-1}a_{m-2}.....a_2a_1); a \in \{0,1\}$ **is equally describable as**

$$a(x)=a_{k-1}x^{k-1}+a_{k-2}x^{k-2}+.....+a_1x+a_0 \quad \textbf{(2.9)}$$

such that $GF(2^k)=\{(a_{k-1}a_{k-1}.....a_1a_0): a_i \in \{0,1\}\}$**. The multiplicative identity** '1' **is represented as** $1=(000....01)$ **and the additive identity** '0' **as** $0=(000.....00)$**.**

b) For the prime field:

if in a Prime field, $g(x)$ **and** $h(x)$ **are two integer coefficient polynomials such that**

$g(x)=a_lx^l+a_{l-1}x^{l-1}+.....+a_1x+a_0$**,**

$h(x)=b_mx^m+b_{m-1}x^{m-1}b_{m-2}x^{m-2}+.....+b_1x+b_0$

and their product $c(x)$**, also an integer coefficient polynomial be represented as in equation (2.10)**

$$c(x)=g(x).h(x)=c_{i+m}x^{i+m}+.....+c_1x+c_0 \quad \textbf{(2.10)}$$

then their coefficients product can be equally described as

$$(a_la_{l-1}....a_1a_0).(b_mb_{m-1}.....b_1b_0)=(c_{i+m}c_{i+m-1}.....c_1c_0.) \quad \textbf{(2.11)}$$

This implies that elements of the finite prime field $GF(p^k)$ **comprising the set of integers** $\{0,1,2,3,.....,p-1\}$ **can be represented in normal basis polynomials of degree less than** k, **in** $GF(p)[x]$ **just as the field** $GF(p^k)$ **is represented with respect to an irreducible polynomial** $W(x)$ **of degree** k, **over** $GF(p^k)$**. Any element** A **of** $GF(p^k)$ **can then be expressed as a polynomial** $A(x)$ **of degree** $\leq (k-1)$ **with coefficients as in equation (2.12)**

$$A(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + ... + a_2x^2 + a_1x + a_0 \qquad \textbf{(2.12)}$$

where $a_i \in \{0,1,2,3,....., p-1\}$**.**

Consequently, if $\xi \in (2, p)$ **then the basic field operations may be generalized as in table 2.3.**

Table 2. 3 Basic field arithmetic operations formulae

arithmetic operation	$GF(2^k)$ **or** $GF(p^k)$ **field**
Addition	$\sum_{i=m-1}^{0} c_i x^i = \sum_{i=m-1}^{0} ((a_l + b_l) \bmod \xi) x^i$
Multiplication	$\left(\sum_{j=m-1}^{j=0} a_j x^j \left(\sum_{i=m-1}^{0} b_i x^i \right) \right) \bmod w(x)$
Squaring	$\left(\sum_{i=0}^{k-1} a_i x^i \right)^2 \bmod w(x)$

The coefficient polynomial representation will be used to design a finite field element multiplication LSI circuit in chapter 3.

2.4. Elliptic curve basics

2.4.1 Introduction to Elliptic curves

Elliptic curves [1,3 - 9,13,15 - 17,22 - 23,61 - 76] ranks 3rd after straight lines and conics in the family of curves. In the generalized model of Weierstrass [4,12,22,31,60], elliptic curves can be represented by the expression

$$E: y^2 + a_1xy + a_2y = x^3 + a_2x^2 + a_4x + a_6 \qquad \textbf{(2.13)}$$

with the coefficients $\{a_i\}$**,** $i = 1,2,3,4,6$ **as elements of a finite field** F **. It is customary [15,31,64 – 70] to describe an elliptic curve either defined either** $GF(2^k)$ **with the parameters** $a,b \in GF(2^k), b \neq 0$ **or over** $GF(p^k)$ **with** $a,b \in GF(p^k)$ **as the set of all solutions** $(x,y) \in GF(2^k)$ **or** $(x,y) \in GF(p^K)$ **by the system of equations (2.14) together with a point** O**, the point at infinity.**

$$E: \begin{cases} y^2 = x^3 + ax + b \quad \textbf{;for prime fields } p \geq 3; a,b,(x,y) \in GF(p^k) \textbf{ and} \\ \qquad\qquad 4a^3 + 27b^2 \neq 0 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textbf{(2.14)} \\ y^2 + xy = x^3 + ax^2 + b \textbf{ ; for binary Galois fields and } a,b,(x,y) \in GF(2^k) \end{cases}$$

In elliptic curves defined over the binary fields $GF(2^k)$**, the elements are represented by k-bit binary code words making arithmetic operations binary VLSI implement friendly. The prime**

field expression in equation (2.14) can equally represent an EC defined over real numbers. Calculations with real numbers are prone to round-off errors. Cryptographic arithmetic is required to be error free. EC defined over real numbers are not suitable for implementation in cryptosystems. Restricting $a, b, x, y \in F_q$ **such that** q **is a prime or at least power of a prime** p**, i.e.,** $q = p^h$ **where** $h = 1,2,3,...$**, does not only obviate this bottle neck but turns around prime field EC to elliptic curve cryptosystems software implementation friendly. Software based cryptosystem protocol deployed in secure and trusted communication system though small and potable but is venerable to fraud particularly in the banking service. Another part of the contribution to knowledge in this study is making the prime field ECC hardware implementation friendly.**

2.4.2 Elliptic curve arithmetic operations

Elliptic curve arithmetic operations comprise elliptic points' addition, elliptic curve point doubling and elliptic curve scalar or point multiplication. The third operation is supported by the first two operations which are in turn supported by finite field: addition, multiplication, squaring and inversion operations. Efficiency of elliptic curve arithmetic thus, depends on the manner of the underlying field element representation. Similarly, the major run time contributing operations are: the finite field multiplication and inversion however, the most run time consuming operation in elliptic curve cryptosystem implementations like ECDHA and ECDSA is the point multiplication. Consequently, elliptic curve point multiplication determines the overall efficiency of elliptic curve cryptosystems. Elliptic curve cryptosystem arithmetic is big integer operand arithmetic and most emerging secure and trusted

communication systems are elliptic curve cryptography supported. These are the constrained environment devices like pocket size laptop computers, PDA, mobile phones, pagers, handheld or Palm OS that connect to the internet over electronic data channels to handle electronic data, voice and picture (3G) signals. They do this by performing several millions of big integer EC points addition, point doubling and point multiplication operations per second. What follows is a brief explanation of these operations obtainable in [4-5,7-9,11,15-17,22,26-26,31,67 -76].

2.4.2.1 Elliptic curve points' addition and point doubling

Points' addition and point doubling can be performed either geometrically or analytically. If $P(x_1, y_1)$ **and** $Q(x_2, y_2)$ **are two points on an EC elliptic curves over** (F_{2^k}) **or** $E(F_{p^k})$**, then** $P + Q = R = (x_3, y_3)$**. When** $Q = 0$**,** $P^* = (x_1^*, y_1^*)$ **and** $R^* = P^* + P^* = 2P^* = (x_3^*, y_3^*)$ **that is** $2P^* = R^*$**. The geometrical method is as shown in figure 2. 1. The analytical method is as summarised in table 2.4. The derivation of the formulae of table 2.4 can be found in [24,31,70 - 71,73].**

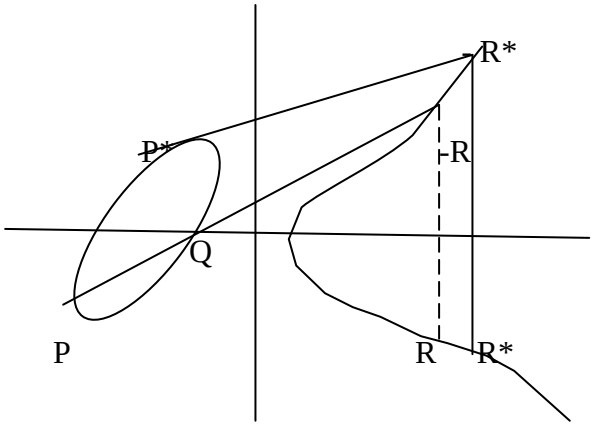


Figure 2.1: Elliptic curve geometric point addition and point doubling

Table 2. 4 **Addition rules for** $\{(x,y),x,y\} \in E(F_{2^k}), E(F_{q_k})$

Field	$(x,y),x,y \in GF(2^k)$	$(x,y),x,y \in GF(p^k)$
Equation	$y^2 + xy = x^3 + ax^2 + b$	$y^2 = x^3 + ax + b$ **;** $4a^3 + 27b^2 \neq 0$
	$O + O = O$	$O + O = O$
	$(x,y) + O = O + (x,y) = (x,y)$	$(x,y) + O = O + (x,y) = (x,y)$
	$(x,y) + (x, x+y) = O; (x,y) \in E(F_{2^k})$	$(x,y) + (x-y) = O; (x,y) \in E(F_{p^k})$
λ	$\frac{y_2 + y_1}{x_2 + x_1}$ **; if addition i.e.** $P \neq Q$ $x_1^* + \frac{y_1^*}{x_1^*}$ **;if doubling i.e.,** $P = Q$	$\frac{y_2 - y_1}{x_2 - x_1}$ **; if addition i.e.;** $P \neq Q$ $\frac{3(x_1^*)^2 + a}{2y_1^*}$ **; if doubling i.e.;** $P = Q$
x_3	$\lambda^2 + \lambda + x_1 + x_2 + a$ **; if addition** $\lambda^2 + \lambda + a$ **; if doubling**	$\lambda^2 - x_1 - x_2$ **; if addition** $\lambda^2 - 2x_1^*$ **; if doubling**
y_3	$\lambda(x_1 + x_2) + x_3 + y_1$ **; if addition** $(x_1^*)^2 + (\lambda + 1)x_3$ **; if doubling**	$\lambda(x_1 - x_3) - y_1$ **; if addition** $\lambda(x_1^* - x_3) - y_1^*$ **; if doubling**

Computing EC points 'addition and point doubling operations, most often is either in affine or in projective coordinate systems [7,9,11,31,64,71 - 73], the choice being determined by the ratio ξ **[77], the duration of one inversion to one multiplication i.e.;**

$$\xi = \frac{\text{time to compute an inverse}}{\text{time to multiply}}. \qquad \textbf{(2.15)}$$

Thus, the choice $C = \begin{cases} \textbf{affine points coordinate system; if } \xi < 1 \\ \\ \textbf{projective points coordinate system; if } \xi >> 1 \end{cases}$ **(2.16)**

That is, only large values of this ratio ξ, attract projective point's coordinate system implementation. On the other hand in terms of inter field operations' relationship, in [4] this ratio is placed at 3 to 8 for binary field and 30 to 80 for a prime field. Similarly, while squaring operation in binary fields is negligible, in the prime field, it is considered to be 0.8 of one multiplication operation. The duration of addition operation in both binary and prime fields, is assumed to be negligible.

Consequently, in using EC arithmetic consideration must be given to the following: i) field and the field representation i.e.; normal basis or polynomial basis, ii) implementation strategies, iii) efficiency of the underlying field operations i.e.; the number of field: inversion, multiplication m, squaring s and addition a, operations required to execute the elliptic curve points' addition operation and point doubling operations . Usually the number of inversions is far greater than the number of multiplication operations which is in turn by far greater than the number of addition operations, iv) field generating polynomial i.e.; trinomial or pentanomial and v) implementation i.e.; the speed of realizing these basic field operations in the underlying field [74].

2.4.2.2 Elliptic curve point multiplication $W = mp$

Elliptic curve point multiplication is analogous to modular exponentiation in non-ECC domain. and it is essentially scaling a point $P = (x, y)$ with a scalar quantity m, in such manner that the product $W = mP = (x_w, y_w)$, also lies on the given EC as earlier stated

in section 2.4.2. Elliptic curve point multiplication is a strictly error free repeated addition section 2.4.2. Elliptic curve point multiplication is a strictly error free repeated addition of two points and doubling of a point on the elliptic curve operation. Furthermore, elliptic curve point multiplication is performed in the affined coordinate systems [69]. The affined coordinate system require a minimum of one field inversion operation to evaluate a basic arithmetic operation thus, making elliptic curve point multiplication in affined coordinates a costly exercise. Hence, there is a need for EC point multiplication operation's speed-up schemes.

The search for a cost effective and fast EC point multiplication otherwise speed-up schemes [4, 5, 24] continues. Some suggested methods include: i) the formulation of specific strategies that would enable the base point P, to be represented in affine and projective coordinates systems so that EC point multiplication operation can be performed in a combine affine and projective coordinates system environment. Or on the alternative, perform all necessary arithmetic operations except inversion in projective coordinate system and return to execute just a single inversion in the initial affine coordinate system. A problem of circumstances in the migration is the compute intensive nature, resulting in a slow computational procedure. ii) Developing high-speed EC arithmetic operations strategies such as several: doublings $2P$, additions $2P+Q$, tripling $3P$ and quadrupling $4P$ [75]. iii) The use of Double Base Number System (DBNS) [4] (for prime field elliptic curves) . iv) The use of addition chains, addition-subtraction or binary chains as well as differential or special addition chains [70-71] has

been presented. For the last two suggestions the scalar quantity k is represented in either binary, ternary or Near Adjacent Form (NAF) forms with a view to discouraging the use large numbers of EC point addition operations. However, all the four suggestions focuses on EC defined over positively elements finite fields. There have been no major contributions on point multiplication operation for elliptic curves defined over RR7SQSD elements' finite fields. In this study contribution to knowledge will be made on: i) RR7SQSD addition/subtraction based very big integer operands multiplication/modular exponentiation scheme. ii) RR7SQSD based EC point multiplication peed-up scheme.

2.4.2.3 Order of computational complexity

This is a computational cost (timing) analysis of EC arithmetic operations in an underlying field. It is essentially a figure of merit for adjudging the degree of efficiency of EC: points' addition, point doubling and point multiplication algorithms or formulae. Practically, the order of complexity of the given algorithm is determined by calculating the total numbers' of field: addition, multiplication, squaring and inversion operations. An EC arithmetic operation algorithm or formular with fewer number of the various field operations is considered efficient. Tables 2.5 and 2.6 where A,M,S and I denotes Addition, Multiplication, Squaring and Inversion operations respectively show extracts from the results' [31,76] of such complexity cost computations for points' addition, point doubling and point multiplication operations. In this study these values will be used as bench marks.

Table 2.5 Computational complexity cost analysis of EC operations

Authors	Curve operation	F_p								F_{2^m}							
		Affine				Projective				Affine				Projective			
		A	M	S	I	A	M	S	I	A	M	S	I	A	M	S	I
[34]	Addition	*	3	*	1	*	16	*	*	*	2	1	1	*	15	5	*
	Doubling	*	4	*	1	*	10	*	*	*	2	1	1	*	14	4	*
[72]	Addition	5	2	1	1	*	*	*	*	9	2	1	1	*	*	*	*
	Doubling	4	5	2	1	*	*	*	*	4	2	2	1	*	*	*	*

Table 2.6 Computational complexity analysis for fast EC point multiplication

Author	Designation	EC operation	Prime field				Binary field			
			A	M	S	I	A	M	S	I
[1]	A	P+Q	*	2	1	1	*	2	1	1
	D	2P	*	2	1	1	*	2	1	1
	DA	2P+Q	*	9	2	1	*	9	2	1
	T	3P	*	7	4	1	*	7	4	1
	TA	3P+Q	*	9	4	2	*	9	3	2
	Q	4P	*	9	5	1	*	9	3	2
	QA	4P+Q	*	11	4	2	*	10	6	2
[6]	Several Doublings	2P	*	5	5	1	*	5	5	1
		4P	*	9	9	1	*	*	*	*
		8P	*	13	13	1	*	*	*	*
		16P	*	17	17	1	*	*	*	*

2.4.3 EC point multiplication and ECC

It is a known fact that EC point multiplication operation is the centre point of ECC. ECC can be constructed on group of points on an EC defined over a finite field and the arithmetic is error free. A high-speed point multiplication scheme means a highly efficient EC based cryptographic protocol. The work in [8] while proposing an efficient elliptic curve

exponentiation in $GF(p)$ observed that there is a considerable level of advantage of the affine coordinate system over the projective. Comparing the ratio of computational amount of division in GF (p) to that of multiplication as in equations (2.15) - (2.16), he found this ratio to be greater than 9. The significance of this study is that in recent times speed-up schemes such as performing direct computation in affine coordinate with several doublings have continued to capture researchers' attention. Several novel works such as [8, 57, 67, 70, 72] show-cased efficient algorithms for direct $3p,4p,18p,16p$ computation on EC over $GF(2^k)$ but all without exception are binary CMOS VLSI implementation friendly. To do the same in non-binary or higher radix processing logic threshold systems requires the incorporation of relevant interfacing units. (In the course of the present study, the existing binary or Galois field ECC concept is modified into a higher radix otherwise Multiple-Valued Logic (MVL) ECC implement capability.

2.5 Elliptic curve cryptography in practice

2.5.1 The Domain parameters

For cryptographic applications an EC defined over $GF(2^k)$ or $GF(p^k)$ is described by the seven-parameter single quantity Γ, called the domain parameters [36, 48].

$$\Gamma = (q, FR, a, b, G, n, h) \quad \textbf{(2.17)}$$

These parameters strictly control and ensure the cryptographic compliance of the EC. The member elements of the domain parameters may be defined as follows: q is a prime or 2^k that defines the field; FR is the field elements representation: i.e., polynomial normal basis,

normal vector space. a,b**: are the curve coefficients. Their values depend upon the security requirement; a base point** $G=(x_G,y_G)\in E(F_q)$ **that has the largest order** n**, usually a large prime. The quantity** h **is the ratio of the elliptic curve's total number of rational points** $N=\#E(F_q)$ **to the order of the base point** n**. One necessary condition for cryptographic compliance is that** n **divides** N **without a remainder that is,** $h=E(F_q)/n$ **is an integer [73]. The Domain parameters enable a member in a PKC communication community to generate his/her public and private keys by selecting a random integer** d**, such that** $1<d<n-1$ **and computes the public key** $Q=dG$**. A validation process is required of every public key's legitimacy.**

2.5.2 Popular ECC protocols

A glossary of the various public key elliptic curves, cryptosystem protocols can be found in [4, 63]. These protocols, though differ in presentation and application, do have some common basic features particularly in the cases of Diffie-Hellman and ElGamal's. Assuming a communication community of two users A and B who have will agree on a very large prime n**. Both parties are expected to select privately, random integers** (d_A,d_B) **from the interval** $[1,n-1]$ **and publish the same in an insecure channel, the corresponding products** $(Q_A=d_AG,Q_B=d_BG)$ **with a view to interchanging their public keys. Assuming A wants to send a secret message to B over the insecure channel. User A then codes his plain text message, encrypts the coded message** m **using the formular** $(d_AG,(d_AQ_B)_x+m)$ **and sends this cipher text to B over the insecure channel. User B on receipt of the message decrypts it by computing**

$m+(d_A Q_B)_x - d_B d_A G_x = m+(d_A d_B G)_x - (d_B d_A G)_x = m$. **The process of plain message text embedment, encryption and decryption has never been done with elliptic curves defined over RR7SQSD elements finite field.**

2.6 Secure and trusted communication systems' VLSI circuits

Cryptosystems widely used in today's secure and trusted communication gadgets concentrate on systems of elliptic curves defined over binary Galois fields $GF(2^k)$**. Elliptic curve cryptographic processor architectures [64] often consist of program memories, an arithmetic unit that performs points addition and point doubling operations, the arithmetic unit controller and a main controller as basic components. Major components for the arithmetic unit VLSI are field element: adders, multipliers, squaring and inversion units. The design of these units, particularly high performance multipliers and inversion circuits has also continued to arrest the attention of the scientific community over the years. For example: The work of [77] presented two Systolic Multipliers that perform product-sum computation** $AB+C$ **in finite fields** $GF(2^m)$**. Before then in 1981, Massey and Omura in a paper titled "Computational method and apparatus for finite field arithmetic" presented the maiden design for multiplication in finite fields using normal basis. Others works such as [76] following the same suite presented finite field multiplier based on self-dual Normal Basis using only** $2(m^2-m)$ **XOR gates. The work of [79] equally followed a similar path by suggesting an efficient Optimal Normal base Type II multiplier that requires** $1.5(m^2-m)$ **XOR gates. The time complexities of both Massey-Omura and [79] are**

similar. Again the work in [80] constructed an efficient gate circuit for computing multiplicative inverses over $GF(2^m)$ based on a recursive algorithm superior to conventional algebra for computing multiplicative inverses. The work in [81] uses a serial-in parallel-out multiplication strategy to realize fast inversion in $GF(2^m)$ algorithm for VLSI implementation. This algorithm required a computational time of $O(m\log_2 m)$. A bit-serial architecture for efficient addition and multiplication in binary finite fields using polynomial basis representation implemented with low-voltage/low-power properties presented in [82] requires an equivalent of 28 gates and it is scalable.

There have been similar implementations based on prime field VLSI circuits. Other works worthy of mention include: [83] that proposes a systolic Gaussian Elimination over $GF(p)$ with partial pivoting for the triangularization of large dense n x n matrices over prime Galois fields. An AT^2-Optimal Galois Field multiplier VLSI circuit for error detection and error correction as presented in the work of [84] is based on Discrete Fourier Transformation (DFT). This work is asymptotically optimal with respect to area A and time T with a lower bound of $AT^2 = \Omega(n^2)$ and a matching computation time T in the range of $\left[\Omega(\log n, O(\sqrt{n}))\right]$. The works of [82,85] discussed a multiply/ accumulator elliptic curve arithmetic unit that combines integer and polynomial arithmetic into a single functional unit for $Gf(p)/GF(2^m)$ These works showed that dual field multipliers have less power in polynomial mode than in integer mode A number of EC cryptosystems have also been proposed over the years using either field

programmable Level System Circuits (FPLSC) [86] or Field Programmable Gate Array (FPGA) [87] or T1 MSP430x33 family microcontrollers [88]. Again no work has been done as far as archival search can vet, on either SMVL ECC or on implementing EC cryptosystems based on RR7SQSD finite fields. In this study, an overview of EC point multiplication SMVL VLSI circuit organization, synthesis of the functional logic circuits and the design implementation of the basic building block for such circuits are very necessary.

2.7 Summary and conclusion of the literature review

The literature survey has been wide and broad based but it is acknowledged that is by no means exhaustive. For the entire archival search it has not been possible find previous works on the following areas (though it is possible that they exist in the far-reach):

i) **Suggestions on the RR7SQSDNS as an alternative processing threshold logic system and its arithmetic.**
ii) **RR7SQSD addition/subtraction chains formulation and its application to big integer operands multiplication/modular exponentiation Scheme.**
iii) **Arithmetic operations on EC defined over RR7SQSD element finite fields or the implementation of RR7SQSDNS base ECC.**
iv) **Design and implementation of a MVL circuit basic building block to function on RR7SQSD signal processing threshold voltage profile.**

Their formulation and realizations for the first time will be implemented in this study.

CHAPTER 3

METHODOLOGY

3.0 Preamble

This chapter presents in section 3.1, the theoretical frameworks of the formulated research objectives detailed in section 1.3.2 and in section 3.2, the methods used to realize them. Experiments were conducted on the various formulated concepts to ascertain their correctness. Similarly, circuits designed were (by way of simulation) subjected to near-real-life conditions to evaluate their performances.

3.1 Theoretical frame work

In section 3.1.1 the basis of an RR7SQSD arithmetic operation of practical significance is presented. The proof that the RR7SQSDNS is an appropriate SMVL algebraic structure and the description of EC defined over such an RR7SQSD element finite fields is presented in section 3.1.2. This will facilitate the design of a hardware implementable algorithm for the arithmetic operations on EC defined over RR7SQSD fields. In section 3.1.3, the architectural organisation of the SMVL ECC arithmetic unit VLSI circuit layout is presented. Formulation of the mathematical model of the SMVL arithmetic unit VLSI circuits' basic building block and the synthesis of the constituent functional logic circuits are presented in section 3.1.4. Numerical examples are given to validate the correctness of the various mathematical models in each case

3.1.1 RR7SQSD based big integer operands arithmetic

The primary objective here is to show that the RR7SQSD number system indeed supports error-free arithmetic operations on very big word length integer operands and at constant

time. It is therefore pertinent to present the generalized nature of Restricted Radix-h Symmetrical r-nary $(h,r)>2$, signed digit number system in section 3.1.1.1. The reason for the choices of radix-7, symmetrical radix-7 Signed Digit Number System (SDNS), and the restriction imposed on the radix-7 SDNS that transformed it into quaternary SDNS are given in sections 3.1.1.2 and 3.1.1.3 respectively.. Section 3.1.1.4 establishes the fact that RR7SQSDNS arithmetic is not only a SMVL analogous to the binary logic modulo-2 arithmetic but also do support very efficient arithmetic operation strategies similar to those in [33, 35 - 36, 38]. A scheme for RR7SQSDNS parallel addition/multiplication is presented in section 3.1.1.5. Section 3.1.1.6 shows that even the fool-hardy multiplicative inverse calculation process can be computed in RR7SQSD domain. The principle of an RR7SQSD addition/subtraction chain is presented in section 3.1.1.7. The section ends with a numerical example to prove the formulated addition/subtraction chain.

3.1.1.1 The Restricted Radix -h Symmetrical r-nary signed digit number system axioms

In a given list of radix-h $(h>2)$, character digits, reshuffling the order of appearance of any member-character digits does not change the significant value of that character digit. For example, the digit sets L={0,1,2,3,4,5,6,7,8,9} and L' ={1,0,3,6,2,4,5,9,8,7} are all 10-valued, radix - 10 character digit sets. The significant value of figure '5' in set L does not change due to its change of position in set L'. Similarly, the elements of the set of an h-valued positive thresholds processing logic system are the character digits of a radix – h number system with one to one correspondence. Arithmetic operations on the set of the h-valued logic threshold elements are modulo-h arithmetic operations. Thus, if

the h-valued member elements are (symmetrical or non-symmetrical) signed digits then, the corresponding h-valued processing logic threshold system is an h-valued (symmetrical or non-symmetrical) signed-digit processing threshold logic system and the arithmetic operations on such systems are equally radix-h (symmetrical or non-symmetrical) signed-digit arithmetic operations.

Definition 3.1 **A SDNS is said to be symmetrical if and only if for every positive character digit θ_i there is a corresponding negative character digit $-\theta_i$ except the digit '0', otherwise it is non-symmetrical.**

Axiom 3.1 **For every radix-h positional number system with the character digit set $L = \{0,1,2,.....,(h-1)\}$ there exist a $(2h-1)$ -valued symmetrical radix-h signed-digit number system with the character digit set $L' = \{\overline{(h-1)}, \overline{(h-2)}, ,\bar{2},\bar{1},0,1,2,3,,,,(h-2),(h-1)\}$ where $\bar{i} = -i$**

The character digits set, L and L' are both higher radix number systems and thus are Multiple Valued processing threshold Logic (MVL) systems [88 – 95]. The difference between the two sets is that L' is a symmetrical signed digit number system and as such supports very efficient arithmetic. Corresponding Symmetrical Multiple Valued Logic (SMVL) system's VLSI circuits could utilize both rails of the signal voltage for processing of information, miniaturized size, high compactness, greater reliability and very high processing speed. However, for large values of h and where the elements of $L' = \{\overline{(h-1)}, \overline{(h-2)}, ,\bar{2},\bar{1},0,1,2,3,,,,(h-2),(h-1)\}$ directly represent signal voltage levels then, such SMVL VLSI circuits will radiate large amount of heat. Replacing the large

character digit elements of the set $L=\{0,1,2,.....,(h-1)\}$**, with smaller values in a symmetrical radix-r (**$r=\left(\frac{h+1}{2}\right)$**) signed character digit set, reduces the processing logic threshold signal voltage level. The reduced processing threshold values enable higher radix-h symmetrical signed digit arithmetic operations to be performed in the new radix-r signed digit number system platform. Consequently, the corresponding SMVL VLSI circuits sustain the miniature, compact, high-speed properties and acquire low in-circuit heat radiation ability required of today's emerging secure and trusted communication systems.**

Axiom 3.2 **For any radix-h** $(h>2)$**, positional number system with the character digit set** $L=\{0,1,2,.....,(h-1)\}$**, there is a restricted radix-h symmetrical** $r=\left(\frac{h+1}{2}\right)-nary$ **signed-digit number system with the restricting value** a **and of character digit set** $L''=\{[r-k]\}$ **where** $k\in\{h,(h-1),.....,2,1\}$ **if and only if**

i) $\frac{h}{2}\neq\left\lceil\frac{h}{2}\right\rceil$ **ii)** $a\leq\left(h-\left(\frac{h+1}{2}\right)\right)\equiv\left(\left(\frac{h+1}{2}\right)-1\right)$

For example, in the radix–9 number system where the character digit set $L=\{0,1,2,3,4,5,6,7,8\}$**, the SD set L''= {0,1,2,3,-5,-4,-3,-2,-1} is not symmetrical but the set L'''= {0,1,2,3,4,-4,-3,-2,-1} = {-4,-3,-2,-1,0,1,2,3,4} is symmetrical. Not-with-standing both L'' and L''' are examples of the restricted radix-9 symmetrical r-nary SDSN with the delimiters** $a=5>\left(h-\left(\frac{h+1}{2}\right)\right)$ **and** $a=4=\left(h-\left(\frac{h+1}{2}\right)\right)$ **respectively. It is observed**

that redundancy level reduces drastically for values of $a < \left(\left(\frac{h+1}{2}\right)-1\right)$ **and the character digit set of the restricted radix-h symmetrical r-nary SDNS is an Abelian group in which arithmetic operations are closed.**

3.1.1.2 The choice of "radix-7" and the nature of RR7SQSDNS

In this study, the value of, h **is taken as 7 and the following reasons informed the choice of "7";**

i) The character digit-set $L = \{-3,-2,-1,0,1,2,3\}$ **simplifies conversion to binary logic (if need be) procedures as shown in table 3.1.**

Table 3.1 Decimal-Binary-RR7SQSD equivalents

Unsigned decimal	**4**	**5**	**6**	**0**	**1**	**2**	**3**
Binary coding	**100**	**101**	**110**	**000**	**001**	**010**	**011**
RR7SQSD coding	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**

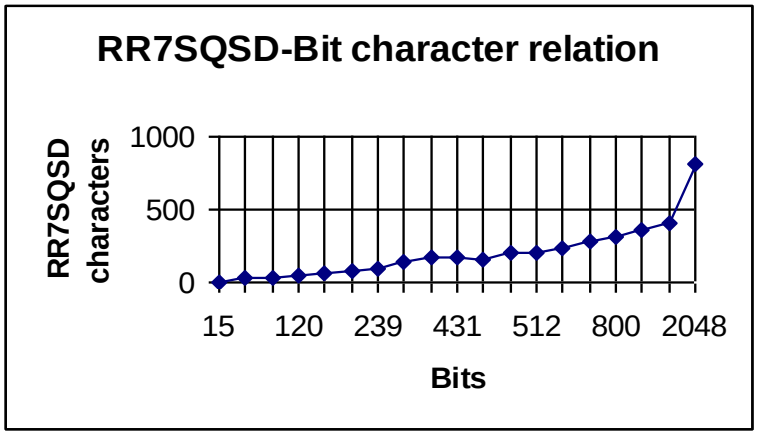
ii) The processing signal threshold voltage levels of –3V, -2V, -1V, 0V, 1V, 2V and 3V are compatible with most emerging semiconductor devices' [86 - 93] supply voltages and thus can directly represent the SMVL thresholds of {-3,-2,-1,0,1,2,3}

iii) The primitive root of *p = 7* is applicable to a set of cryptographically friendly prime moduli that will be elucidated later in this study.

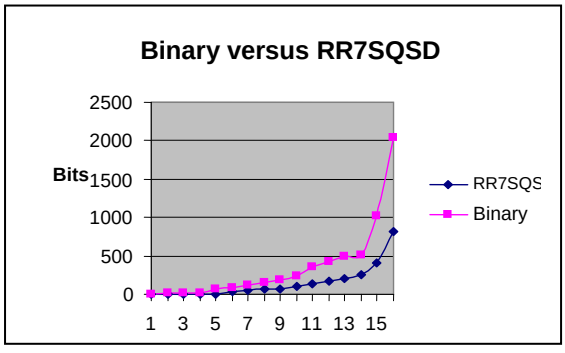
iv) The information per line richness of the RR7SQSD system as depicted in the plots of figure 3.1 meets the required cryptosystem standards. For example, figure 3.1a shows the RR7SQSD-bit equivalent relationship. The plot is linear with a small

rise at 239 bits which means numbers of less than 239 bits in length attract less RR7SQSD

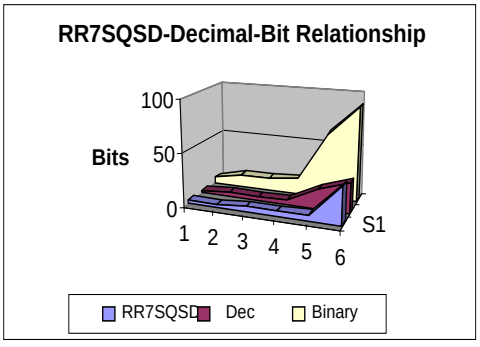
v) suggesting greater information density with operands of smaller magnitude. This fact is further substantiated by figure 3.1b where the same is compared with the binary number system. Figure 3.1c is a tripartite Decimal-RR7SQSD-Binary comparison. Figure 3.1d is a projection of RR7SQSD ECC key size to that of the conventional number system ECC under the background of other or non-EC cryptosystems.



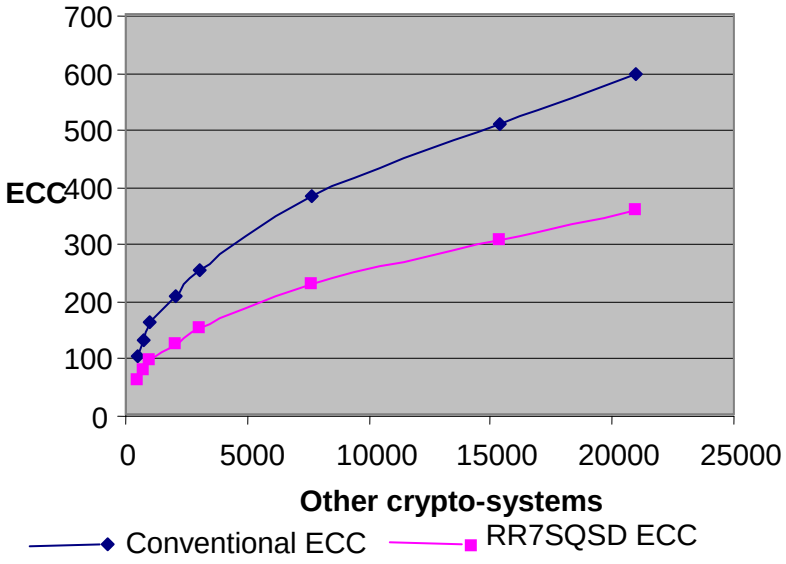
a) RR7SQSD characters versus Bits



b) Binary versus RR7SQSD



c) Decimal/RR7SQSD/binary



d) Comparative equivalent security level Key sizes in bits

Figure 3.1: Plots of RR7SQSD number system information and cryptographic content

The plot also show that an RR7SQSD based ECC will not contradict the widely acknowledged statement that ECC provides less key size for the same level of security than other non EC cryptosystems.

3.1.1.3 Nature of radix – 7 SDNS and the restriction

The nature of the radix-7 SDNS can therefore be described as follows. Any integer X **expressible as**

$$X \in \{x_{m-1}.x_{m-2}....,x_0\} = \begin{cases} \sum_{i=0}^{m-1} x_i r^i \text{ ; if } X \geq 0 \text{ or} \\ \sum_{i=0}^{m-1} -x_i r^i \text{, if } X < 0 \end{cases}$$

can be adequately represented within the symmetrical $r-nary$ **signed-character-digit set** $L=\{\bar{r},\overline{(r-1)},\overline{(r-2)},...,-2,-1,0,1,2,...,(r-2),(r-1),r\}$**. For example, lf** $|X|=89$ **and** $r=7$ **such that** $L=\{-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6\}$ **then** $89=155_7$ **while** $-89=\bar{1}\bar{5}\bar{5}$**. Thus the *radix – 7 symmetrical* SDNS represented by the symmetrical digit set** $L=\{-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6\}$ **providing a range of numbers from -** $48_{10}\le x\le 48_{10}\equiv -66_7\le x\le 66_7$ **gives 97 distinct numbers. Each number is possible of 13 different ways of representation if each element in the symmetrical digit set is replaced by two SD components (*$\alpha_{0j},\ \alpha_{1j}$*). With a total of 169 ways of representing any number that lie in the range** $-48_{10}\le x_{10}\le 48_{10}\equiv -66_7\le x_7\le 66_7$**, the *radix- 7* symmetrical SDNS fronts a redundancy percentage of 74% to provide an efficient carry-free radix – 7 or higher radix symmetrical signed digit arithmetic capability. However, owing to the reasons enunciated in section 3.1.1.3, in order to realize high speed efficient radix-7 signed digit arithmetic circuits with low power consumption and thus small in-circuit heat generation, it is necessary to embed the radix-7 signed digit arithmetic in lower radix-r signed digit arithmetic. Hence, in line with the discussion in section 2.2.1 and with** $h=7$**,** $r=\left(\frac{h+1}{2}\right)=4$ **and taking** $a=\left(\left(\frac{h+1}{2}\right)-1\right)=3$ **the system thus obtained is referred to in this as the Restricted Radix-7 Symmetrical Quaternary Signed Digit Number System (RR7SQSDNS). Consequently, the RR7SQSDNS is an appropriate SMVL processing threshold system.**

3.1.1.4 RR7SQSDNS arithmetic schemes

It was observed that the signed digit arithmetic schemes suggested in [41, 61] can be slightly modified for the RR7SQSD domain. Consequently, in this study, it is considered that with addition and multiplication operations as the major runtime consumers in secure communication systems and other compute constrained environment devices, a single circuit capable of performing either addition or multiplication operation will be of immense benefit. Formulation of such a scheme is presented next.

Let α_i and β_i be two one-RR7SQSD operands and τ_i , δ_i be the result of an arithmetic operation and the carry digit, then similar to equation (2.2) in section (2.2.2), equation (3.1) holds true.

$$\delta_i = \begin{cases} \bar{1} & ; If\ (\alpha_i \Theta \beta_i) < \bar{a} \\ 1 & ; if\ (\alpha_i \Theta \beta_i) > a \\ 0 & ; if\ otherwise \end{cases} \qquad (3.1)$$

$$\tau_i = (\alpha_i \Theta \beta_i) - 7\delta_i$$

where: the operator $\Theta \in \{+,-,*,/\}$, $\alpha_i, \beta_i, \tau_i \in \{-3,-2,-1,0,1,2,3\}$ **and** $\delta_i \in \{1,0,\bar{1}\}$**.The evaluation of equation (3.1) is as presented in appendix CH3.1. Equation (3.1) can also be used to convert decimal numbers to RR7SQSD compatible number systems and vice versa. For example, the entire 19 value radix-10 SDN system representation of the symmetrical digit set** $L \in \{\bar{9},\bar{8},....,\bar{2},\bar{1},0,1,2,.....,8,9\}$ **can be RR7SQSD coded as** $L \in \{\bar{1}\bar{2},\bar{1}\ \bar{1},\bar{1}0,\bar{1}1,\bar{1}2,\bar{1}3,\bar{3},\bar{2},\bar{1},0,1,2,3,1\bar{3},1\bar{2},1\ \bar{1},10,11,12\}$ **correspondingly, which of course**

included the 13-value radix – 7 SDN system symmetrical digit set of { $\bar{6},\bar{5},\bar{4},\bar{3},\bar{2},\bar{1},0,1,2,3,4,5,6$ **}. One drawback of equation (3.1) is that it did not take cognisance of carry-in from previous RR7SQSD arithmetic operation. Thus, equation (3.1) in terms of addition operation is an RR7SQSD XOR or an RR7SQSD half adder and in terms multiplication it is a one RR7SQSD multiplier.**

3.1.1.5 Arithmetic operations in an RR7SQSD multi-digit environment

To carter for the short fall of equation (3.1) and thus in multi RR7SQSD environment, carry-in from previous operation into the ith character digits operation has to be considered. If (α_i, β_i) , (δ_i, δ_{i-1}) and (z_i, τ_i, ,γ_i) are the *ith* digit pair of a radix *r* input operands, the carry–out and carry-in pairs and the linear, interim and final result digits set respectively, then the three operand arithmetic operations (RR7SQSD full adder) can be adequately described by equation (3.2).

$$z_i = \alpha_i \Theta \beta_i$$

$$\delta_i = \begin{cases} \bar{1} \text{ ; if } & z_i < \bar{a} \\ 1 \text{ ; if } & z_i > a \\ 0 \text{ ; if } & \bar{a} \le z_i \le a_i \\ \tau_i = z_i - 7\,\delta_i & \\ \gamma_i = \tau_i + \delta_{i-1} & \end{cases} \qquad \textbf{(3.2)}$$

provided i) $\alpha_i, \beta_i, \tau_i, \gamma_i \in \{\bar{3},\bar{2},\bar{1},0,1,2,3\}$ **and *ii)*** $\delta_i, \delta_{i-1} \in \{-1,0,1\}$**. The computation process is two-phase and its successful execution automatically means a carry-free addition operation in the second phase. Carry to the next stage is generated only in the first phase. Unfortunately, equation (3.2) does not ensure no-carry generation in the**

second phase for certain combination of input operands a problem similar to that enunciated in section 2.2.3.

For example, restricting to addition and multiplication operations such that $z_i = \alpha_i + \beta_i$ **or** $\alpha_i * \beta_i$ **:**

Case I. $z_i = 4$ ***and*** $\delta_{i-1} = \bar{1}$**, Expected result is** $\delta_i = 0$ ***and*** $\gamma_i = 3$**. However,** $\alpha_i \Theta \beta_i = 4 > 3$ $\delta_i = 1$**.** $\tau_i = z_i - r\,\delta_i = \bar{3}$ **and** $\gamma_i = \tau_i + \delta_{i-1} = \bar{3} + \bar{1} = 3$ **to give a carry out in the 2nd phase.**

Case II. $z_i = \bar{3}$ ***and*** $\delta_{i-1} = \bar{1}$ ***;what is expected are:*** $(\delta_i, \tau_i) = (\bar{1}, 3)$ ***and*** $\gamma_i = \tau_i + \delta_{i-1} = 3$ **result but what is obtained is:** $(\delta_i, \tau_i) = (0, \bar{3})$ ***land*** $\gamma_i = \tau_i + \delta_{i-1} = \bar{3} + \bar{1} = 3$**. The carry-out** $\delta_i = \bar{1}$**, comes in the second phase of the operation where it cannot be accounted for.**

Case III. $z_i = \bar{4}$ **and** $\delta_{i-1} = 1$**. The desired is:** $(\delta_i, \tau_i) = (0, 3)$ **and a** $\gamma_i = \tau_i + \delta_{i-1} = 3 + 1 = \bar{3}$ **result but since** $\alpha_i \Theta \beta_i = \bar{4}$ **so what one obtain is a** $(\delta_i, \tau_i) = (\bar{1}, 3)$ **and a** $\gamma_i = \tau_i + \delta_{i-1} =$ $3 + 1 = \bar{3}$**.**

Case IV. $z_i = 3$ **and** $\delta_{i-1} = 1$**. The expectation is a** $(\delta_i, \tau_i) = (1, \bar{3})$ **and a** $\gamma_i = \tau_i + \delta_{i-1} = 0 + \bar{3} = \bar{3}$ **but what one observed is a** $(\delta_i, \tau_i) = (0, 3)$ **and a** $\gamma_i = \tau_i + \delta_{i-1} = 3 + 1 = \bar{3}$ **.These cases are summarized in table 3.2**

It can be seen that one is either trying to stop a false "-1**" or** "+1" **propagating carry or re-instating an annulled "**-1**" or** "+1" **carry propagation. To identify potential carry generating operand pairs one have to consider the magnitudes of the previous operand**

pair $(\alpha_{i-1}, \beta_{i-1})$ and not just its carry-out, while performing the addition operation on the current digit pair (α_i, β_i) On a positive identification, the propagation of the fictitious-carry $\in (\bar{1},0,1)$ is

stopped or encouraged to generate a desired carry $\in (\bar{1},0,1)$ by injecting error correcting signal digits $(\tau'_{1i}, \tau''_{1i}) \in \{\bar{3},0,3\}$. The choice of a "-3" or a "+3" is determined by the sign of the interim sum τ_i. A zero choice means no correction is necessary. Consequently, a unified algorithm for large RR7SQSD parallel addition and multiplication operation can be developed. In deed equation (3.3) describes the multi-RR7SQSD addition/multiplication operation. The realization of a dual RR7SQSD full adder/multiplier circuit based on equation (3.3) will be explained later.

Table 3.2 Summarised analysis of the RR7SQSD Addition/Multiplication scheme.

Input	Desired result	Obtained Result	Suggested remedy $(\delta_{i,}, \delta_{i-1}, \gamma_i)$	Action taken	
				Particular $(\delta^f{}_i, \gamma_i)$	Generalized $(\delta^f{}_i, \gamma_i)$
$z_{i,}\delta_{i-1}{}^f$	$(\delta_i, \tau_i), \gamma_i$	$(\delta_i, \tau_i), \gamma_i$	$\delta_i, \delta_{i-1}, \tau_i$	$(\delta^f{}_i, \tau_i)$	$(\delta_{i-1}+\delta_i, \tau^f{}_i + \delta_{i-1})$
$4,\bar{1}$	$(0,3),3$	$(1,\bar{3}),3$	$0,1,3$	$(\bar{1},3)$	$(\delta_{i-1}+\delta_i, \tau^f{}_i + \delta_{i-1})$
$\bar{3},\bar{1}$	$(\bar{1},3),3$	$(0,\bar{3}),3$	$\bar{1},0,3$	$(\bar{1},3)$	$(\delta_{i-1}+\delta_i, \tau^f{}_i + \delta_{i-1})$
$\bar{4},1$	$(0,\bar{3}),\bar{3}$	$(\bar{1},3),\bar{3}$	$0,1,\bar{3}$	$(1,\bar{3})$	$(\delta_{i-1}+\delta_i, \tau^f{}_i + \delta_{i-1})$
$3,1$	$(1,\bar{3}),\bar{3}$	$(0,3),\bar{3}$	$1,0,\bar{3}$	$(1,\bar{3})$	$(\delta_{i-1}+\delta_i, \tau^f{}_i + \delta_{i-1})$

$$
\left.
\begin{aligned}
\delta_{1i} &= \begin{cases} \bar{1} & ; if \quad \alpha_i \Theta \beta_i < \bar{a} \\ 1 & ; if \quad \alpha_i \Theta \beta_i > a \\ 0; & \quad if\ otherwise \end{cases} \\
\tau_{1i} &= \alpha_i \Theta \beta_i - r\,\delta_{1i} \\
\delta_{2i} &= \begin{cases} \bar{1} & ; if \quad \tau_{1i} + \tau'_{1i} + \delta_{1i-1} < \bar{a} \\ 1 & ; if \quad \tau_{1i} + \tau'_{1i} + \delta_{1i-1} > a \\ 0 & ; if\ otherwise \end{cases} \\
\tau_{2i} &= \tau_{1i} + \tau^{l}_{1i} + \delta_{1i-1} - r\,\delta_{2i} \\
\delta_{3i} &= \begin{cases} \bar{1} & ; if \quad \tau_{2i} + \tau^{''}_{1i} + \delta_{2i-1} < \bar{a} \\ 1 & ; if \quad \tau_{2i} + \tau^{''}_{1i} + \delta_{2i-1} > a \\ 0 & ; if\ otherwise \end{cases} \\
\tau_{3i} &= \tau_{2i} + \tau^{''}_{1i} + \delta_{2i-1} - r\delta_{3i} \ ; \ \{\tau'_{1i}, \tau''_{1i}\} \in \{\bar{3}, 0, 3\} \\
\gamma_i &= \tau_{3i} + \delta^{f}_{i-1}; \qquad \delta^{f}_{i} = \delta^{f}_{3i} + \delta^{f}_{i-1} \\
&\textit{In all other cases} \\
\delta_i^{ff} &= \begin{cases} \bar{1} & ; if \quad \alpha_i \Theta \beta_i < \bar{a} \\ 1 & ; if \quad \alpha_i \Theta \beta_i > a \\ 0 & ; otherwise \end{cases} \\
\tau^{ff}_{1i} &= \alpha_i \Theta \beta_i - r\delta^{ff}_{i} \\
\gamma_i &= \tau^{ff}_{1i} + \delta^{ff}_{i}
\end{aligned}
\right\} \quad (3.3)
$$

The following example further elucidates the above explanation conveyed in equation(3.3)

Numerical Example 3.1 Let A = 54708330621626_{10}, B = 56845529339651_{10} **such that in the RR7SQSD number system** $\alpha = 2\bar{2}\bar{3}\bar{2}\bar{2}\bar{3}\bar{1}\bar{2}302\bar{3}0\bar{2}322$ **and** $\beta = 2\bar{2}0\bar{1}\bar{2}0\bar{2}\bar{2}132\bar{0}\bar{3}\bar{2}023$**. The addition processes using equation (3.2) and the algorithm of equation (3.3) are shown side-by-side in figure 3.2**

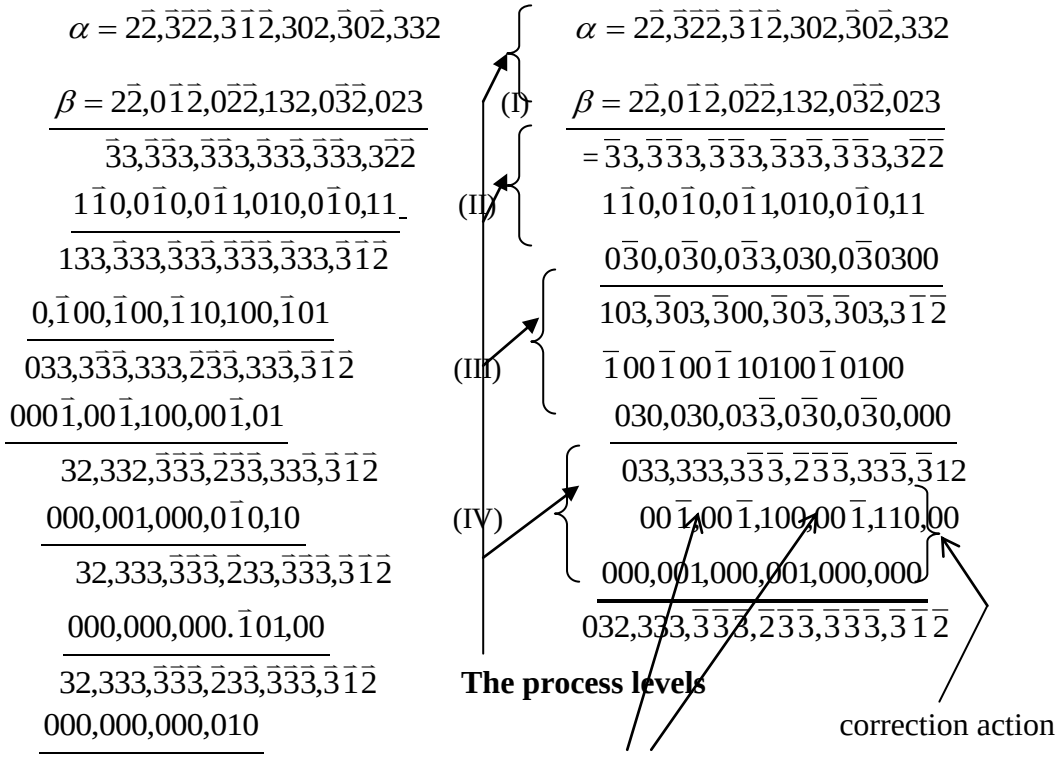


Figure 3.2: Solution to numerical example on RR7SQSD multi-digit addition

It can be seen from figure 3.2 that while 8 steps are required to arrive at the final solution when using the un-compensated method of equation (3.2) only four steps are required to arrive at the final result when using our algorithm of equation (3.3). In the later case from the right the first two columns do not satisfy the specified conditions so equation (3.2) is used to compute their sums.

However, a "1" carry generated from the 2nd column into the 3rd column where $\alpha_i + \beta_i = 3$ triggering a carry propagation chain made the 3rd column to qualify for action. The same condition applies in 4 , 8, 9, 10, 14, 15 and 17. With a ratio of *8 to 4* there is a saving of almost 50% in speed.

3.1.1.6 Multiplicative inverse computation in RR7SQSD domain

The RR7SQSD number system equally supports the computation of non-zero element's multiplicative inverse using either the prime modulus p approach that employs Fermat's last theorem $b^{-1} = b^{p-2} \bmod p$ provided $(p,b)=1$ or the Extended Euclidean Algorithm (EEA) approach [56]. However, in an RR7SQSD domain the EEA is more suitable. For example, working in RR7SQSDNS domain, to find $110^{-1} \bmod 2\bar{1}\bar{2}$ using Fermat's last theorem is to compute $110^{2\bar{2}\bar{3}} \bmod 2\bar{1}\bar{2}$. This is by far scarier than using the EEA method in [58]. This is evidenced in the solution to the numerical example 3.2 presented in figure 3.3.

Numerical example 3.2. To find $110^{-1} \bmod 2\bar{1}\bar{2}$ the using EEA in [64].

Solution: The computation is presented under the following headings

The Euclidean algorithm.	**The extended Euclidean algorithm**		**The Checking**
	1	0	
$2\bar{1}\bar{2} = (1)110 + 1\bar{2}\bar{2}$	0	1	
$110 = (1)1\bar{2}\bar{2} + 32$	1	$\bar{1}(1)$	$\bar{3}\,\bar{1}\,3(2\,\bar{1}\,\bar{2}) = 11\,\bar{1}\,\bar{1}\,1$
$1\bar{2}\bar{2} = (\bar{2}\,\bar{1})32 + \bar{2}0$	$\bar{1}$	$2(\bar{2}\,\bar{1})$	$\bar{1}\,01(110) = \bar{1}\,\bar{1}\,110$
$32 = (2)\bar{2}0 + 2$	$\bar{2}0$	$\bar{3}\,1(2)$	$R(x) = 1$
$\bar{2}0 = (\bar{1}\,3)2 + 1$	$\bar{3}\,\bar{1}$	$\bar{1}\,0(\bar{1}\,3)$	
	$\bar{3}\,\bar{1}\,3$	$\bar{1}\,01$	

Hence $\bar{1}\,01 = 110^{-1} \bmod 2\bar{1}\bar{2}$.

Figure 3.3: Solution to numerical example on EEA

The EEA approach shall be adopted in all RR7SQSD finite field element inverse computations. The methodology of such computational process is presented in section 3.2.2.2 .

3.1.1.7 An RR7SQSDN application:-A multiply-by-7 Addition/subtraction chain

It has been reiterated that one way of speeding up EC point multiplication is to use addition-subtraction chain. Addition/ subtraction chains compute the product of two integers in an unconventional manner. The foregoing starting with brief explanation of conventional addition chain shows that RR7SQSDNS also support addition- subtraction chain computation.

Definition 3.3. **An addition chain [72 - 73,94 - 100] is a sequence of integers** $B_0, B_1, B_2, ..., B_\lambda, ..., B_{n-1}$ **starting from** B_0 **and ending with** $B_{n-1} = K_r$ **such that** B_λ **is the sum of two previous members** B_i **and** B_j **of the chain.**

That is, If the sequence $B_0\ B_1\ B_2$**,….,**B_l**, ……,**B_{n-1}**; with** $B_l \in \{b_0, b_1, .., b_\mu, ..., b_\lambda\}$ **and** $b_\mu \in \{0,1,2,..., r-1\}$ **is an addition chain of the integer** K_r **in the radix-r positional number system and** (B_i, B_j) **are two earlier elements of the sequence** $\{B_l\}$**, then**

$$B_l = \begin{cases} B_0 \quad \text{; if } l = 0 \\ B_i + B_j \text{ ;if } 0 < l, i, j < n-1 \\ B_{n-1} = K_r \text{ ; if } l = n-1 \end{cases} \qquad \textbf{(3.4)}$$

The integer K_r, may present several lengths of different addition chains in the same radix r number system. For example if K = 67 and r = 10 then some of the chains are:

1,2,3,5,7,10,11,21,31,52,62,67

1,2,3,4,6,9,10,16,25,29,54,64,67

1, 2, 4, 8,16,33,67 and

1,2,3,5,8,11,19,30,49,60,65,67.

Addition chains provide a method for computing mP **exponentiation. Thus given the integer** $m = 67$ **then** mP **becomes**

P 2P 3P 5P 8P 11P 19P 30P 49P 60P 65P 67P

When *m* and P **are both scalar quantities, then** B_{n-1} **in equation (3.4) is their product and the length of the addition chain gives the number of operations required computing the product** mP**. There had been very good works on binary, octal, ternary or binary signed-digit, sliding window, factor, power tree and other forms of conventional addition-subtraction chains [72-73] but none on RR7SQSD addition chains.**

Definition 3.4: **Let the string** $\alpha_{n-1}, \alpha_{n-2}, \alpha_{n-3}, ..., \alpha_j, ...\alpha_0$ **be the RR7SQSD representation of the integer** K **and the sequence** $\psi_0, \psi_1, \psi_2, ..., \psi_j, ..., \psi_{\lambda-1}$ **its RR7SQSD addition/subtraction chain then an RR7SQSD addition/subtraction chain can be represented by equation (3.5)**

$$\psi_j = \begin{cases} \psi_0 \in \{1,2,3\}; \text{ if } \; j = n-1 \\ \\ 7\psi_{j-1} + \alpha_{n-j} \quad ; \text{ if } \; n-2 \le j \le 0 \end{cases} \qquad \textbf{(3.5)}$$

where: $\alpha_i \in \{-3,-2,-1,0,1,2,3\}$ **and** $\alpha_{n-j} \in \{0,1,2,3\}$ **being the value to be added or subtracted from** $7\psi_{j-1}$ **base on the RR7SQSD analyzed. Equation (3.5) in effect means that RR7SQSD addition/subtraction chain is an addition/subtraction chain based on a principle of multiplying a previous element of a sequence by 7 and adding or subtracting RR7SQSD**

multiples of the multiplicand. The RR7SQSD multipliy-by-7 addition/subtraction chains require fewer number of operations compared to [72] or [73] and can be used for a wide variety of applications such as the multiplication of operands of very long word length in a data word constrained compute environments and for computing elliptic curve point multiplication.

3.1.1.8 Numerical example on RR7SQSD addition/subtraction chain computation

The numerical example 3.3 in figure 3.4 further explains the principle of the proposal.

Numerical example 3.3. It is required to compute the shortest addition chains of the following numbers 92,155, 3038,7985 and 434679.

Solution: Using equation (3.5), the chains are as shown in figure 3.4

K	α_k	**Multiply-by-7 addition/subtraction chain**
93	**2,-1, 2**	**2, 13, 93**
155	**3, 1, 1**	**3, 22,155**
3038	**1, 2,1,0,0**	**1, 9,65,434,3038**
7985	**3, 2,2,02**	**3, 23, 165, 1141, 7985**
434679	**1,-3,-2,-1, 0,2,0,0**	**1, 4,26,181,1267,8871,62097,434679**
142897	**1, 2,-3,-3,-3, 0,-1**	**1,9,60,417,2916,20414,142897**
6984572	**1, 1, 3, 3,-3, 0,1,2,0**	**1,59,413,2909,30363,142542,997796,6984572**

Figure 3.4: Solution to numerical examples on RR7SQSD addition/subtraction chains.

Having formulated the principle and illustrated by way of the numerical example 3.1 to 3.3, that the RR7SQSDNS supports very efficient the arithmetic the proof of its ability to extend this property to arithmetic operations of practical significance is presented in section 3.2.1

3.1.2 Mathematical model of the algorithm for arithmetic operations on EC defined over $SGF(7^m + \psi_i)$

The presentation of this section is aimed at establishing the theoretical model for the design and testing of a hardware-implementable algorithm that will withstand the mathematical rigour of tested PKC protocols and in particular those cryptosystem protocols that can be used on elliptic curves defined over RR7SQSD element finite fields. To achieve this objective it is necessary to: i) show that RR7SQSDNS is equally an algebraic structure or field (referred to as the RR7SQSD elements finite field), ii) outline the principle of constructing an RR7SQSD finite field and iii) define the various field arithmetic operations These points are presented in sections 3.1.2.1 to 3.1.2.3 respectively.

3.1.2.1 RR7SQSD finite field basic axioms

Definition 3.2: RR7SQSD finite field. **An RR7SQSD finite field $F_{RR7SQSD}$, designated $SGF(7)$ is an order-7 Galois field with elements in $L \in \{-3,-2,-1,0,1,2,3\}$ complete with addition and multiplication operations defined.**

Axiom 3.3. The elements of $SGF(7)$: **The character digit set $L \in \{-3,-2,-1,0,1,2,3\}$ is an RR7SQSD domain or a multiple-valued logic variable recoded rearrangement of the elements of the prime field $GF(7)$. Recoding the elements of $GF(7)$ in the RR7SQSD character digit set does not in any way adversely affect the finite field arithmetic properties.**

Axiom 3.4. **The basic arithmetic operations in $F_{RR7SQSD}$, are similar to those of conventional binary Galois fields, closed and obey the commutative, associative and distributive laws.**

Axiom 3.5 ***RR7SQSD field basic arithmetic operations:*** **All basic arithmetic operations of the RR7SQSD ground field $SGF(7)$, addition, subtraction, multiplication, multiplicative and thus division are closed and can be executed using equation (3.1) where '0' and '1' are the respective additive and multiplicative elements.**

This is the foundation basis of all subsequent larger RR7SQSD finite field arithmetic hence, for convenience reproduced as in equation (3.6).

$$\delta_i = \begin{cases} \bar{1} & \textit{;if } (\alpha_i \Theta \beta_i) < \bar{a} \\ 1 & \textit{;if } \alpha_i \Theta \beta_i > a \\ \mathbf{0} & \textit{; if otherwise} \end{cases} \qquad \tau_i = (\alpha_i \Theta \beta_i) - 7\delta_i \qquad \textbf{(3.6)}$$

where: the operator $\Theta \in \{+,-,*,/\}$, $\alpha_i, \beta_i, \tau_i \in \{-3,-2,-1,0,1,2,3\}$ **and** $\delta_i \in \{1,0,\bar{1}\}$**. The evaluation of the basic arithmetic functions of $SGF(7)$ using equation (3.6) is as shown in appendix CH3.1**

Axiom 3.6 **If x is a generator set of a conventional prime finite field $GF(q)$, the generator set y of $SGF(q)$ is x re-coded in the RR7SQSD character digit set. Alternatively, a generator set $\{\alpha_1, \alpha_2, ...\alpha_i, ..\alpha_n\}$ for $GF(q)$ is also the generator set but recoded for $SGF(q)$.**

For example, the generators of $GF(7)$ are 3 and 5 are the same for $SGF(7)$ but recoded in the SMVL variables as 3 and -2 respectively.

Axiom 3.7 RR7SQSD Extension fields $SGF(p^k)$**: If an** $SGF(q)$ **exist for any ground field** $GF(q)$ **then the extended** $SGF(p^k)$ p **a prime, also exist for every** $GF(p^k)$ **and the RR7SQSD string word elements are direct recoding using the RR7SQSD arithmetic scheme for the elements of** $GF(p^k)$ **just as all nonzero elements of** $SGF(p^k)$ **are generated by a primitive element α where α is the root of a primitive irreducible polynomial**

$$F(x) = x^k + \sum_{j=k-1}^{0} f_j \alpha^j \quad \text{.....(3.7)}$$

Axiom 3.8 $SGF(p^k)$ ***elements representation:*** **All nonzero elements of the field** $SGF(p^k)$ **are points on the field and therefore can be represented as:**

i) Vector basis i.e.; as powers of α that $\alpha^k = \sum_{j=k-1}^{0} f_j x^j$ **provided F(α) = 0.**

ii) in polynomial or normal basis form, for example the element point

$$p(x,y) = (1\bar{2}31\bar{2}, 23\bar{1}0\bar{2}) = (x^4 - 2x^3 + 3x^2 + x - 2, 2x^4 + 3x^3 - x^2 - 2)$$

iii) in integer representation as $= p(1867,5780)$

Axiom 3.9 RR7SQSD Optimal extension fields SGF(7^k+ψ): **If the normal prime field** $GF(p)$ **exists and there are two integers α,** $\beta \in p$ **such that α + β = *p* and αβ < *2p* then, there is a restricted radix – h symmetrical** $\left\lceil \frac{h-1}{2} \right\rceil - nary$ **signed digit element optimally extended finite field** $SGF(h)$ **over the ground field** $GF(p)$ **where** h **is any prime from the sequence** $\{p^k \pm i\alpha\beta\}$**. Alternatively it can be said that,** $\forall\ GF(p)$, $\exists\ \ SGF(p^k \pm i\alpha\beta) = SGF(h)$ **provided** $\alpha, \beta \in p$ **, α + β = *p* and αβ < *2p*** **and for** $j < i$ $SGF(h_i)$ **is concentric to** $SGF(h_j)$**.**

$SGF(h)$ **is essentially a vector space and higher order *h* hardware circuits can be realized by simple cascading of the circuits of lower order *h*. For example, the fields** $SGF\left(h \in \{13,37,61,79,109,157,181,241,.....\}\right)$ **are concentric. That is blocks of** $SGF(7)$ **can easily form a block of** $SGF(13)$**. Similarly, blocks of** $SGF(13)$ **in turn can form a block of** $SGF(37)$ **which in turn can form a block of** $SGF(61)$ **and so on. Some 134 special primes generated, are as shown in appendix CH3.2.**

3.1.2.2 Principles of constructing RR7SQSD elements finite fields

The existence of SD element Galois fields can be demonstrated by constructing SGF(7^m) as follows: Let x **and *l*, in the RR7SQSD number system be a single independent variable and coefficient respectively such that**

$x, l \in \left\{\left(\overline{a-1}\right), \left(\overline{a-2}\right), ..., \bar{2}, \bar{1}, 0, 1, 2, ..., (a-2), (a-1)\right\}$ **with** $a \geq \left\lceil \frac{h-1}{2} \right\rceil$ **then the following can be define thus**

i) **A *degree m* SD polynomial Q(x) as**

$$Q(x) = \sum_{i=0}^{m-1} l_i x^i \text{ .} \qquad \textbf{(3.8)}$$

ii) **A primitive SD polynomial is the least SD polynomial primitive element α, in** $SGF(p^m)$ **provided** $\alpha \in \left\{\left(\overline{a-1}, ..., \bar{2}, \bar{1}, 0, 1, 2, ..., (a-1)\right)\right\}$**.**

iii) **If *a, is an element* and *α*, a root *of* $SGF(p^m)$, then a_i can be represented as $\left\{\overline{\alpha}^{p^{m-2}},...,\overline{\alpha}^{2},\overline{\alpha},\overline{1},0,1,2,...\alpha^{p^{m-2}}\right\}$. Reducing these elements by a reduction modulo polynomial help to represent elements of $SGF(p^m)$ in polynomial basis.**

iv) **A SD irreducible or monic polynomial is any non-constant SD polynomial that cannot be factored.**

Thus, taking *p(x)* as an irreducible (monic) signed digit polynomial [55] of degree *m* over F_7 such that

$$p(x) = x^m + \sum_{i=1}^{m} a^{m-i}x^{m-i} : \ |a_i \in \{\overline{3},\overline{2},\overline{1},0,1,2,3\} \ \textit{for i= 0,1,2,…,m-1}\ ) \qquad \textbf{(3.9)}$$

And referring to the universal set of such polynomials *p(x)* of degree less than *m* over F_7 as a signed digit element finite field F_7^m , designated with SGF(7^m), it can proved that:

SGF(7^m) **= {$a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_2x^2 + a_1x^1 + a_0 : a_i$, x Є {-3, $\overline{3},\overline{2},\overline{1},0,1,2,3$} } (3.10).**

Similar to positive element finite fields, the elements of $SGF(7^m)$ are represented by strings of length *m* RR7SQSD so that $SGF(7^m)$ can be alternatively represented as

SGF(7^m) **= {($a_{m-1}a_{m-2}a_{m-3}\ldots\ldots a_1a_0$): a_i Є {$\overline{3},\overline{2},\overline{1},0,1,2,3$} } (3.11)**

Consequently, It can be said that a $SGF(7^m)$, $m > 0$ contains 7^m elements each of which can be uniquely represented with a degree up to *m-1* RR7SQSD polynomial *p(x)* in the form:

p(x) **= $a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots\ldots\ldots + a_2x^2 + a_1x^1 + a_0 : a_i$, x^i Є {$\overline{3},\overline{2},\overline{1},0,1,2,3$}.**

The methodology for constructing $SGF(7^m)$ is presented in section 3.2.2.2

3.1.2.3 RR7SQSD elements finite field $SGF(7^m)$ arithmetic

For design compatibility the scheme of arithmetic operations in an RR7SQSD element finite field is made to follow the similar pattern of conventional prime Galois fields and are as formulated thus. Let a(x) and b(x) be two RR7SQSD finite field elements in the polynomial basis representation and w(x) be a monic irreducible polynomial, such that

$a(x)=\sum_{j=0}^{k-2} a_j x^j$ **,** $b(x)=\sum_{i=0}^{i-1} b_i x^i$ **and** $w(x)=\sum_{\tau=0}^{n} w_\tau x^\tau$ **where** **:** $k,l<n$

$a(x),b(x),w(x)\in SGF(7^m)$ **then;**

(a) $SGF(7^m)$ **addition**

RR7SQSD finite field elements addition is performed RR7SQSD coefficient component coefficient-wise with ignored carries as described in equation (3.12)

$$\mathbf{a(t) + b(t) = c(t) = (\sum_{i=0}^{n-1} c_i t^i) \bmod w(t)} \qquad \mathbf{(3.12)}$$

where $c_i = a_i + b_i - 7\delta_i$ and δ_i as earlier defined.

(b) $SGF(7^m)$ **multiplication:**

RR7SQSD finite field multiplication is described as in equation (3.13) where the multiplier *a(t)* and multiplicand *b(t)* element operands are represented in RR7SQSD polynomial basis.

$$\mathbf{a(t) * b(t) = a(t) . b(t) \bmod w(t) = [\sum_{i=0}^{k-1} a_i . b(t)] \bmod w(t)} \qquad \mathbf{(3.13)}$$

$= (a_{k-1}t^{k-1}(b_rt^r + b_{r-1}t^{r-1} + \ldots b_1t + b_0) + a_{k-2}t^{k-2}(b_rt^r + b_{r-1}t^{r-1} + \ldots b_1t + b_0)$

$+\ldots + a_1t(b_rt^r + b_{r-1}t^{r-1} + \ldots b_1t + b_0) + a_0(b_rt^r + b_{r-1}t^{r-1} + \ldots b_1t + b_0))$ **mod w(t)**

$$= \{\sum_{n=0}^{k-1} a_n t^{k-n}(b_{r-1}t^{r-1} + ... + b_1t + b_0) \bmod w(t)\} \bmod w(t) \qquad \textbf{(3.14)}$$

From equation (2.4) in section 2.2.1

let $d_i = a_{k-n}b_{r-m} - 7\rho^0_h$ **then**

$$\textbf{a(t) * b(t)} = \{\sum_{i=0}^{k+r-1} d_i t^i \textbf{ mod w(t)}\} \textbf{mod w(t)}$$

$$= \{\sum_{i=0}^{k+r-i}(\sum_{n=0,m=0}^{n=k-1,m=r-1} a_{k-n}b_{r-m} - 7p^0{}_{mn}) \textbf{ mod w(t)}\} \textbf{ mod w(t)} \qquad \textbf{(3.15)}$$

where $\rho^0_{nm} = \rho^0_{ij}$ **as defined in equation (3.4).**

The operands are multiplied RR7SQSD component wise with ignored product digit carry ρ^0_{nm} **and computing the residue modulo some given irreducible polynomial w(t). The design of an RR7SQSD field element multiplication circuit will be presented in section 3.6.1.3.**

(c) $SGF(7^m)$ **squaring**

Polynomial squaring operation is linear and is modulo-RR7SQSD

$$(\sum_{j=0}^{m-1} b_j x^j)^2 = (\sum_{j=0}^{2(m-j)} b_l x^{2(l)}) \bmod w(x) \qquad \textbf{(3.16)}$$

(d) $SGF(7^m)$ **exponentiation**

A Field exponentiation operation is multiplying copies of the element. i.e.;

$$(b_{j-1}b_{j-2}....b_1b_o)^t = \{(b_{j-1}b_{j-2}.....b_1b_o)_1.(b_{j-1}b_{j-2}....b_1b_o)_2.....(b_{j-1}b_{j-2}.....b_1b_o)_t\} \bmod w(x) \quad \textbf{(3.17)}$$

(e) $SGF(7^m)$ **multiplicative inverse and field element inversion**

A multiplicative inverse $b^{-1}(x)$ **of** $b(x) \in SGF(7^m)$ **must satisfy the equation** $b(x).b^{-1}(x) = 1$**. The efficient algorithm presented in [60] can equally be adopted for RR7SQSD finite field's element inverse computation. If** $a \in SGF(7^k)$ **and** $a \neq 0$ **then by factorization** $a^{-1} = a^{7^k-7} = a^{7[7^{k-1}-1]} = \left[a^{7^{k-1}-1}\right]^7$ **such that**

$$7^{k-1}-1 = \begin{cases} \left[7^{(k-1)/2}-1\right]\left[7^{(k-1)/2}+1\right] & \textbf{; If k is odd} \\ 7\left[7^{(k-2)/2}-1\right]\left[7^{(k-2)/2}+1\right] & \textbf{; If k is even} \end{cases} \quad \textbf{(3.18)}$$

Since $SGF\left(\left(7^{23}\right)^{19}\right)$ = $SGF(7^{667})$**, the number of field multiplication operations necessary to compute the inverse of an element in** $SGF\left(\left(7^{23}\right)^{19}\right)$ **may be determined as shown in figure 3.9. It can be deduced from this figure that if a** "+1" **signify RR7SQSD field multiplication operation and** "+6" **signify a 'multiply-by-7 operation then only 8+4 = 12 multiplication operations are required as against 14 when doing the same calculation in** $GF\left(2^{667}\right)$**.**

$$7^{667} - 7 = 7\left(7^{333} - 1\right)\left(7^{333} + 1\right)$$

$$7^{333} - 1 = 7\left(7^{166} - 1\right)\left(7^{166} + 1\right) + 6$$

$$7^{166} - 1 = \left(7^{83} - 1\right)\left(7^{83} + 1\right)$$

$$7^{83} - 1 = 7\left(7^{41} - 1\right)\left(7^{41} + 1\right) + 6$$

$$7^{41} - 1 = 7(7^{20} - 1)\left(7^{20} + 1\right) + 6$$

$$7^{20} - 1 = \left(7^{10} - 1\right)\left(7^{10} + 1\right)$$

$$7^{10} - 1 = \left(7^{5} - 1\right)\left(7^{5} + 1\right)$$

$$7^{5} - 1 = 7\left(7^{2} - 1\right)\left(7^{2} + 1\right) + 6$$

Figure 3.5: Application of RR7SQSDNS in finite field element inverse computation

Procedure for computing of RR7SQSD element finite field multiplicative inverse computation is presented in section 3.2.2.3.

3.1.3 Formulation of the VLSI's basic building block circuit

The formulation of the SMVL VLSI circuit basic building block circuit begins with its design specification is presented in section 3.1.3.1. The SMVL processing threshold signal profile and the control lines are presented in section 3.1.3.2 while the mathematical formulation of the SMVL ECC arithmetic unit VLSI functional logic circuits' basic building block is presented in section 3.1.3.3. Positive and negative literal CP – gate circuits are presented in section 3.1.3.4. The characteristic equation of the basic building block is presented in section 3.1.3.5.

3.1.3.1 The basic building block circuits design specifications

Expectations of the basic building block is that, it must operate on voltage mode and be capable of multiplexing 7- RR7SQSD input profile signals to a single output point based on an RR7SQSD input control signal without signal voltage degradation, signal distortion, and fidelity tamper. Precisely, it is to functions as a SMVL signal voltage multiplexing circuit. The circuit in view can be formulated using the principle of tree-type universal logic modules and complementary pass gates.

3.1.3.2 The RR7SQSD logic system's processing signal's profile and control line

This study proposes the RR7SQSD logic system in which the processing logic threshold values lie in the range $L \in \{-3,-2,-1,0,1,2,3\}$ as an alternative to binary logic which can replace it in all applications where binary logic holds sway. The RR7SQSD Logic is a Symmetrical Multiple-Value Logic (SMVL) system of the form $20\bar{1}321\bar{1}3\bar{1}0\bar{2}123\bar{1}\bar{1}021\bar{0}3\bar{3}$.

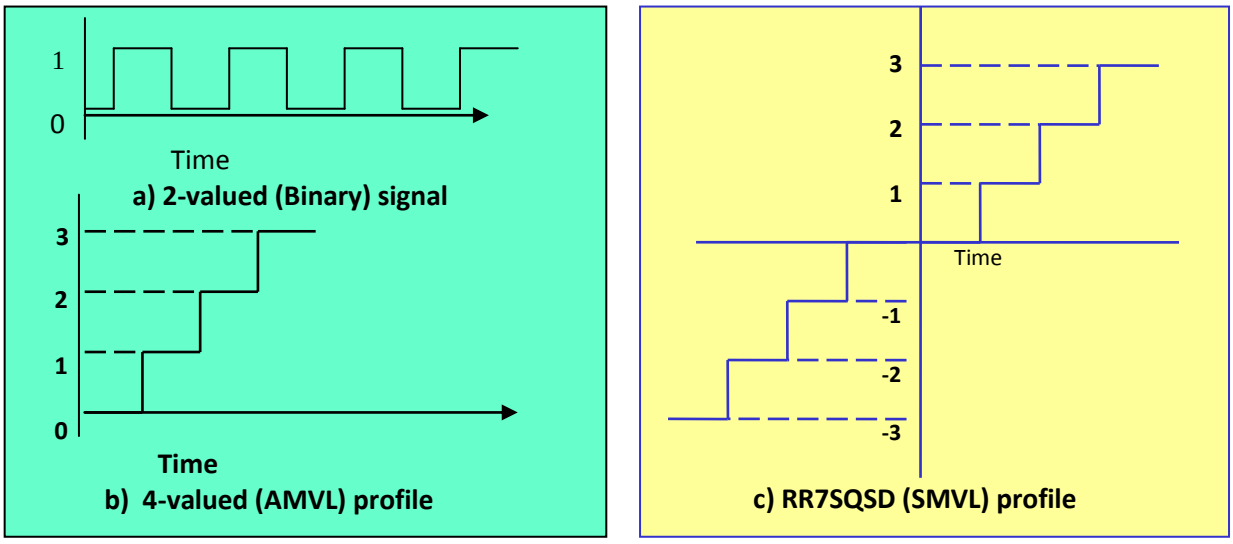


Figure 3.6: Processing logic signals' profiles

Fig.3.6 shows comparatively the signal profiles of binary or 2-valued logic, an asymmetrical 4-value otherwise quaternary logic (AMVL) and the 7-valued otherwise RR7SQSD logic (SMVL). The richness of the SMVL processing threshold logic is also depicted in the nature of the corresponding control lines necessary to address a given number of input variables. Consider the number of control lines required to implement 7 arbitrary input functions as shown in figure 3.7. A binary logic system requires an 8 to 3 lines encoder to adequately address the 7 input signals; a quaternary system requires a 2–strand wire bus width for the same purpose. The RR7SQSD system needs just a single strand wire.

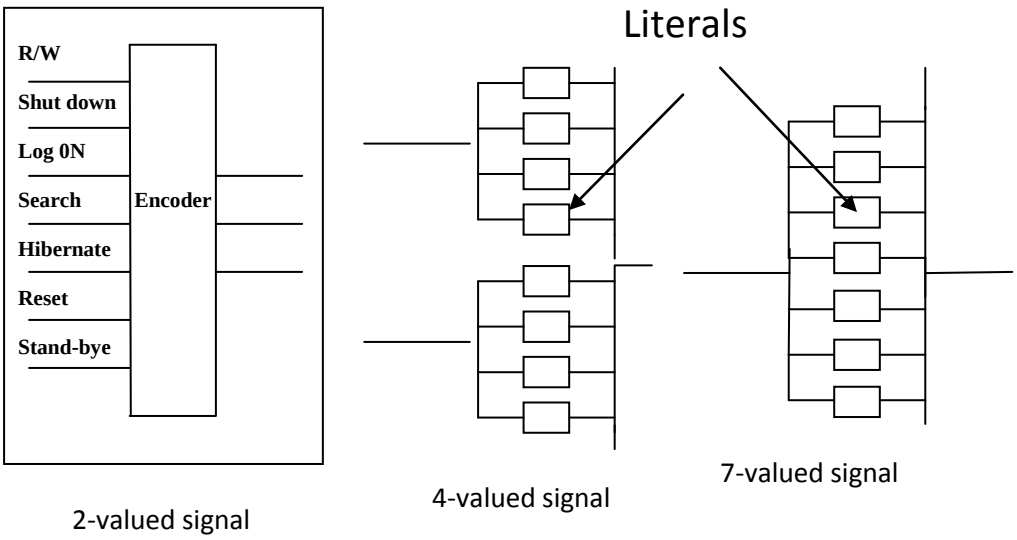


Figure 3.7: Comparison of processing logic threshold control signal lines

3.1.3.3 The restricted radix - h symmetrical r-nary Tree-type Universal Logic Module

A Tree-type Universal Logic Module (*T _ULM*) generalizes the principle of operation of a MVL basic building block [37]. In a T_ULM, let L = {0,1,2,……, h-1} be the

finite set of h **logic threshold values in the** $h-valued$ **logic system and the sequence** $\{x_k\}$ **be the set of elements of a control variable X with** h^k **distinct states. If** $A = (a_0, a_1, a_2, \ldots a_i, \ldots, a_{h^k-1})$ **be some set of residue input functions, then the T-ULM output as shown in figure 3.8 is described by the expression**

$$T_{ULM\,out} = \Phi(A; X) = \Phi(\{a_i\}; \{x_j\}) = \{a_i\}; \text{ if } x_j = i \qquad (3.19)$$

where $i = 0,1,2,3,\ldots, h-1$ **and** $j = 1, 2, 3$

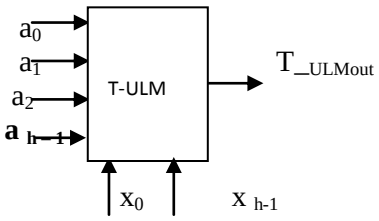


Figure 3.8: Functional block of the T_ULM

When the control sequence $\{x_j\}$**, of a T_ULM is just a single variable MVL signal such that** $x_1 = x_2 = \ldots = x_k = x$ **, then the T -ULM is simply referred to as an** $h-valued$ **T-gate defined by equation (3.20)**

$$T_{OUT} = T_{ULM} = \Phi(\{a_i\}; x) = a_i \text{ ; if } x = i \qquad (3.20)$$

Axiom 3.11 If the finite set of the logic threshold values L = {0,1, 2 ,….., h – 1) transforms into a new set $L' = \{[r-k]\}$ **and** $k \in \{h, h-1, h-2, \ldots, 1\}$**; such that** $\{a_i\}, \{x\}, i \in \{(\overline{r-1}), (\overline{r-2}), \ldots, \bar{2}, \bar{1}, 0, 1, 2, \ldots, (r-2)(r-1)\}$ **then the** T_ULM **is referred to as an** $h-valued$ **restricted symmetrical r- nary signed-digit** T_ULM **otherwise called the** $h-valued$ **restricted symmetrical r-nary signed digit** T_gate **.**

3.1.3.4 Positive and negative literal complementary pass gate transistor circuits

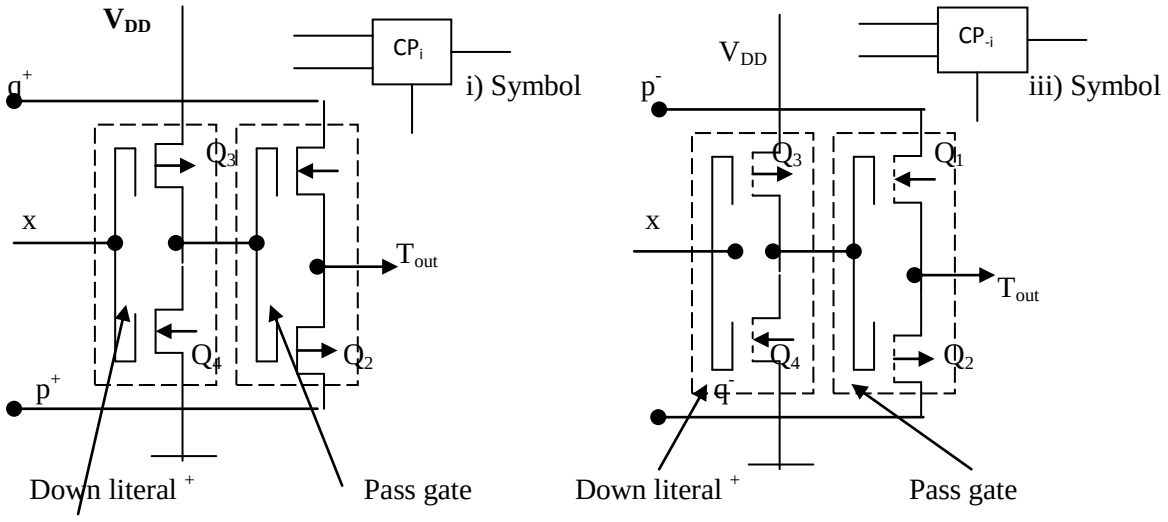
RR7SQSD CP-gate derived T-gate is a concatenation of positive literal threshold CP gate tree-type network [41] and negative literal threshold CP gate tree-type network into one integration. A positive CP pass gate is of positive logic threshold voltage, responding if and only if there is a positive input control voltage to pass r-valued negative or positive input data/address/ system control signal voltages to output. Similarly, a negative CP gate is of negative logic threshold voltages responding if and only if there is a negative r- valued input control signal and transmit the r-valued positive or negative data/ address/control signal voltages to output The positive and negative CP gates are fabricated of CMOS pass transistors (the transmission gates) with differently signed thresholds voltages obtained by virtue of surface dopant control through selective ion implant and correspondingly literal circuits. An interconnection of the positive and negative CP gates, made into a single integration will result in a new device with positive and negative CP gates characteristics and inheritance very wide spectrum of application. Fig. 3.9 shows the various aspects of positive and negative CP-gates.

The parallel connected transistors $Q_{1(a \text{ and } b)}$, $Q'_{1(a \text{ and } b)}$ constitute the respective pass transistors and the series connected $Q_{2(a \text{ and } b)}$, $Q'_{2(a \text{ and } b)}$ make up for the required down and up literals for the positive and negative CP gates. If the magnitude of x is lower than the threshold of the literal circuit, then transistor Q_{1a} /Q'_{1a} switches ON and Q_{1b}/Q'_{1b} is OFF and p is transmitted from input to output, on the contrary Q_{1b}/Q'_{1b} is ON while Q_{1a} /Q'_{1a} is switched OFF and q is passed to the output. Thus, the analogue

switch transistor circuits of both CP gates switch alternatively, according to the sign and magnitude of the control signal x, at the common gates of the literal circuits connected in series to the power supply V_{DD} or $-V_{DD}$. The modes of their respective operations are as described in the definitions of literal and CP-gate equations of (3.21) and (3.22).

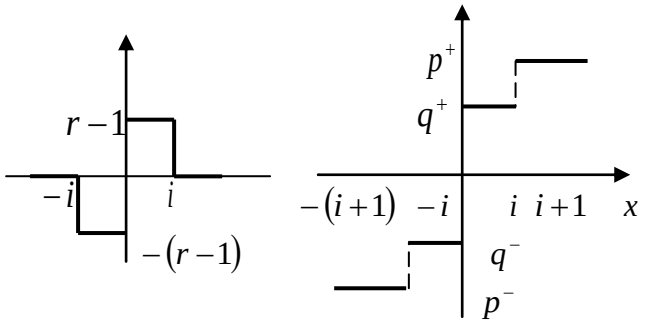
Definition 3.5 **The down literal $D_i(x)$ and the CP -gate CP_i are defined as**

$$D_i(x) = \begin{cases} r-1 \ ; \text{if } 0 < x < i \\ \\ 0 \quad ; \text{if } (i+1) < x < (r-1) \end{cases} \quad ; CP_i(p,q;x) = \begin{cases} p \ ; \text{if } x < i \\ \\ q \ ; \text{if } x > (i+1) \end{cases} \qquad (3.21)$$



Circuit diagram of the positive type

iv) Circuit diagram of the negative type



control	Pass gate		output
x	Q_1	Q_2	O/p
x < i	ON	OFF	P^-/q^+
x > i+1	OFF	ON	$q^-/p+$

i = v_{th}

v) Down Literal versus control x **vi) CP output versus control x** **vii) truth table**

Figure 3.9: Mono polar CP-gates.

Definition 3.6 **A negative literal D_i (-x) and its corresponding CP- gate CP_{-i}, circuits may be defined as follows:**

$$\mathbf{D_{-i}(-x)} = \begin{cases} \mathbf{0;\ if}\ \bar{i} < |x| < 0 \\ \mathbf{-(r-1);\ if}\ -(r-1) < |x| < \bar{i} \end{cases} \qquad \mathbf{CP_{-I}(p,q,-x)} = \begin{cases} \mathbf{p;\ if}\ -i < |x| < i \\ \mathbf{q\ ;\ if}\ -(r-1) < |x| < (r-1) \end{cases} \qquad \mathbf{(3.22)}$$

Where p, q are input signals and x is a control. Besides, i = {0,1,2,….,r-2}and p, q, x ∈ L = {0,1,2,……..,r – 1}.

CP – gates are mono-polar voltage mode operating multiplexer units that meet the required electrical parameters in section 3.1.3.2.1. Consequently, a parallel interconnection of the positive literal CP-gate and the negative literal CP-pass gate will function according to the specification required of a SMVL VLSI circuit.

3.1.3.5 The RR7SQSD CP-gate derived T – gate equation

An RR7SQSD CP – gate derived T – gate may be described as follows:

Rewriting equation (3.19) in standard multiplexing function (conditioned disjunction) form for values of $h = 7$ **such that** $L = \{0,1,2,3,4,5,6\} \equiv L' = \{-3,-2,-1,0,1,2,3\}$

Then $T_{out} = T(a_0, a_1, a_2, a_3, a_4, a_5, a_6; x)$

$$\equiv T'(a_{-3}, a_{-2}, a_{-1}, a_0, a_1, a_2, a_3; x) = a_i \ \textbf{;if}\ x = i \qquad \textbf{(3.23)}$$

where $i, \{a_i\}, x \in \{\bar{3},\bar{2},\bar{1},0,1,2,3\}$

Equation(3.23) represents the functional description of the 7-valued or Restricted Radix -7 Symmetrical Quaternary Signed Digit (RR7SQSD) T – gate. An alternative form of equation (3.23) depicting the quaternary nature of the T-gate is as presented in equation (3.24)

$$T(p_a, q_b, r_c, l_d; x) = \psi_m \quad \text{; if } x = m \qquad \textbf{(3.24)}$$

such that $m \in \{a,b,c,d\}$, $\psi \in \{p,q.r.l\}$ **and** $p,q,r,l,x \in \{\bar{3},\bar{2},\bar{1},0,1,2,3\}$ **and in particular,** $a \in \{\bar{3},3\}$, $b \in \{\bar{2},2\}$, $c \in \{\bar{1},1\}$ **and** $l \in \{0\}$

Definition 3.7 **Let** $a_i \in \{a_{-3}, a_{-2}, a_{-1}, a_0, a_1, a_2, a_3\}$ **be some SMVL residue input functions and** x, **a single SMVL control signal variable such that** $a_j, x, j \in \{-3,-2,-1,-0,1,2,3\}$**. Let** T, T_{OUT} **be the multiplexing function and the T-gate output functions respectively. Then the operation of an RR7SQSD T-gate is described by equation (3.25).**

$$T_{out} = T(a_{-3}, a_{-2}, a_{-1}, a_0, a_1, a_2, a_3; x) = a_j \quad \textbf{if } x = j \qquad \textbf{(3.25)}$$

Or

$$\mathbf{T\{p_a, q_b, r_c\ l_d; x) = P_m \text{ ; if } x = m} \qquad \textbf{(3.26)}$$

such that m ∈ {a,b,c,d}; p,q,r,l and x ∈ L = (-3,-2,-1,0,1,2,3} where in particular, a ∈ (-3, 3}, b ∈ {-2, 2}, c ∈ {-1, 1} and *l* = {0}.

Now expressing equation (3.25) or (3.26) with respect CP-gates, the following axioms and theorem suffices:

Axiom 3.12 **(About number of CP gates) If the integer** $h = 4i - 1$; $i \in \{1,2,3....\}$ **represents a number of logic values, then an** $h - valued$ **symmetrical r-nary signed-digit arbitrary MVL**

function f(x) of n variables and control variable x , where f(x), x ∈{ -(r-1),-(r-2), ……, -2,1,0,1,2,3…….(r-2),(r-1)}, can be completely expressed and constructed using a combination of $r^n - 1$ negative threshold literals and r^n positive threshold literal r-nary CP gates.

Theorem 3.1 **If** $F_1 = \{f_i(x^-)\}$ **and** $F_2 = \{f_j(x^+)\}$ **are** $\frac{h+1}{2} - valued$ **negative and positive signed threshold literal** $r = \frac{h+1}{2} - nary$ **CP gate derived T-gate networks respectively, then their parallel interconnection** F **, terminating with a common zero literal CP gate is an** $h - valued$ **symmetrical** $r = \frac{h+1}{2} - nary$ **signed-digit CP gate derived T-gate.**

$$F = F_1 + F_2 = CP_0(\{f_i(x^-)\}, \{f_j(x^+)\}, x) \qquad \textbf{(3.27)}$$

Corollary **If $F_1 = f(-x)$ and $F_2 = f(x)$ are two dissimilar mono polar CP gates expressible MVL functions, then the concatenation of F_1 and F_2 is a CP gate derived T gate expressible MVL function, F(x).**

Theorem 3.2 **If $f_{-3}(x)$, $f_{-2}(x)$, $f_{-1}(x)$, $f_0(x)$, $f_1(x)$, $f_2(x)$ and $f_3(x)$ are some non-symmetrically signed literal MVL variable functions then a 7- value symmetrical Quaternary signed digit CP gate derived T – gate otherwise RR7SQSD T-gate can be constructed using the expression**

$CP_0(CP_{-2}(CP_{-3}(f_{-3}(x), f_{-2}(x); x), CP_{-1}(f_{-1}(x), f_0(x); x), CP_1(CP_0(f_0(x), f_1(x); x),$

$CP_2(f_2(x), f_3(x); x); x); x)$

= $CP_0(CP_{-2}(CP_{-3}(-3,-2; f(x)), CP_{-1}(-1,0; f(x)); f(x)), CP_1(CP_0(0,1; f(x)), CP_2(2,3; f(x)); f(x)); f(x))$

$$= CP_0CP_{-2}\,CP_{-3}\,(-3,-2;x),CP_{-1}\,(-1,0;x);x),\ CP_1(CP_0\,(0,1;x),CP_2\,(2,3{:}x);x);x) \qquad (3.28)$$

As an example, figure 3.10 shows the CP-gate layout of an 11-valued symmetrical hex-nary signed digit T-gate.

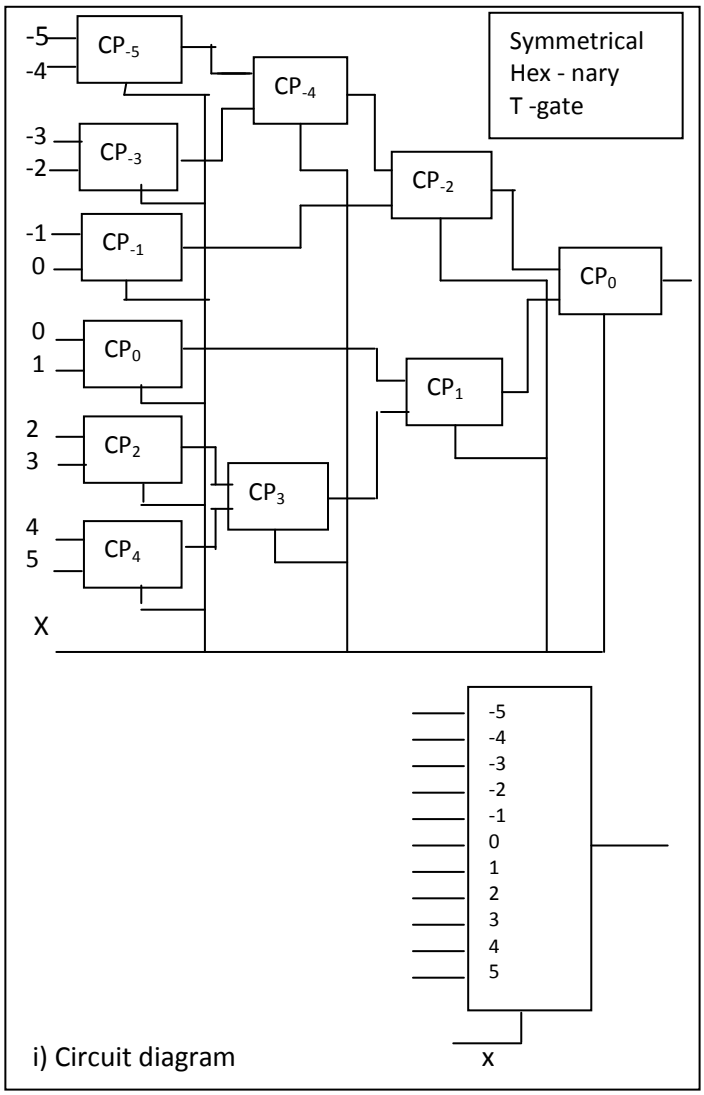


Figure 3.10: Layout of an 11-valued symmetrical hex-nary signed-digit CP-gate derived T-gate.

3.2 Method

This section explains the methods used to implement the various mathematical models formulated in section 3.1, to achieve the set objectives of this study. These methods come in form: algorithms, simulations and experiments. The simulations conducted with specific goals and objectives were performed on Pentium IV and Intel Core 2 Duo processors using Quick Basic version 4.5 (QB 4.5) source code while the experiments were performed on electronic work bench Multisim. Hence in section 3.2.1 the method used to prove the ability of RR7SQSNS to support addition/multiplication operation of integers of word lengths that are of ECC interest is presented. Section 3.2.2 presents the step by step account of how the theory of section 3.1.2 is used for the design of the hardware-implementable point multiplication on EC defined over RR7SQSD finite fields' algorithm.

3.2.1 Procedure for computing big integer multiplication

This section presents the proof that RR7SQSDNS supports fast arithmetic computations that are of very practical application significance. Some of these arithmetic operations are modular exponentiation $G = P^k \bmod h$ **and computing the product of big integers in a word length compute constrained environments without truncation. Both are critical arithmetic operations in secure communication and critical service delivery systems. The use of addition chains and thus, the prospect of using the RR7SQSD multiply-by-7 addition/subtraction chain for the execution of these operations have been presented in section 3.1.1.7. Consequently, this section in addition develops the relevant algorithms.**

3.2.1.1 Computing modular exponentiation $G = P^k \bmod h$

To compute modular exponentiation, the RR7SQSD form $f_\lambda = \{e_{k-1}, e_{k-2}, ..e_i ...e_1, e_0\}$**, of the exponent** k**, is obtained. Next the value of** P **is raised to the power of the most significant RR7SQSD of the exponent** e_{k-1} **that is** $P^{e_{\lambda-1}}$**. This is a 'jump starting' value. The remaining RR7SQSD** e_i **are scanned one at a time, the jump-start value/previous partial exponentiation result are raised to power 7 i.e.** $\left(P^{e_{\lambda-1}}\right)^7$ **and multiplied by the scanned RR7SQSD multiple of the jump-start value** $\left(P^{e_{\lambda-1}}\right)^{e_i}$ **, to obtain a current partial modular exponentiation** $Q = \left(P^{e_{\lambda-1}}\right)^7 * \left(P^{e_{\lambda-1}}\right)^{e_i}$**. Exponentiation is at the count of every RR7SQSD that is** $\{-3,-2,-1,0,1,2,3\}$ **but the '0' RR7SQSD scan does not attract a multiplication by** $\left(P^{e_{\lambda-1}}\right)^{e_i}$**. Algorithm code named AM1A in figure 3.11 describes this approach. A problem countered here is overflow of the system data word length as a result of the partial exponentiation's fast divergence due to the raising to powers 7 at every count of an RR7SQSD. This problem was contained by representing 7 in binary (**111_2**) so that** H^7 **decomposes to a three consecutive self multiplications followed by a reduction with the last iteration step result and then updating with** $\left(P^{e_{\lambda-1}}\right)^{e_i}$**. The self multiplication operations are respectively doubling and adding in non-modular arithmetic. Doubling is relatively cheaper than adding hence efficiency was further improved by representing '7' in the Near Adjacent Form (NAF)** $00\bar{1}$**. This way the triple double-and-add becomes three consecutive self multiplications.**

C: Algorithm **AM1A**
Input M, k, h, r
Output: $G = M^k \bmod h$
Obtain RR7SQSD form e_k of k
1. $H = H_0 = M^{e_{\lambda-1}} \in \{M, M^2, M^3\}$
; if $e_{\lambda-1} \in \{1,2,3\}$
H = 0 ; otherwise
$Q = H$
2. for i from $\lambda - 1$ down to 0 step –1do
$Q = Q^7 \bmod h$:
$H = H_0^{e_1} \bmod h$
3. $T = Q * H \bmod h$
$Q = T$
4. $G = Q$
Output G
End

Example 3.4 : To compute 5^{29}.
Output: G= $5^{29} = 18626451492309570 3125$
Solution: M = 5; k = 29
$\{e_k\} \equiv (1,\bar{3},1)_{RR7SQSD}$: when I =1: $e_{start} = 1$
$M^{start} = 5^1 = 5$:QL = 5 :H = $M^{start} = 5^1 = 5$
And when $i = 2; e_i = \bar{3}$: y=e_i =-3 :Q = H = QL=5
For j =2: *I* = 0:Q = Q*Q=25
j=1: *I* =0:Q=25.25=625
J=0: *I* =-1:Q=(625).(625)=390625
Q=Q*QL^{-I}:=390625/5=78125
Q=Q*H^y=78125*5^{-3}=625: QL = Q = 625
I = 3:y=e_3=1
J = 2:*I* = 0:Q =Q*Q = 625*625=39062
J = 1:*I* = 0:q =Q*Q = 390625*390625
J = 0:*I* =-1:
Q=Q*Q=232830643653869628906 25
Q=Q*QL^{-1} =Q/625 = 37252902984619140625
Q=Q*H^I = Q*5 = 18626451492309570 3125
$G = 18626451492309570 3125$

(a) RR7SQSD exponentiation algorithm. b) Computation process sketch

Figure 3.11: RR7SQSD modular exponentiation

The numerical example 3.4 shown in figure 3.11b further elucidates the computation steps. For operands of very large word length even with NAF representation speed of operation execution can still be of great concern hence speed up strategies are very necessary. Figure 3.12 shows some speed up strategies. The speed up Algorithm AM1A was also further improved upon when the RR7SQSD form of K is partitioned into groups of τ RR7SQSD for multiple RR7SQSD scanning. The procedure is as described in algorithm AM2.

C: Algorithm **AM1B**
Input M, k, h
Output: $G = M^k \bmod h$
Obtain RR7SQSD form ff_λ of K
Partition ff_λ into m groups of $\tau(=3)$
RR7SQSD i.e. $(\alpha_{mi}, \beta_{mi}, \gamma_{mi})$
Obtain the decimal equivalent ϕ,
of the first partition
$(\alpha_{m\lambda-1}, \beta_{m\lambda-1}, \gamma_{m\lambda-1})$
1. $H = M^{(\alpha_{mk-1}, \beta_{mk-1}, \gamma_{mk-1})}$; if $\phi \neq 0$:
else $H = M^0$
2. for i from λ-2 downto 0 step –1 do
$Q = H^\phi \equiv \left\{M^{(\alpha_i, \beta_i, \gamma_i)}\right\}^\phi$
2b. $Q = Q * M^{(\alpha i, \beta i, \gamma i)} = \left\{M^{(\alpha_i, \beta_i, \gamma_i)}\right\}^\phi * (\alpha i, \beta i, \gamma i)$
2c. $H = Q$
3. G = Q:output G
4. End

C: Algorithm **AM2**
Input: K,P
Output: $W = kP$
1. Obtain RR7SQSD equivalent $f_{k_{RR7SQSD}}$, of k
1a. Partition $f_{k_{RR7SQSD}}$ to m groups of τ_s RR7SQSD
2. *For Tt from 1 to m do:* $\psi_{Tt} = \alpha_{mTt}\beta_{mTt}\lambda_{mTt}$
2a. Obtain decimal equivalent of ψ_{Tt}
3. $H_0 = \psi_1 : Q_0 = \psi_1$
3a. for $z = 2$ to *m do :* $H_{trim} = \psi_z$
4. *For d from* $\tau_s - 1$ *downto 0 step -1 do :*
$Q = Q_0$
4a. for l *from 1 to 3 do:* $Q = Q + Q$
4b. $Q_0 = Q - Q_0$
4c. $H = Q_0 + H_{trim} : Q_0 = H$
5. $W = H$ *:output W*
6. *End*

Fig. 3.12 Speed up strategies for RR7SQSD modular exponentiation

Numerical example 3.5 below summarizes procedure for using RR7SQSD for modular exponentiation computation.

***Numeric Example 3.5*: To compute:** $M^{6984572} (\bmod h)$ **Using: i) Single RR7SQSD scan ii) Multiple RR7SQSDs scan and iii) Hardware implement-friendly multiple RR7SQSDs scan with NAF coding.**

Solution: **For the purpose of clarity, no numerical value shall be assigned to M.** $k = 6984572_{10} \equiv f_\lambda = \left\{113, 3\bar{3}0, 120\right\}_{RR7SQSD}$**. This is partitioned into groups of three RR7SQSD** $(\alpha, \beta, \gamma) = (113, 3\bar{3}0, 120) \equiv (59, 126, 63)$**. The first partition being**

$e_{\lambda-1}{}^{l} = 113 \Rightarrow H = M^{59}$**. The numerical example 3.5 in figure 3.13 sketches the computation procedures for each case. As it can be seen that where as a one RR7SQSD at a time scan would yield an addition/subtraction chain of *1P, 8P, 59P, 416P, 2909P, 20363P, 14252P, 997796P, 6984572P* . However, using algorithm AM2 a 3-RR7SQSD at a time scan example (2iii) yields the very short addition/subtraction chain of *59P, 20363P, 6984572P* which drastically reduces the number of operations.**

i) Direct single RRSQSD scan (Algorithm AM1) $e_{i-1}=1 \Rightarrow H=M^1$

H	Q	H
$M^2 \rightarrow M^4 \rightarrow M^8$	$M^8 M^{-1}$	$M^7 M^1 = M^8$
$M^{16} \rightarrow M^{32} \rightarrow M^{64}$	$M^{64} M^{-8} \rightarrow M^{56}$	$M^{56} M^3 = M^{59}$
$M^{118} \rightarrow M^{236} \rightarrow M^{472}$	$M^{472} M^{-59} \rightarrow M^{413}$	$M^{413} M^3 = M^{416}$
$M^{832} \rightarrow M^{1664} \rightarrow M^{3328}$	$M^{3328} M^{-416} \rightarrow M^{2912}$	$M^{2912} M^{-3} = M^{2909}$
$M^{5818} \rightarrow M^{11636} \rightarrow M^{23272}$	$M^{23272} M^{-2909} \rightarrow M^{20363}$	$M^{20363} M^0 = M^{20363}$
$M^{40726} \rightarrow M^{81452} \rightarrow M^{162904}$	$M^{162904} M^{-20363} \rightarrow M^{142541}$	$M^{142541} M^1 = M^{142542}$
$M^{285084} \rightarrow M^{570158} \rightarrow M^{1140336}$	$M^{1140336} M^{-142542} \rightarrow M^{997794}$	$M^{997794} M^2 = M^{997796}$
$M^{1995512} \rightarrow M^{3991184} \rightarrow M^{7982363}$	$M^{7982363} M^{-997796} \rightarrow M^{6984572}$	$M^{413} M^0 = M^{6984572}$

ii) Multiple RRSQSD scan Modular exponentiation (Algorithm AM1B)

m	eem	H	Q	H
1	$3\bar{3}0$	$\left(M^{59}\right)^{343} \rightarrow M^{20237}$	$M^{20237} M^{126} \rightarrow M$	M^{20363}
0	120	$\left(M^{20363}\right)^{343} \rightarrow M^{6984509}$	$M^{6984509} M^{63} \rightarrow M$	$M^{6984572}$

iii) Implement friendly multiple RR7SQSD scan with NAF coding (AM2)

$\alpha_{m-1}\beta_{m-1}\gamma_{m-1}=113 \Rightarrow Q=H=59 \equiv 7\,[(1*7)+1]+3 \equiv 2[3[D]+2[A]]=6[D]+4[A]$

m	$\alpha_m\beta_m\gamma_m$	H	Q	j	l_j	Q	Q	H
1	$3\bar{3}0$	59	59	2	0	$59 \rightarrow 118$	472	
					0	$118 \rightarrow 236$	- 59	
					$\bar{1}$	$236 \rightarrow 472$	413	
				1	0	826	3304	
					0	1652	**- 413**	
					$\bar{1}$	3304	2891	
				0	0	5782	23128	20237P
					0	11564	- 2891	+ 126P
					$\bar{1}$	23128	20237	20363
0	120	20363	20363	2	0	40726	162904	
					0	81452	- **20363**	
					$\bar{1}$	162904	142541	
				1	0	285082	1140328	
					0	570164	-***142541***	
					$\bar{1}$	1140328	997787	
				0	0	1995574	7982296	6984509P
					0	3991148	-**997787**	+ 63P
					$\bar{1}$	7982296	6984509	6,984572P

Figure 3.13 Solution to numerical Example 3.5 on speed up strategies.

3.2.1.2 Experiment on RR7SQSD non-finite field application:- Big integer multiplication

The principle of RR7SQSD addition/subtraction chain as demonstrated in algorithm AM2 is extended to find the product of two very big integer operands in a word length compute constrained environment. The results of this experiment will prove that the RR7SQSD multiply-by-7 addition/ subtraction chain scheme is a powerful tool for computing the product of integer operands word lengths of cryptographic significance. Operands of length 15 decimal character digits (47 bits) and up to those of length 160 decimals character digits (501 bits) as permitted by the simulating software, which would ordinarily attract scientific presentation or rounding-off in Pentium III and IV were used as specimen test values for *k* and *P* data.

In performing the experiment as explained in section 3.2.1.1, these scary and weary decimal digit strings of the multiplicand p were partitioned into *m* groups of word length υ where υ, is a character digit less than the computing system data word length. A Buick basic version 4.5 source code programme used to simulate the process of multiplication is as presented in Appendix CH 3.3. The program automatically transforms the full decimal digits length of the multiplier k into the RR7SQSD form for the sequential scanning. The multiply-by-7 procedure (algorithm AM2) is then executed on each partition of the multiplicand P. This is followed immediately, by the addition or subtraction operation of the scanned RR7SQSD digit multiple of the multiplicand

$e_i P$, to obtain a partial product. The process then is repeated till all RR7SQSD have been scanned and treated accordingly. The raw results of this experiment are as shown in Appendix CH4.1. A summary of these results are presented and discussed in section 4.1.

3.2.2 Methodology on the design and testing of the hardware friendly algorithm for arithmetic operations on EC defined over $SGF(7^m + \psi_i)$

This section presents the methods for implementing of the various theoretical formulation proposed in section 3.1.2.

3.2.2.1 Procedure for constructing RR7SQSD finite fields

The goal of here is to show that RR7SQSD element finite fields and in particular the type of $SGF(7^m)$ can be constructed using any of the existing methods in literature. For the purpose of demonstrating this assertion in this study, the method discussed in section 2.3.1.5 is followed.

To construct $SGF(7^m)$ is to find monic irreducible RR7SQSD polynomials $q(x) \in SGF(7^m)$ of degree $m \geq 1$ with coefficients in SGF(7^m). A list of all the SD monic Polynomials of degree *m* in *SGF(7^m)[x]* with constant terms was made and for each one of the signed digit polynomials listed, substitute the value of $x \in \{\bar{3},\bar{2},\bar{1},0,1,2,3\}$. The reducible Polynomials becomes zero, otherwise the Polynomial is irreducible. Thus, using this brute force method *21, 112 and 57580* RR7SQSD monic irreducible polynomials corresponding for *degrees 2, 3*

and *7 were found.* The result of constructing the 21 irreducible polynomials and thus the elements of $SGF(7^2)$ is presented and discussed shown in section 4.2.1.

3.2.2.2 Procedure for computing multiplicative elements in RR7SQSD finite field of type $SGF(7^m)$

It has been explained in section 3.1.2.3 the algorithm presented in [60] can be used to compute RR7SQSD finite field's inverse elements. It is also observed that the Extended Euclidean Algorithm (EEA) based polynomial multiply method in [22,54,58] is VLSI implement friendly and the nature of RR7SQSD finite fields finds it more appropriate.

Given an element $b(x) = b_{m-1}(x^{m-1}) + b_{n-2}(x^{m-2}) + ... + b_1(x) + b_o$ and the irreducible prime modulus $w(x) = x^{n-1} + w_{n-2}(x^{n-2}) + w_{n-3}(x^{n-3}) + ... + w_1(x) + w_0$ such that $b(x), w(x) \in SGF(7^k \pm \psi)$, $b_l, w_t \in \{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$ and $n > m$. To find $b^{-1}(x)$, one need to multiply the highest coefficients of b(x) b_{m-1} with $\chi_i(x) \in \{-3, -2, -1, 0, 1, 2, 3\} x^{n-m}$ and check for simultaneous coalescences with the highest coefficient of w_{n-1} and the degree or power of the modulo w(x) .This is subtracted from highest degree of w(x) . Record of all coalescence values of $\chi_i(x)$ are kept while at each iteration step, $w_i(x)$ becomes $b_{i-1}(x)$ and $b_i(x)$ becoming $R_{i-1}(x)$, the process of computing the Greatest Common Divisor (GCD) continues while the remainder $R(x) \neq +1$ with $p = i_{\max}$ .At end, starting with four initializing variables say $D_0(x) = 1, H_0(x), D_1(x) = 0, H_1(x) = 1$ the row elements of a $2 - by - p$ dimensional matrix: $\{\{D_0(x) - \chi_k(x).D_1(x)\}.w(x), b(x).\{H_0(x) - \chi_k(x).H_1(x)\}\}$ are computed. For $k = p$ down to 1 step -1. The multiplicative inverse

$b^{-1}(x) = \{.\{H_0(x) - \chi_k(x).H_1(x)\}\}$ **provided** $v_k(x) = \{\{D_0(x) - \chi_k(x).D_1(x)\}.w(x) \pm b(x).\{H_0(x) - \chi_k(x).H_1(x)\}\} = +1$**. On condition of** $v_k(x) = 0$**, b(x) and w(x) are not co prime. In all other cases, it simply implies the existence of an error in calculation. The method is exemplified in figure 3.14 with the numerical example 3.6 directly, in RR7SQSDNS domain and a simulation experiment that source input operands from the sequence** $H \in \{37,79,241,17011,117721,.....\}$**.**

i) Direct RR7SQSD domain multiplicative inverse computation

Numerical example 3.6 **Compute** $(-2x-2)^{-1} \bmod(x^2 - 3x - 2)$ **which are field element and irreducible polynomial in** $SGF(7^2)$ **respectively.**

Solution; **i) Classical Euclidean Algorithm ii) the EEA**

$$
\begin{array}{r}
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad 2x-2 \\
-2x-2\overline{)\,x^2-3x-2} \\
\quad\quad x^2-2x-3 \\
\hline
\quad -x+1
\end{array}
$$

$$
\begin{array}{r}
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad 2 \\
-x+1\overline{)\,-2x-2} \\
\quad\quad -2x+2 \\
\hline
\quad\quad 3
\end{array}
$$

$$
\begin{array}{r}
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad 2x \\
3\overline{)\,-x+1} \\
\quad\quad -x \\
\hline
\quad\quad 1
\end{array}
$$

1	0
0	$1(3x-2)$
1	$-3x+2(2)$
-2	$-[-(-3x+2)2]+1$
1	$-[x-3]+1$
-2	$-x-3(2x)$
$-3x+1$	$-[2x(-x-3)]-3x+2$
	$-[-2x^2+x]-3x+2$
	$2x^2-x-3x+2$
	$2x^2+3x+2$

iii) Checking:

$$(-3x+1).(x^2-3x-2) = -3x^3+3x^2+3x-2$$
$$(-2x-2).(2x^2+3x+2 = 3x^3-3x^2-3x+3)$$
$$R(x) = +1$$

Hence, $(-2x-2)^{-1} \bmod x^2 - 3x - 2 = 2x^2 + 3x + 2$**.**

Figure 3.14: Solution to numerical example on computing inverse in $SGF(P^k)$**.**

The EEA method is used to compute all the inverses of all the elements in $SGF(7^2)$ using the irreducible polynomial $x^2 - 3x - 2$. The complete set of multiplicative inverse of the elements of $SGF(7^2)$ thus obtained is presented and discussed section 4.2.2.1 figure 4.3.

ii) Experiment on field element multiplicative inverse computation in the indirect RR7SQSD domain

The EEA method is also used to compute the multiplicative inverses of a few special primes presented in appendix CH3.2. In this particular case the computation process is performed in decimal digit domain but with RR7SQSD finite field representable moduli. The purpose of this experiment was to ascertain the accuracy of EEA on RR7SQSD finite field multiplicative inverse computation even with certain special primes from the number sequence $H \in \{7^m + \alpha\beta i\}$ that is, $H \in \{37,79,241,17011,117721,.....\}$ as discussed in section 3.2.1.1 axiom 3.9. The QB 4.5 source code program for the experiment is presented in appendix programs CH3.5 and the output of the experiment is presented and discussed in section 4.2.2.2. This experiment is a necessary integral module in all future computations in this study.

3.2.2.3 Design and testing of a hardware-implementable EC point multiplication algorithm

In this study the algorithm in view is based on the multiply-by-7 addition subtraction chain thus, it is an extension of the algorithm AM2 enunciated section 3.2.1.1. Thus some necessary components of the algorithm are a unified EC point's addition and

point doubling module, a jump-start module , a $7P$ realization module and an updating module. The process of computing the x, y components were made autonomous in section 3.1.2.5. Therefore the design of the algorithm starts with the derived formulae integrity testing. The section ends with the methods for testing the algorithm.

3.2.2.3.1 Equation of an EC defined over RR7SQSD finite fields

This section introduces the nature of elliptic curves defined over RR7SQSD element finite fields. Precisely, elliptic curves defined over RR7SQSD finite field in this study, are those elliptic curves defined over $SGF\left(7^m+\psi\right)$ hence, it is therefore sufficient to describe these curves by the same normal prime Galois field elliptic curve equation presented in equation (3.19).

$y^2 = x^3 + ax + b$, $4a^3 + 27b^2 \neq 0$ **with** $a, b, x, y \in SGF(7^m + \psi)$ **(3.29)**

Where: $m = 1,2,3,.....$, $\psi = i\alpha\beta$; $i = 0,1,2,....$; $\alpha + \beta = 7$ **and** $\alpha\beta < 2P$

3.2.2.3.2 Autonomous EC points' addition and point doubling X and Y component equations and their accuracy

It was stated in section 1 and subsequently in the opening paragraph of section 2.4.2 that EC arithmetic operations comprises EC point's addition, EC point doubling and EC point multiplication. The elliptic curve point addition and point doubling equations described in table 2.4 sections 2.4.2.1 can also be used to compute points' addition and point doubling on EC defined over RR7SQSD fields. These equations are very efficient. However, it can be seen that y_3 computes if and only if x_3 is known. In this study using the concept of slope-intercept and co-linearity of three points lying on a straight

line determinant equations in [101], the x and y components' computations (x_w, y_w), are made autonomous in order to make the implementation parallel architecture friendly. The modified formulae are as presented in equation (3.30).

a) Point addition; $U \neq V$ **and let** $\nabla x = x_v + x_u$ **;** $\Delta x = x_v - x_u$ **;** $\Delta y = y_v - y_u$ **and**

$\Delta_{xy} = x_v y_u - x_u y_v$

$$x_w = \frac{\Delta^2 y - \Delta^2 x \nabla x}{\Delta^2 x} \; ; \; y_w = \frac{\Delta^2 x\left[\Delta y \nabla x - \Delta_{xy}\right] - \Delta^3 y}{\Delta^3 x}$$

b) Point doubling

$$x_w = \frac{\left(\left(3x_u{}^2 + a\right)^2 - 8x_u y_u{}^2\right)}{\left(2y_u\right)^2} \; ; \quad y_w = \frac{\left(3x_u{}^2 + a\right)\left[12x_u y_u{}^2 - \left(3x_u{}^2 + a\right)^2\right] - 8y_u{}^4}{\left(2y_u\right)^3} \qquad \textbf{(3.30)}$$

3.2.2.3.3 Projective versus Affine coordinate system

In order to establish the appropriate coordinate system (affine or projective), the order of complexity for computing equation (3.30) was calculated. Table 3.3 shows the summary of the computation.

Table 3.3 EC point addition/doubling complexity in the projective and affine

Coordinate	Projective					Affine				
Operation type	A	M	S	I	Total	A	M	S	I	Total
Addition	7	18	5	1	31	7	10	3	1	21
Doubling	4	12	5	1	22	4	19	5	1	20
$\Delta\left(\sum A - \sum D\right)$					9					1

It can be seen that: i) While in the projective coordinate system, the number of addition operations >> the number of doubling operations in the affine coordinates both addition and doubling operations approximately have equal number of operations. ii) The difference in the total number of operations i.e.; $\Delta\left(\sum A - \sum D\right)$ **in projective is by far greater than** $\Delta\left(\sum A - \sum D\right)$ **in affine coordinate. iii) In either projective or affine coordinate system the number of addition operation is always less than the number of multiplication operations. On this basis all future calculations in this study were carry out on the affine coordinate system.**

3.2.2.3.4 Integrity of points' addition and point doubling formulae

The SMVL ECC arithmetic unit VLSI circuit being proposed in this study is expected to function directly on threshold voltage levels in the range $-3,-2,-1,0,1,2,3$ **and the arithmetic of** $SGF\left(\left(7^m + \psi_l\right)^n\right)$ **must be capable of being performed in a ground field of type** $SGF(7)$ **or lower extension field of type** $SGF\left(7^l\right), l < m.n$ **as well as in the decimal domain. It is necessary to affirm that there is no degradation of the computed results in both domains. Consequently, two numerical examples were used to provide the confirmation.**

i) **the corresponding formulae in table 2.4 are compared with the modified in 3.3.1.1 using numerical example 3.7 and**
ii) **An embedded arithmetic operation on EC defined over** $SGF\left(7^m + \psi_i\right)$ **will be executed using numerical example 3.8.**

Numerical example 3.7: **Let the elliptic curve E, be defined as the solution of** $y^2 = x^3 + x + 1$ **over the Galois field GF (23). The group E has the following 28 points**

including O [68].

$(0,1),(0,22),(1,7),(1,16),(3,10),(3,13),(4,0),(5,4),(5,19),(6,4),(9,19),(7,11),(7,12),(9,7),(9,16),(11,3),(11,20)$

, $(12,4),(12,19),(13,7),(13,16),(17,3),(17,20),(18,3),(18,20),(19,5),(19,18)$

It is required to perform EC point's addition of the two points: $(3,10),(9,7)$ **and EC point doubling of the point** $(3,10)$ **using: i) the formulae of table 2.4 and ii) the present effort as described in equation (3.20). The computation is as shown in figure 3.13 and the results are the same in all the two options.**

Addition: $\underline{(3,10)+(9,7)=(17,20)}$

Using table 2.2

$$m=(7-10)(9-3)^{-1}=(-3)(6)^{-1}=11$$

$$x_w=11^2-3-9\equiv 17$$

$$y_w=11(3-17)-10=20$$

using the present effort, equation (3.30)

$$x_w=\frac{9-36(12)}{36 \bmod 23}=\frac{9-18}{13}=-9\left(13^{-1}\right)$$
$$=-9(16)=-6\equiv 17$$

$$y_w=\frac{6^2[(-3.12)-[9.10-3.7]]-(-3)^3}{6^3}$$
$$=-4\left(9^{-1}\right)=-4(18)=-3\equiv 20$$

Doubling: $\underline{(3,10)+(3.10)=(7,12)}$

Using Table 2.2

$$m=\left(3\left(3^2\right)+1\right)(20)^{-1}=6$$

$$x_w=6^2-6=7$$

$$y_w=6(3-7)-10=12$$

using the present effort, equation (3.30)

$$x_w=\frac{(3(3)^2+1)^2-8(3)\left(10^2\right)}{4(10)^2}=-16=7$$

$$y_w=\frac{\left(3.3^2+1\right)\left|12.3.10^2-\left|\left(3.3^2+1\right)^2\right|-8.10^4\right|}{(2.20)^3}==-11\equiv 12$$

Figure 3.15: Comparative solutions to numerical example on EC arithmetic formulae

3.2.2.3.5 A unified EC points addition and point doubling operations computation scheme

For efficient arithmetic and parallel hardware-implement-friendly architecture equation (3.21) was further transformed into the system of equations (3.31a) to (3.31f) so that a point's addition or point doubling operation can be executed in a single module:

$$\left.\begin{aligned} x_W &= GC^{-2} \\ y_W &= HC^{-3} \end{aligned}\right\} \quad \textbf{(3.31)}$$

$$\Delta x = x_v - x_u ; \Delta y = y_v - y_u ; \nabla x = x_v + x_u$$

$$C = \begin{cases} \Delta x & \text{; If addition} \\ 2y_u & \text{; If doubling} \end{cases} \quad \textbf{(3.31a)}$$

$$B = \begin{cases} \Delta^2 x \nabla x & \text{; If addition} \\ x_U y_u & \text{; if doubling} \end{cases} \quad \textbf{(3.31b)}$$

$$A = \begin{cases} \Delta y & \text{; if addition} \\ 3x_U{}^2 + a & \text{; if doubling} \end{cases} \quad \textbf{(3.31c)}$$

$$F = \begin{cases} y_U x_V - y_V x_U & \textbf{if addition} \\ 1 & \textbf{; if doubling} \end{cases} \quad \textbf{(3.31d)}$$

$$G = \begin{cases} A^2 - C^2 \nabla x & \textbf{; if addition} \\ A^2 - 8B & \textbf{; if doubling} \end{cases} \qquad \textbf{(3.31e)}$$

$$H = \begin{cases} C^2\left[A\nabla x - F\right] - A^3 & \textbf{if addition} \\ A\left[12B - A^2\right] - \dfrac{C^4}{2} & \textbf{; if doubling} \end{cases} \qquad \textbf{(3.31f)}$$

The corresponding computation flow diagram is as shown in figure 3.16.

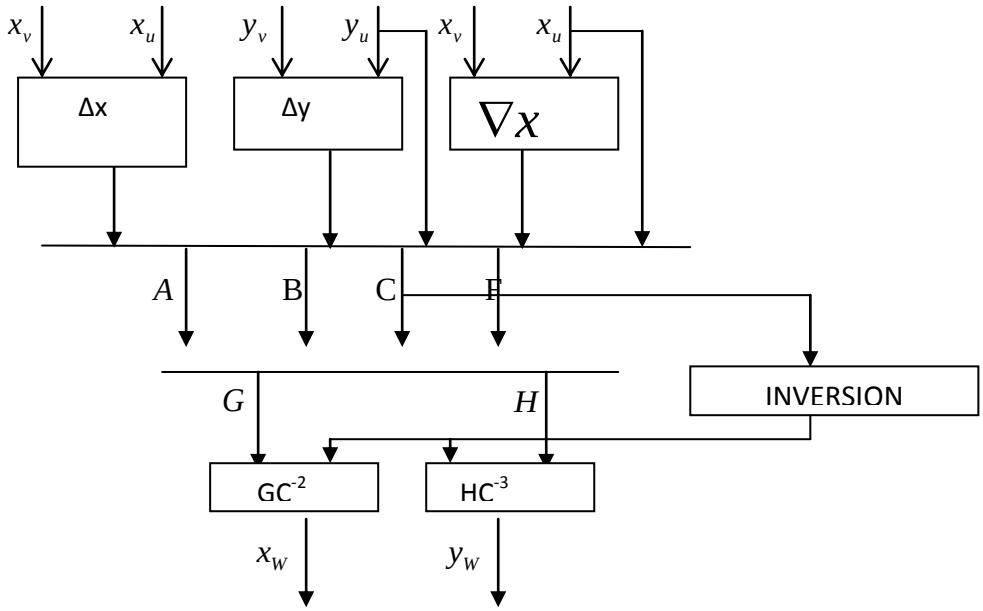


Figure 3.16: Computation flow diagram of the unified EC addition/doubling scheme

Variables are batched and elements of a batch are computed in parallel. For example, the elements of the set variables $\{A, B, C, F\}$ are computed in parallel just as $\{G, H\}$ are computed in parallels. This way the order of complexity in any batch equals only that of the element with the highest complexity order. Inversion is done only once in the course of the

entire calculations. The x, y coordinates of a batch element is computed simultaneous as shown in figure 12. Consequently, the order of complexity was calculated for the transformed equation (3.20) as shown in table 3.4. The transformation brought slight improvement in the order of complexity as compared with the previous order of complexity in table 3.3. A total of 22 operations for addition and just 17 doubling operations, as against 21 additions and 20 doubling are required. Table 3.5 shows a comparative analysis of the results. It includes data from similar computation in [22] for comparison. The first two rows show the total number of the arithmetic operations in binary and prime and the third row is that of the RR7SQSD finite fields. There is slight reduction in the number of addition operations. This shall be part of the considerations when designing the VLSI circuit later in this study.

Table 3.4 Transformed equation order of complexity

Operation	Addition operation				Doubling Operation			
Variables	A	M	S	I	A	M	S	I
A	1	-	-	-	-	-	1	-
B	1	1	1	-	-	-	-	-
C	-	-	-	-	-	-	-	-
F	1	2	-	-	-	-	-	-
G	1	1	2	-	1	-	1	-
H	2	3	1	-	2	1	3	-
C^{-1}	-	-	-	1	-	-	-	1
C^{-2}	-	1	-	-	-	1	-	-
C^{-3}	-	1	-	-	-	1	-	-
X	-	1	-	1	-	1	-	-
Y	-	2	-	-	-	2	-	-
Total	**6**	**11**	**4**	**1**	**4**	**6**	**6**	**1**
	$\sum = 22$				$\sum = 17$			

Table 3.5 Implementation factors' comparative operation cost summary

Field types	Point addition	*Point Doubling*
Binary fields [74]	9 addition, 2 multiplication, 1 squaring, 1 inversion	4 addition, 2 multiplication, 2 squaring, 1 inversion
Prime fields	5 addition, 2 multiplication,1 squaring, 1 inversion	4 addition, 5 multiplication, 2 squaring, 1 inversion
RR7SQSD fields	6 addition, 11 multiplication, 4 squaring, 1 inversion	4 additions, 6 multiplications, 6 squaring, 1 inversion.

Numerical example 3.8 **Given the elliptic curve** $y^2 = x^3 - 5x + 8$ **defined over** $GF(37)$**, and its' scattered 45 points (not shown) on the** x, y **plane:**

$(1,2),(1,35),(5,21),(5,16),(6,3),(6,34),(8,6),(8,31),(9,27),(9,10),(10,25),(10,12),(11,27),(11,10),$

$(12,23),(12,14),(16,19),(16,18),(17,27),(17,10),(19,1),(19,36),(20,8),(20,29),(21,5),(21,32),$

$(22,1),(22,36),(26,8),(26,29),(28,8),(28,29),(30,25),(30,12),(31,9),(31,28),(33,1),(33,36),$

$(34,25),(34,12),(35,26),(35,7),(36,7),(36,30),0.$

Compute EC:

i) $(1,2)+(11,2)$

ii) $(5,21)+(33,1)$

iii) $(17,27)+(3,19)$

in $SGF(1\bar{2}2)$

Solution: **The given elliptic curve in the normal prime number domain is defined over the finite field** $GF(37)$ **with its elements** $x \in \{0,1,2,3,4,5,6,.....36\}$ **which on the given** (x, y) **coordinate system is of type I as shown in figure 3.15. To execute the embedded arithmetic that is, to perform arithmetic operations on EC defined over the Restricted Radix-37 Symmetrical** $18 = \frac{37-1}{2}$ **-nary Signed Digit (** $RR37S18SD$ **) element finite field EC, the** x, y **plane is transformed into a** x', y' **plane as shown in fig. 3.15. In this new plane, the range of possible values now ranges between** $\bar{3}3,\bar{2}\bar{3},\bar{2}\bar{2},.....\bar{3},\bar{2},\bar{1},0,1,2,3,.....,22,23,3\bar{3}$ **that is** $-18,-17-,-16,....,-1,0,1,,,,18$ **in the** x', y' **plane and the elliptic curve defined over** $SGF\left(7^2 - 12 = 37 \equiv 1\bar{2}2\right)$ **became the trace of type II. The operand values and point coordinates are correspondingly coded in RR37S18SD domain that is, in the character digit set of** $\left\{\bar{3}3,\bar{2}\bar{3},\bar{2}\bar{2},.....\bar{3},\bar{2},\bar{1},0,1,2,3,.....,22,23,3\bar{3}\right\}$**. The transformation facilitates the execution of the EC point addition and doubling operations of** $SGF(37)$ **on RR7SQSD strings. In the now RR7SQSD finite field of** $SGF(1\bar{2}2)$**, the given elliptic curve equation** $y^2 = x^3 - 5x + 8$ **also becomes** $y^2 = x^3 - 1\bar{2}x + 11$**. Similarly, the given points transform into** $(01,02),(2\bar{3},\bar{1}\,\bar{3}),(1\bar{2},\bar{2}\bar{2}),(1\bar{3},01),(23,\bar{1}\,\bar{3}),(\bar{1}\,1,12)$ **.respectively.**

For convenience rewriting equation (3.30) in the form of equation (3.32)

$$\left.\begin{aligned} x &= \frac{\Delta^2 y - \Delta^2 x \nabla x}{\Delta^2 x} \\ y &= \frac{\Delta^3 y - \Delta^2 x[\Delta y \nabla x + \Delta x y]}{\Delta^3 x} \end{aligned}\right\} \qquad \textbf{(3.32)}$$

The detailed computations of the EC point's addition operations using equation (3.22) are as

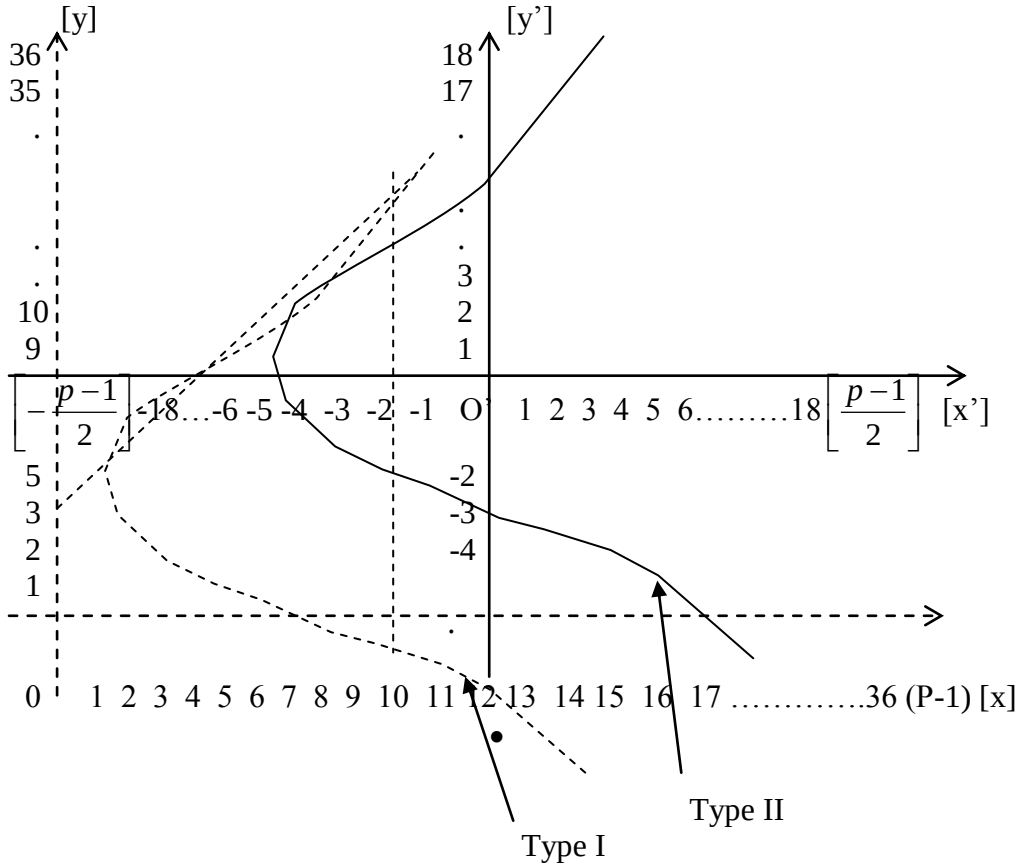


Figure 3.17: Embedding the RR37S18-narySD EC into a normal prime field elliptic curve.

shown in table 3.6. Although it was also noticed that the computation may need correction due to the carries ignored which is characteristic of finite field arithmetic operations. An ignored or thrown away carry is essentially an addition or subtraction of '4' hence a necessary and sufficient correction action is to iteratively, for a maximum of seven times, either subtract or add '4' from or to the computed value of x_c or y_c. However, all computed values are indeed points on the given elliptic curve thereby proving that equation (3.31) provides efficient arithmetic on elliptic curves defined over RR7SQSD finite fields.

Table 3.6 **Solutions to numerical example on** $SGF(7^2-12)$

Points	$x-$ var *iables*	y-variables	Δxy	x	y
$(1,2)+(11,27)\equiv$ $(01,02)+(2\bar{3},\bar{1}\,\bar{3})$	$\nabla x = 2\bar{2}$ $\Delta x = 23$ $\Delta^2 x = \bar{1}1$ $(\Delta^2 x)^{-1} = 1\,\bar{1}$ $(\Delta^3 x)^{-1} = \bar{3}\bar{3}\bar{2}$ $\Delta^2 x \nabla x = 02$	$\Delta y = \bar{1}2$ $\Delta^2 y = \bar{2}2$ $\Delta^3 y = \bar{2}0$ $\Delta y \nabla x = 20$	$\Delta xy = 11$	$\frac{2\bar{2}-\bar{1}1.2\bar{2}}{\bar{1}1}$ $x = \bar{2}\bar{3} \equiv 20_{10}$ $x_c \equiv x = 20_{10}$ $(x_w, y_w) = (28,8)$	$\frac{\bar{2}0-\bar{1}1.\bar{3}3}{\bar{1}1}$ $y = 1\bar{3} \equiv 04_{10}$ $y_c = y + 1\bar{3}$ $= 11 \equiv 8_{10}$
$(5,21)+(33,1)\equiv$ $(1\bar{2},\bar{2}\bar{2})+(1\bar{3},01)$	$\nabla x = 01$ $\Delta x = \bar{2}\bar{2}$ $\Delta^2 x = 123$ $(\Delta^2 x)^{-1} = 21\bar{3}$ $(\Delta^3 x)^{-1} = \bar{2}02$ $\Delta^2 x \nabla x = 23$	$\Delta y = 23$ $\Delta^2 y = \bar{1}1$ $\Delta^3 y = 20$ $\Delta y \nabla x = 23$	$\Delta xy = 11$	$\frac{\bar{1}1+\bar{2}\bar{3}}{23} = 22$ $\equiv 16_{10}$ $(x_w, y_w) = (16,19)$	$(20+\bar{1}0).\bar{2}02$ $= \bar{2}1 \equiv 24_{10}$ $y_c = y + 1\bar{3} + 1\bar{3}$ $+1\bar{3}+1\bar{3}+1\bar{3}+1\bar{3}$ $= \bar{3}3 \equiv 19_{10}$
$(17,27)+(31,9)\equiv$ $(23,\bar{1}\,\bar{3})+(\bar{1}1,12)$	$\nabla x = 1\bar{3}$ $\Delta x = \bar{3}\bar{2}$ $\Delta^2 x = 20$ $(\Delta^2 x)^{-1} = \bar{3}\bar{3}\bar{2}$ $(\Delta^3 x)^{-1} = 3\bar{1}\bar{2}$ $\Delta^2 x \nabla x = \bar{2}3$	$\Delta y = 2\bar{2}$ $\Delta^2 y = 03$ $\Delta^3 y = \bar{1}1$ $\Delta y \nabla x = 33$	$\Delta xy = 00$	$\frac{03-2\bar{3}}{20} =$ $20.\bar{3}\bar{3}\bar{2} =$ $01 \equiv 01_{10}$ $(x_w, y_w) = (01,35)$	$\frac{\bar{1}1-20.32}{\bar{2}\bar{2}}$ $= \bar{3}\bar{3}.3\bar{1}\bar{2}$ $= 0\bar{2} \equiv 35_{10}$

3.2.2.3.6 Optimal method for $7P$ **realization**

The conventional way to realize *7P or a* multiply by-7 operation is to perform 7

consecutive elliptic curve addition operations: i.e. $7P = P + P + P + P + P + P + P$**. This is time consuming and a better strategy is very necessary. Hence it is imperative to find an optimal method of realizing** $7\psi_{j-1}$ **if an RR7SQSD addition/subtraction chain is to be used to compute EC point multiplication. Table 3.7 shows some options of realizing** $7P$**; [A] for Addition operation and [D] for a Doubling operation). Criteria for choosing the optimal option shall be based on least number of addition operations and order of complexity of computation. For example in case I, initially a copy of point** p **is made and then the original copy of point** p **is doubled to obtain** $2p$**. This was added to the copy previously made to obtain** $3p$ **which was in turn doubled to obtain** $6p$**. Finally to obtain 7p, a copy of point** p **is added to** $6p$ **It can be observed that in all the cases the first step is to double** p**. In case VI, values of** $3p,4p$ **are directly computed before adding together. Bearing in mind that addition is a binary operation and in terms of computer instruction sequencing, two operands have to be fetched from main/auxiliary memory or CPU general purpose registers in order to execute this operation successfully it then follows that the fewer the number of addition operations the faster the system could be. From table 3.7, the following three variants out of the seven were of interest.**

i) **Case II; '7' is represented in NAF so that only one addition operation follows three relatively cheap doubling operations to give *7P = 8P - P*.**

ii) **Case V; *2P is computed first before computing 7P = 3P+4P*. This option shall be referred to as the indirect computation**

Table 3.7 Methods of realizing the multiply-by-7 operation

CASES	I		II		III		IV		V		VI		VII	
OPERATIONS	2P	D	2P	D	2P	D	2P	D	2P	*	*	*	*	*
	3P	A	4P	D	3P	A	4P	D	3P	*	3P	*	*	*
	6P	D	8P	D	5P	A	6P	A	4P	D	4P	*	*	*
	7P	A	7P	A	7P	A	7P	A	7P	A	A	A	7P	DR
TOTAL	2D+2A		3D+1A		1D+3A		2D+2A		1D+1A		0D+1A		0D+0A	

iii) Case VI; Direct *3P* ,*4P* (elliptic curve point Tripling [T] and Quadrupling [Q]) operations are before performing a single point addition operation of *7P* = *3P*+*4P* . This shall be referred to as the semi direct computation.

These cases separately require the availability of: {2P, 4P, 8P, 7P}, (2P,(3P,4P)), {3P,4P} multiples of P to implement *7P*.

3.2.2.3.7 Comparative cost analysis

The computational order of complexity was computed for cases II, V and VI as analyzed above. The summary of the computation is as presented in figure 3.18 where ST and GT denote subtotal and Grand Total respectively. It is observed that the cost of 7P formation is minimal with the indirect approach. It requires only a total of 137 operations. One needs to compute 2P (Doubling operation) straight from the given point's coordinates and use this to form 3P+4P=7P. This option formed the basis for a 'One-stop multiply-by-7' algorithm for computing the elliptic curve point multiplication operation $W = kP$

	a) Case II (NAF) 2P,4P,8P; 7P = 8P - P				
Coordinate	X		Y		
Operation	A M S I	ST	A M S I	ST	GT
2P	2 4 7 -	13	3 8 6 -	17	30
4P	2 8 5 -	15	3 8 6 -	17	32
8P	2 8 6 -	16	3 10 6 -	19	35
7P	4 10 5 1	20	7 13 4 1	25	45
TOTAL	10 30 23 1	74	16 39 22 1	78	142

	b) Case V indirect 2P Start, 7P=3P+ 4P				
Coordinate	X		Y		
Operation	A M S I	ST	A M S I	ST	GT
2P	2 6 6 -	14	3 10 7 -	20	30
3P	4 8 3 -	15	5 13 2 -	20	32
4P	2 7 4 -	13	3 12 4 -	19	35
7P	3 10 3 1	17	3 13 2 1	19	45
TOTAL	11 31 16 1	59	14 48 15 1	78	137

	c) Case VII Direct 3P, 4P start; 7P=3P + 4P				
Coordinate	X		Y		
Operation	A M S I	ST	A M S I	ST	GT
3P	7 15 10 -	27	11 21 10 -	42	69
4P	6 15 11 -	32	9 21 12 -	42	74
7P	4 10 3 1	18	7 14 3 1	25	43
TOTAL	17 38 21 1	77	21 51 24 1	136	186

Figure 3.18. Comparative cost analysis of $7P$ operation formation.

3.2.2.3.8 The One-stop multiply-by-7 addition/subtraction algorithm

The one-stop multiply-by-7 addition/subtraction chain algorithm for computing elliptic curve point multiplication $W = kP_G = k(x_G, y_G)$ **in affine coordinate system is as shown in figure 3.19. The algorithm requires neither intermediary addition nor doubling EC operations to compute 7P. With *k* in RR7SQSD number system, the scheme simply establishes the appropriate initial condition** $\psi_0 \in \{1P, 2P, 3P\}$ **for the multiply-by-7 addition/subtraction chain in the selector module, compute the 7P using** $7P = 4P + 3P$ **and update** $7\psi_{j-1} + \alpha_{n-j}$**, as appropriate. The set of variables** (B_i, E_i, C_i) **corresponds to the** (x, y) **components and the denominator in the course of the elliptic curve point: doubling** $2P$**, tripling** $3P$**, quadrupling** $4P$**, and sextupling** $7P$ **operations. Similarly, the set of variable names** D_J, F_J, H_J **represents intermediary quantities obtained during the course of the arithmetic operations For example,** $(B_0, E_0, C_0), (B_1, E_1, C_1), (B_2, E_2, C_2), (B_{UD}, E_{Ud}, C_{Ud})$ **represent the** $1P, 2P, 3P$ **module outputs that establish the starting point for the addition chain and the update after a multiply –by-7 operation. In likewise manner,** $A_{17}, C_{27}, D_{37}, E_{77}, B_{UD}$ **are names of variables whose values are computed during the course of computing the point doubling, tripling, quadrupling, sextupling (** $2P, 3P, 4P, 7P$ **) and** α_{n-j} **. Inversion is only at the end of the multiply-by-7 step hence the method is very efficient.**

Selector Modules

1P Module	**2P Module**	**3P Module**
$B_0 = X_G$	$A_1 = 3X_G + A$	$A_2 = Y_G C_1^{\,3} - E_1$
$E_0 = Y_G$	$B_1 = A_1^{\,2} - 8X_G Y_G^{\,2}$	$B_2 = A_2^{\,2} - F_1 D_1^{\,2}$

$C_0 = 1$ $C_1 = 2Y_G$ $C_2 = C_1 D_1$

$X_{UD} = B_0$ $D_1 = X_G C_1^{\,2} - B_1$ $H_2 = B_1 D_1^{\,2} - B_2$

$Y_{UD} = E_0$ $F_1 = X_G C_1^{\,2} + B_1$ $E_2 = A_2 H_2 - E_1 D_1$

$W_{UD} = C_0$ $E_1 = A_1 D_1 - 8Y_G$ $X_{UD} = B_2$

$X_{UD} = B_1$ $Y_{UD} = E_2$

$Y_{UD} = E_1$ $W_{UD} = C_2$

$W_{UD} = C_1$

2P module

$A_{17} = 3X^{2}{}_{I7P} + A$

$B_{17} = A_{17}^{\,3} - 8X_{I7P} Y_{I7P}^{\,2}$

$C_{17} = 2Y_{I7P}$

$D_{17} = X_{I7P} C_{17}^{\,2} - B_{17}$

$F_{17} = X_{I7P} C_{17}^{\,2} + B_{17}$

$E_{17} = A_{17} D_{17} - 8Y_{I7P}^{\,4}$

7P Module

$A_{77} = E_{27} C_{37}^{\,3} - B_{37} C_{27}^{\,3}$

$B_{77} = A_{77}^{\,2} - F_{37} D_{37}^{\,2}$

$C_{77} = C_{27} C_{37} D_{37}$

$H_{77} = B_{37}(C_{27} D_{37})^2 - B_{77}$

$E_{77} = A_{77} H_{77} - E_{37}(C_{27} D_{37})^3$

3P module

$A_{27} = Y_{I7p} C_{17}^{\,3} - E_{17}$

$B_{27} = A_{27}^{\,2} - F_{17} D_{i7}^{\,2}$

$C_{27} = C_{17} D_{17}$

$D_{27} = X_{I7P} C_{27}^{\,2} - B_{27}$

$F_{27} = X_{I7p} C_{27}^{\,2} + B_{27}$

$H_{27} = B_{17} D_{17}^{\,2} - B_{27}$

$E_{27} = A_{27} H_{27} - E_{17} D_{17}^{\,3}$

Update module

$A_{UD} = E_{77} W_{UD}^{\,3} - Y_{UD} C_{77}^{\,3}$

$D_{UD} = B_{77} W_{UD}^{\,2} - X_{UD} C_{77}^{\,2}$

$F_{UD} = B_{77} W_{UD}^{\,2} + X_{UD} C_{UD}^{\,2}$

$B_{UD} = A_{UD}^{\,2} - F_{UD} D_{UD}^{\,2}$

$C_{UD} = C_{77} W_{UD} D_{UD}$

$H_{UD} = X_{UD}(C_{77} D_{UD})^2 - B_{UD}$

$E = A_{UD} H_{UD} - Y_{UD}(C_{77} D_{UD})^3$

4P module

$A_{37} = Y_{I7P} C_{17}^{\,3} - E_{17}$

$B_{37} = A_{37}^{\,2} - F_{27} D_{27}^{\,2}$

$C_{37} = C_{17} D_{17} D_{27}$

$F_{37} = B_{27} C_{37}^{\,2} + B_{37} C_{27}^{\,2}$

$H_{37} = B_{27} D_{27}^{\,2} - B_{37}$

$E_{37} = A_{37} H_{37} - E_{27} D_{27}^{\,3}$

Output module

$$x_{7P} = \frac{B_{77}}{C_{77}^{\,2}} + \frac{B_{UD}}{C_{UD}^{\,2}}$$

$$y_{7P} = \frac{E_{77}}{C_{77}^{\,3}} + \frac{E_{UD}}{C_{UD}^{\,3}}$$

Figure 3.19. One-stop multiply-by-7 EC point multiplication Algorithm.

3.2.2.3.9 Testing the one-stop multiply-by-7 EC point multiplication algorithm by simulation

The aim of this experiment is to prove that point multiplication on elliptic curves defined over RR7SQSD finite fields can be computed using the one-stop multiply-by-7 addition/subtraction chain algorithm. Although to test the algorithm we need the corresponding elliptic curve group generating point $P_G(x_G, y_G)$ **however, for the purpose of this experiment an arbitrary curve point** $P(x, y)$ **believed to be a point on a given elliptic curve was chosen. Let** $P(1,14564)$ **be a point on the elliptic curve** $y^2 = x^3 - 53x + 218$**. It is assumed for now that** $P(1,14564) \in E_{17011}(-53,218)$ **and it is desired to multiply the point** $\mathrm{P}(1,14564)$ **by the scalar quantity** k**. To ensure accuracy of the computed result, the naïve method of computing** $kP = P + P + P + ... + P$ **was used to find the value of** $k = k_{Break}$ **at which** $kP = 0$ **and this was found to be at** $k = 3414$ **when** $W = kP = 3414(1,14564) = (1,2447)$**. The implication of this point is that accurate results are obtainable in the EC point multiplication operation** $k(1,14564) \in E_{17011}(-53,17011)$ **only for values of** $0 < k \le 3414$**. For a generator point of** $P(1,14564)$**. Consequently, the testing of the multiply-by-7 addition/subtraction chain algorithm as a means of computing EC point multiplication commenced. A wide range of specimen values of the scalar quantity** k **for example (** $k \in \{29,156,375,1379,2340,2983,3399,3408,3414,1289,873,6085,2713,619,1191\}$**) were used. Table 3.8 shows the manual sequence of the computation for** $k = 3399$**. The experiment presents the computed results in this manner for any input value of** k **particularly the**

various multiply-by-7 addition/subtraction chain nodes of the corresponding value of k**.**

For example, the multiply-by-7 addition/subtraction chain for $k = 3399$ **is** $1,10,69,486,3399$**. Hence at these nodes of** $k \in \{1,10,69,486,3399\}$**, the points on the curve are** $P(x,y) \in \{(1,14564),(12998,8269),(5142,11411),(3929,15851),(1,2447)\}$ **correspondingly.**

Table 3.8 Sequence of RR7SQSD addition/subtraction based EC point multiplication

I	e_k	Add/Subtraction chain		EC Point multiplication	
4	1	$1P$	$1P$	$1P(x_G, y_G)$	$1P(x_G, y_G)$
3	3	$7(P)$	$7P + 3P = 10P$	$7*(1P(x_G, y_G)) + 3P(x_G, y_G)$	$10P(x_G, y_G)$
2	$\bar{1}$	$7(10P)$	$70P - 1P = 69P$	$7(7*(1P(x_G, y_G)) + 3P(x_G, y_G)) - 1P(x_G, y_G)$	$69P(x_G, y_G)$
1	3	$7(69P)$	$483P + 3P = 486P$	$7(7(7*(1P(x_G, y_G)) + 3P(x_G, y_G)) - 1P(x_G, y_G) + 3P(x_G, y_G)$	$486P(x_G, y_G)$
0	$\bar{3}$	$7(486p)$	$3402P - 3P = 3399P$	$7(7(7(7*(1P(x_G, y_G)) + 3P(x_G, y_G)) - 1P(x_G, y_G) + 3P(x_G, y_G)) - 3P(x_G, y_G)$	$3399P(x_G, y_G)$

The EC point multiplication process was simulated on the Intel Core 2 Duo processor using Quick Basic (QB) version 4.5. Appendix CH 3.6 shows the QB 4.5 source code. The details of the raw results of the simulation are as shown in appendix CH4.4 while a summary of the simulation output is presented and discussed in section 4.4.

3.2.2.4 Methodology for RR7SQSD elliptic curve cryptography

The traditional art of ECC is as represented by the flow chart of fig. 3.20. It entails embedding a given plain text message on the points of an elliptic curve, encrypting the embedded message into a cipher, transmitting the cipher and decrypting the cipher at the receiving end. This section presents the procedures for practical implementation of the art of RR7SQSD ECC that is, the methods for realizing text embedment, message ciphering and message deciphering

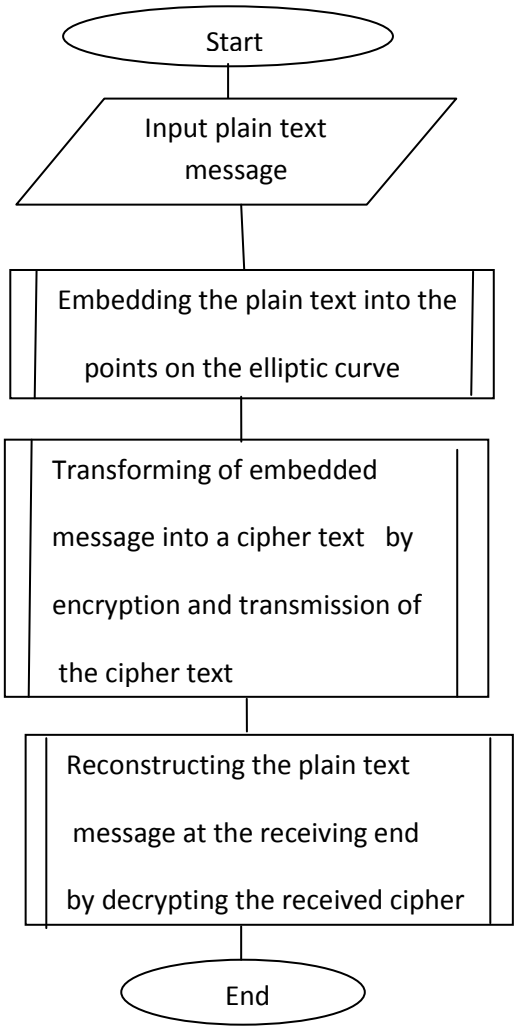


Figure 3.20: Procedure for realizing elliptic curve cryptography

The section begins with the method for implementing plain text message embedment in section 3.2.2.4.1. Coded message encryption and decryption procedure methods are presented in sections 3.2.2.4.2 and 3.2.2.4.3 respectively. An experiment encompassing all the facets of RR7SQSD based ECC is presented in section 3.2.2.4.4. In particular the Diffie- Hellman key exchange protocol is used.

3.2.2.4.1 The Procedure for embedding plain text message on elliptic curves defined over $SGF\left(7^m \pm \psi_i\right)$

The procedure for message embedment consists of three phase. The first is to determine the domain parameters' so as to establish the exact elliptic curve equation and field. The second stage computes: the quadratic residues, the group of rational points $E_q(a,b)$, and its order which is essentially constructing the EC. The third stage encodes the plain text message with the alphabets. Embedding the coded message into the group of rational points $E_q(a,b)$ constitutes the final stage of this experiment. The procedures for the first and second phases are presented next. The third phase will be presented in section 3.2.2.4.2.

The following steps will suffice for constructing the elliptic curve $Eq(a,b): y^2 = x^3 + ax + b \mod q$, defined over $SGF\left(7^m \pm \psi_i\right)$

1a. Select curve coefficients (a,b) and compute the curve order.

1b. Determine a base point $P_G(x_G, y_G)$ often referred to as the generator point which must be of a sufficiently large order such that for the smallest value of k, $kP = 0$

1c. Determine the status of each value of x **i.e., quadratic residue or not, to ensure that the point is in the elliptic group** $E_q(a,b)$

1d. Generate the group $E_q(a,b)$ **that will serve as our reference**

Choose an alphabet with N letters to establish an RR7SQSD coding system.

2a The characters of the alphabet are assigned the numbers 0,1,….., N-1

2b. Obtain the RR7SQSD form of each number

2c Identify characters of the message with the corresponding number codes of the alphabets

2d. Partition very long messages into m **blocks of** l **RR7SQSD and obtain bisection between the** m **plain text blocks and the numbers. That is for any** m_i **partition block, coded** x_{m_i} **is given as:**

$$x_{m_i} = (\alpha_0, \alpha_1, \alpha_2, ..., \alpha_{l-1}) \quad = \sum_{j=l}^{1} \alpha_{l-j} N^{j-1} \qquad \textbf{(3.33)}$$

for $0 \le x_m \le N^l$

3. **Choose an appropriate** k **with a high success probability and that** $q > kN^l$ **.**
4. **Obtain an elliptic group** $P_{i,M}$ **element for each encoded plain text message block.**
5. **Recover the plain text blocks from the point by** $x_{i,m} = \left\lfloor \frac{x}{k} \right\rfloor$

a) Determination of the domain parameters

The optimal extension RR7SQSD element finite field $SGF\left(7^n \pm \psi\right)$ **is particularly chosen for this experiment so that the working elliptic curve is of the form**

$$\left.\begin{array}{l} y^2 = x^3 + ax + b \mod q \;;\quad a,b,x,y \in F_q \\ q = 7^k \pm \psi \;\text{; a prime} \\ \psi \in (\alpha\beta * i + 1) \end{array}\right\} \qquad \textbf{(3.34)}$$

$i = 0,1.2,...$, $k = 1,2,3.,..$

This study ensure cryptographic compliance by avoiding two or more roots of equation (3.35) do not coalesce to give a cusp or non smoothness i.e.; the discriminate

$$\Delta = 4a^3 + 27b^2 \bmod q \neq 0 \qquad \textbf{(3.35)}$$

The pair of values (-53,218) corresponding to a **and** b **satisfy this requirement to guarantee minimum cryptographic security while the prime modulo** q**, was fix at** $q = \left(7^5 + i\alpha\beta\right) = 17011$ **for** $(\alpha,\beta,i) \equiv (3,4,17)$ **. These parameters placed the working elliptic curve equation of this study as presented in equation (3.36)**

$$y^2 = x^3 - 53x + 218 \bmod 17011 \qquad \textbf{(3.36)}$$

This in the RR7SQSD domain can be written as

$$y^2 = x^3 + \bar{1}\,\bar{1}3x + 1\bar{2}\bar{3}1 \bmod 101\bar{3}11 \qquad \textbf{(3.37)}$$

It is now clear that all finite field arithmetic operations are possible in the RR7SQSD domain and since the study is about the organization of a SMVL ECC VLSI circuit, this report continues in the decimal notation to facilitate easy understanding.

b) Computing the quadratic residues

The rational points of the EC group $E_{17011}(-53,218)$ **are generated according to the method outlined in [1,22,45,67,69-70]. Choose** $x \in F_{17011}$ **for which there is a corresponding value of y in the curve. To do this it is necessary to determine the quadratic residues by computing** $x^3 + ax + b$ **for each** $p = 0,1,2,3,...., p-1$ **and use quadratic reciprocity theorem to check if it is a square. For this purpose the Legendre symbol** $\left(\frac{\lambda}{p}\right)$ **which says that for an integer** λ **and prime** $p > 2$

$$\left(\frac{\lambda}{p}\right) = \lambda^{(p-1)/2} \bmod p = \begin{cases} 0 & \text{; if } p \mid \lambda \\ 1 & \text{; if } \lambda \text{ is quadratic residue modulo } p \\ -1 & \text{; if } \lambda \text{ is not a quadratic residue modulo } p \end{cases} \quad \textbf{(3.38)}$$

was employed. Each rational point ϑ_i**, of the elliptic curve group** $E_{17011}(-53,218)$ **was eventually computed using the relation described in equation (3.39).**

$$y_i = \vartheta_i^{(p-1)/4} \quad \textbf{(3.39)}$$

Finally for this stage the order #E (F_q), of the EC group $E_{17011}(-53,218)$ **was calculated using equation (3.49) [74]**

$$\#E(F_q) = \begin{cases} p+1+\sum_{x\in F_q}\left(\frac{x^3+ax+b}{q}\right) & \text{; if } \quad q<10^4 \\ \\ (q+1)-2\sqrt{q} \le \#E(F_q) \le (q+1)+2\sqrt{q} & \text{; if otherwise} \end{cases} \tag{3.40}$$

3.2.2.4.2 Plain text coding and embedment experiment

The purpose of this experiment is to test the procedure of message embedment as outlined in section 3.2.2.4.1 Having identified the domain parameter, determined the exact EC equation determined, established the alphabet to be used and the message was coded, the calculation of the EC rational points and final message embedment is done in accordance with the theory presented in section 3.2.2.4.1. The specimen plain text message used is

"THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG"

The choice of alphabets, the decimal number code as well as the RR7SQSD code are as follows:

i) in RR7SQSD domain the message corresponds to

$$1\bar{1}\,\bar{1}1\bar{1}\bar{2}\bar{2}00310\bar{1}2\bar{2}30\bar{3}\bar{2}0\bar{2}21\bar{3}01120\bar{2}0\bar{1}\,\bar{1}0113\bar{2}0\bar{1}3100\bar{1}021\bar{2}\bar{2}00101\bar{1}\bar{2}1\bar{3}\bar{2}01$$

$$\bar{1}\,\bar{1}1\bar{1}\bar{2}\bar{2}00\bar{2}\bar{2}12\bar{2}\bar{3}\bar{1}\bar{3}01\bar{1}0$$

ii) and in decimal code (for the purpose of clarity we shall work in decimal) corresponds to:

"2008052717210911270218152314142706152427102113161927152205182720080527120126250415 07".

The elements of this code were grouped into blocks of four decimal digits from the right. Table 3.9 shows the alphabet and the corresponding RR7SQSD code in the second column and that of the decimal in the third column. A QB 4.5 source code program was then used to generate the quadratic residues, compute the appropriate rational group of points in $E_{17011}(-53,218)$ **using equations (3.33) to (3.34) and perform the embedment. The QB version source is presented in section Appendix CH 3.4. Manual calculation of all the message points on the EC was also done and was used as a reference. These are shown in table 3.10. Each message block** m_i **in Table 3.9 represents a point on the elliptic curve** $E_{17011}(-53,218)$ **such that the message** m **is seen as an element in** F_{17011}**. Taking a block of message** $x = m_i$ **at a time, the program evaluates** $\lambda = f(x) = x^3 + ax + b \bmod q$ **and tests for quadratic residue modulo** 17011 **of** λ **by computing Legendre's symbol,** $\left(\frac{\lambda}{q}\right) \equiv \lambda^{\left(\frac{q-1}{2}\right)} \bmod q$**. Where a particular block fails the test by evaluating to none other than** 1 **the numeric code of such a block is increased by 1 and the checking process is repeated until** x **now** m'_i **returns a positive check that is ,** $\left(\frac{\lambda}{q}\right) \equiv \lambda^{\left(\frac{q-1}{2}\right)} = 1$**. For example, coding of the fourth block representing the message "LA"=**$0\bar{2}21_{RR7SQSD} \cong 1201_{10}$ **and taking** $x = 1201$**, went as follows:** $\lambda = (1201)^3 - 53*1201 + 218 \bmod 17011 = 13025$ **. Checking for quadratic residue,** $\delta = 13025^{8505} \bmod p \neq 1$**. The value of x was increased in steps of 1 till the value 1204;** $\lambda = 1204^3 - 53*1204 + 218 = 13514$ **;** $\delta = 13514^{8505} \bmod 17011 = 1$**. Consequently, the**

square root is given as $\gamma = 13514^{4252} \bmod 17011 = 15324$**. Hence, the plain text code for the 4th block containing the part of the message "LA" is (1204, 15324). Table 3.10 shows in its last column, the result of manual calculation of the entire message.**

Table 3.9 Alphabet assignment

Symbol	Decimal code	RR7SQSD code
Space	27	$\bar{2}0$
A	1	$\bar{2}1$
B	2	$\bar{2}2$
C	3	$\bar{2}3$
D	4	$\bar{1}\bar{3}$
E	5	$\bar{1}\bar{2}$
F	6	$\bar{1}\bar{1}$
G	7	$\bar{1}0$
H	8	$\bar{1}1$
I	9	$\bar{1}2$
J	10	$\bar{1}3$
K	11	$0\bar{3}$
L	12	$0\bar{2}$
M	13	$0\bar{1}$
N	14	00
O	15	01
P	16	02
Q	17	03
R	18	$1\bar{3}$
S	19	$1\bar{2}$
T	20	$1\bar{1}$
U	21	10
V	22	11
W	23	12
X	24	13
Y	25	$2\bar{3}$
Z	26	$2\bar{2}$

Table 3.10 Coded plain text points $P_{M,i}$

S/N0	Plaintext P_M		$(x, \gamma) \equiv P(x, y)$
	LETTER	CODE	Expected
1	OG	1507	(1507, 16417)
2	SpD	2704	(2704, 393)
3	ZY	2625	(2628, 2950)
4	LA	1201	(1204, 15324)
5	Esp	0527	(528, 12647)
6	TH	2006	(2006, 9629)
7	Rsp	1827	(1828, 1936)
8	VE	2205	(2205, 14215)
9	spO	2715	(2715, 13735)
10	PS	1619	(1619, 915)
11	UM	2113	(2113, 10350)
12	SpJ	2710	(2710, 401)
13	OX	1524	(1525, 4095)
14	SpF	2706	(2708, 11651)
15	WN	2314	(2314, 12327)
16	RO	1815	(1816, 6343)
17	SpB	2702	(2702, 11279)
18	CK	0311	(0311, 4188)
19	UI	2109	(2109, 15967)
20	spQ	2717	(2718, 3716)
21	HE	0605	(610, 3309)
22	T	0020	(0021,08175)

3.2.2.4.3 The Encryption and Decryption operations

The following scenario considered in this study is that a point-to-point communication involves the community of users Michael and Lilian. Michael [M] wants to send a message to Lilian [L] using Diffie-Hellman's protocol [11,22,25,31,68] . Note that followings; equation (3.33) , the arbitrary base point $P_G(x_G, y_G) = (1,14564)$ **and the message already in the form of Table 3.9 has been defined . That is, Michael wants to send the plain text message: "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG " to Lilian, which, he has successfully encoded as in Table 3.9 and embedded in Table 3.10. Michael generates his key pair as follows: He selects a random integer** $1 \le k \le 17011$ **(precisely in this case** $1 \le k \le 3414$**) as his private or secret key. Let this be say** $k_M = 1289$**. He computes his public key by performing elliptic curve point multiplication** $Q_M = k_M G = 1289(1,14564)$**. He obtained** $Q_M = (1517,9225)$ $\in E_{17011}(-53,218)$**. Lilian does the same, she picks a random integer** $k_L = 873$ **(say) and computes her public key** $Q_L = k_L G = 873(1,14564)$**,** $Q_L = (16739,3829). \in E_{17011}(-53,218)$**. Thus, in this communication** $Q_M = (1517,9225)$**,** $Q_L = (16739,3829)$ **are public domains. Michael, with Lilian's public key** $Q_L = (16739,3829)$**, encrypt, all the plain text message points** $\{P_{M,i}\}$**, in Table 3.9, 3rd column , into cipher pair of points using the formular**

$$Q_C = \{(k_M G), (\{P_{M,i}\} + k_M Q_L)\} \quad \textbf{(3.41)}$$

$$= \{(1289 * (1,14564)), (\{P_{M,i}\} + 1289 * (16739,3829))\}$$

For example, for the 4th message point $P_{M,4}$, cipher text is computed to be

$$C_{M\to L,10} = \{(1517,9225),(1204,15324)+1289*(16739,3829)\}$$

$$= \{(1517,9225),(1204,15324)+(7741,10018)\} \qquad = \{(1517,9225),(3081,14317)\}$$

The transmitted cipher text to Lilian by Michael is therefore $\{(1517,9225),(3081,14317)\}$. When Lilian receives this message she uses her private key $k_L = 873$ to compute the plain text point, $P_{M,i}$ in this manner:

$$(P_{M,10} + k_L Q_M) - [k_L (k_M G)] \qquad \textbf{(3.42)}$$

$$= (3081,14317) - [873(1517,9225)]$$

$$= (3081,14317) - [(7741,10018)]$$

$$= (3081,14317) + [(7741,-10018)]$$

$$= (3081,14317) + [(7741,6993)]$$

$$= (1204,15324)$$

This point in table 3.9, row 5, 3rd column contains the plain text "LA". The complete message encoding, encrypting and decrypting experiment will be presented in the next paragraph and the corresponding results in chapter 4 Table 4.2. An eavesdropper cannot do what Lilian does because her secret key is not known to him. To do otherwise, he has to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) that will take him a very long time to solve with all available cutting edge computing

technology systems working simultaneously in parallel.

3.2.2.4.4 An encryption/decryption procedure experiment.

Point multiplication, message embedment, computing the multiplicative inverse of field elements are constituent component operations in the El-Gamal or Diffie-Hellman's key exchange elliptic curve cryptographic protocols. The experiment is a one-all-encompassing simulation program to confirm proper transition of all the various aspects of the presented arithmetic operations necessary to encrypt and decrypt messages with elliptic curves defined over the RR7SQSD element finite field in either of these protocols. Consequently, components of the experiment consist of a quadratic residues and the relevant rational elliptic curve group points generation module, a plain text message embedment module, modules for computing the appropriate public keys of the community of users (Michael and Lilian) in the secured and trusted communication system, modules for encryption by the message sender and a module for decrypting the received cipher message by the authorized receiver. The modules for computing the public keys do multiply-by-7 addition/subtraction chain elliptic curve point multiplication operations. The computed public keys and ciphered blocks are confirmed as true points of the chosen elliptic curve using a curve point's checker module. The input variables were the formal generator point $G(x_G, y_G) = (8360,14564)$ **and the secret keys 1289 and 873 for the sender and message recipients respectively. Again a QB version 4.5 source code program was executed on an Intel Core™2 Duo processor. The program first generate elliptic curve group as it was explained in section 3.2.2.4.1 and**

embed the plain text message "THE QUICK BROWN FOX JUMPS OVER THE OVER THE LAZY DOG" after the encoding process. It then computes the public keys

of Michael and Lilian, encrypted the embedded message using the public key of Lilian, verify that the cipher nodes are all points of the elliptic curve group. Next the program assume the role of the message recipient Lilian, verifies the received message (believed to have been transmitted over an in-secure communication channel) by ensuring that all message nodes are elements of the mutually agreed elliptic curve group and decrypts the cipher using Michaels pubic key. The raw results of this experiment are as shown in Appendix CH 4.5 while a summary of them is presented and discussed in section 4.5.

3.2.3 The VLSI, its basic building block circuit design and synthesis of the functional logic circuits

The methods of realizing the basic building block of the VLSI circuit and the synthesis of the functional logic circuits formulated in section 3.1.3 are presented in this section. In addition the various tests and experiments conducted on the circuits are also presented. Thus, the layout organization of the SMVL ECC arithmetic unit VLSI circuit is presented in section 3.2.3.1 and the multiply-by-7 LSI circuit organization in section 3.2.3.2. The realized transistor circuit of the basic building block herein called the RR7SQSD T-gate circuit is presented in section 3.2.3.3. Testing the building block working according to design specification is presented in section 3.2.3.4 while the simulation experiment to confirm the building block circuit's multiplexing ability is presented in section 3.2.3.5. Synthesis of the functional logic circuits of the multiply-by-7 LSI circuit is presented in section 3.2.3.6. Specific simulation tests conducted to confirm the proper

operation of certain synthesized arithmetic circuits namely, the RR7SQSD full adder and The quasi multiplier are presented in section 3.2.3.7.

3.2.3.1 Organizational layout diagram of the SMVL ECC arithmetic unit VLSI circuit

The block diagram of the proposed VLSI circuit architecture is shown in fig.3.21 with the functions of each block as indicated. The architectural organisation is such that the circuit perform either a one-stop points' addition or point doubling operation on input point(s) $P_u(x_u, y_u)$ **/and** $P_v(x_v, y_v)$ **or a one-stop multiply-by-7 addition/subtraction chain based point multiplication operation** $kG = kP(x_G, y_G)$ **. Hence the organizational layout consist of an EC point's addition/ point doubling block, a field element inversion block and a multiply-by-7 addition/subtraction chain base point multiplication block. All functional logic circuits in any of these subunits were realized using either an RR7SQSD finite field's element parallel multiplication LSI or an RR7SQSD parallel addition LSI circuit.**

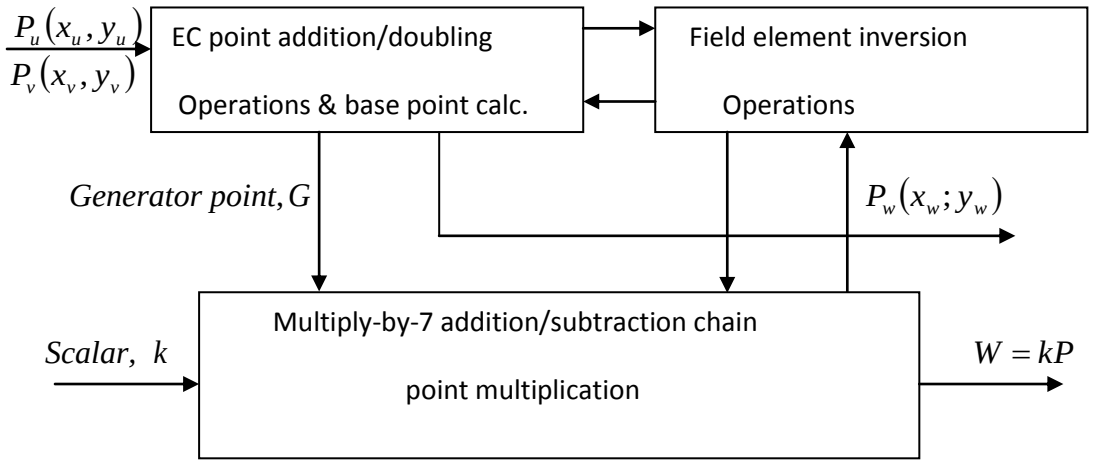


Fig 3.21 Block diagram of the RR7SQSD EC arithmetic unit VLSI

It can be seen that the outputs of the points addition/point doubling or base point generation and the element inversion blocks are all necessary inputs to the multiply-by-7 addition/subtraction chain based point multiplication block. This block thus, is the heart of the VLSI circuit in view hence it was necessary to examine it in detail from the onset. The simplest realization of the type of processor in view is one with the lesser order of computational complexity. Consequently, the presentation starts with the criteria for trading multiplication for addition in section 3.1.3.1. The multiply-by-7 computation unit data flow diagram is presented in section 3.1.3.2. Operation execution speed and number of general purpose registers estimate computations are presented in section 3.1.3.3.

3.2.3.2 Trading multiplication for addition

It has earlier been stated that the order of simplicity in computing elliptic curve arithmetic operations and in particular elliptic curve point multiplication operation is finite field; addition followed by multiplication and then inversion. A fast VLSI circuit is realizable if the design is based on the simplest arithmetic operation. Several efficient strategies for implementing EC point multiplication exist namely, the triple-double-add scheme in [12]. Choosing a particular strategy is often based on a comparative analysis of cost of forming kP . Hence it was necessary to compare the cost of forming kP using the one-stop multiply-by-7 addition/subtraction chain scheme with the triple-double-add scheme using the following numerical example 3.9 as a case study.

Numerical example 3.9 **If P is a point on an elliptic curve and k > 0, make a comparative analysis on the cost to form 314159P using the one-stop RR7SQSD multiply-by-7**

addition/subtraction chain approach and the triple –double–add order of complexity cost computation presented in [12] as reference.

To perform the analysis it was necessary to recall the order of complexity costs of the transformed EC addition/doubling operations and the $2P,3P,4P$ **together with the jump starting** $P_{sel} / 2P_{sel} / 3P_{sel}$ **modules in tables 3.5, 3.6 in section 3.3.1.2 and table 3.8 in section**

3.3.3.5. Consequently, with R representing one multiply-by-7 operation the two options are presented side-by side.

The method as in [12]	**The present effort of one-stop 7P**
314159 = 6.52360 – 1; triple, double-subtract	**314159 = 7.44880 – 1 1R, 1P**
52369 = 8.6545 ; 3 doublings	**44880 = 7.6411 - 3 1R 3P**
6545 = 5. 1091 - 1; Triple, double-subtract	**6411 = 7.916 -1 1R 1P**
109 = 12.91 – 1 ; Triple, double, double-subtract	**1916 = 7.131 - 1 1R 1P**
91 = 18.5 + 1 triple, triple, double-add	**131 = 7.19 - 2 1R 2P**
5 = 6-1 triple, double-subtract	**19 = 7.3 - 2 IR 2P**
	3 = 3P 3P
Total 5 Triple, 4 doublings, 5 double-add	$6R,2(\pm 3P_{sel}),2(-2P_{sel}),3(1P_{sel})$

A similar calculation in [12] gave- 22 doubling and 9 double-add [3]

Also in [12]:

3P or tripling takes 7 [M], 5 [S] and 1[I] on the GF(p) field and 7[M], 4[S]and 1[I] on a binary field,

2P or doubling takes 2]M,] 2[S] and 1[I] for GF(p) fields and 2[M] ,1[S] and 1[I] on the binary field,

2P + Q or Double-Add [DA] takes 9[M] 2[S] and 1[I] on GF(p) and 9[M], 2S and 1[I] for binary field.

Therefore the comparative total cost can be summarized as in table 3.11

Table 3.11 Comparative analysis of cost of forming 314159*P*

	Combined ternary/binary mod 6 approach [12]		RR7SQSD 'one-stop' multiply-by-7 approach
	GF(p)	GF(2)	SGF(7)
2P+Q	5(N/A, 9,2,1)	9(N/A, 9,2,1)	N/A
2P	4(N/A, 2,2,1)	22(N/A, 2,1,1)	N/A
3P	5(N/A,7,4,1)	0(N/A,7,4,1)	N/A
R=7P	N/A	N/A	6[R]+ 1[$3P_{sel}$]/2[$2P_{sel}$]/31[P_{sel}]
Total	N/A[A],95[M],42[S],15 [I]	N/A[A],97[M],41[S],1 7[I]	{168[A], 306[M],102[S],12[I]}+ {(5[A],4[M],3[S]) or (2[A],5[M],4[S])+1[I]}

From the above illustration it can be seen that

a) **the number of multiplication and squaring in [12] are by far less than those of the 'one-stop' multiply-by-7 approach**

b) number of inversion in the 'one-stop' approach is less than those in [12].

b) **Total number of add/subtraction in 'one-stop' is 168 + 5 but the number of add/ subtraction was not computed hence not available (N/A) in [12].**

Further observations:

a) **The aim of this study is to develop a SMVL ECC arithmetic unit VLSI circuit, The approach in [12] had a similar objective but for binary processing threshold logic circuits.**

b) **While in this study inversion is traded for inversion in the work in [12] traded inversion for multiplications.**

c) **Hence the common ground is inversion and the total number of inversions in the 'one-stop' scheme is 12. The work in [12] had 17 inversions thus there is a saving of up to 29%.**

d) **With 1 squaring operation = 0.8 multiplication operation [4], the "one-stop" approach attract a total of 173 additions and 248.8 multiplications.**

It can therefore be concluded that the SMVL ECC arithmetic unit VLSI circuit is better off based on finite field addition, inversion and multiplexing units. For this purpose a data flow diagram of the RR7SQSD EC point multiplication procedure is very necessary.

3.2.3.3 One-stop multiply-by-7 point multiplication operation computation data flow

It has already been established that computation shall be carried out in affine coordinate and that the processing unit that will make the bulk of the functional circuits shall be RR7SQSD addition circuit. Fig.3.22 shows the computation data flow organization of multiply-by-7 addition/subtraction chain based point multiplication on EC defined over RR7SQSD finite fields. On component level it consist essentially of $j-RR7SQSD$ **finite field parallel adders, multiplexers and multiplicative inverters, where** j **is the number of RR7SQSD of the** x, y **coordinates of the elliptic curve base or generator point** $P(x_G, y_G)$**. The multiplexer units MUX, inversion circuits and the RR7SQSD finite field adders are directly SMVL threshold voltage switching to obviate signal level compatibility problems.**

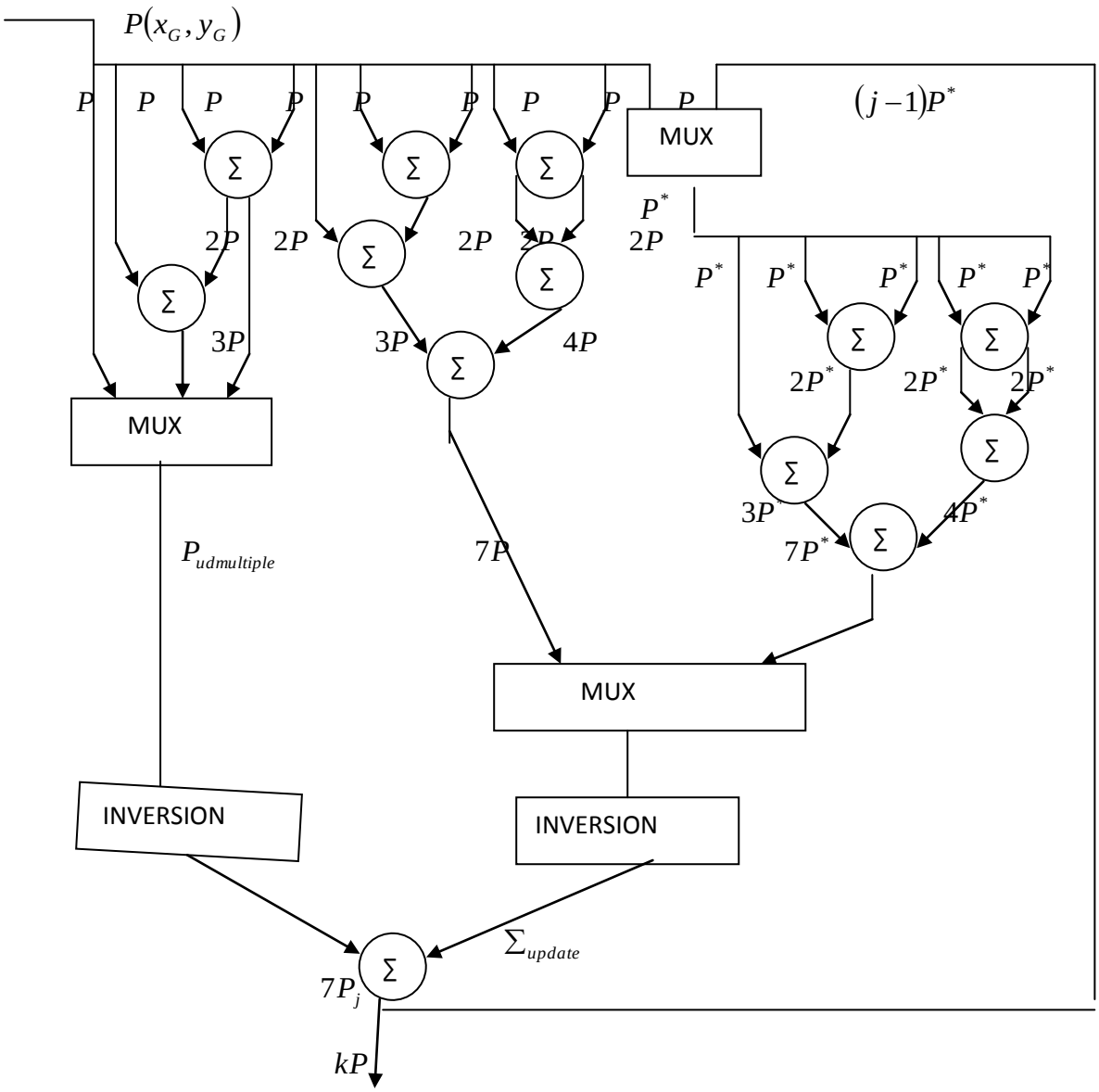


Figure 3.22: Data flow diagram of the point multiplication operation on EC defined over

RR7SQSD finite fields using multiply-by7addition/subtraction chain

All units are realizable using RR7SQSD Complementary Pass (CP) gate derived T-gate circuits hence the speed of the VLSI is a function RR7SQSD adder and field element inversion units.

3.2.3.4 Speed of operation execution and the total number of general purpose registers

The VLSI circuit in view is essentially an arithmetic co-processor and parameter used in judging such a VLSI circuits includes its speed of operation execution and the total number of general purpose registers. These two parameters determine or influence the type of system clock speed and the nature of instruction set respectively to be employed. Since at any given time t the coordinate components x, y of a point are computed in parallel as shown in the diagram of figure 3.23, in these study rough estimates of the circuit speed and total number of the processor registers were estimated using this data flow diagram.

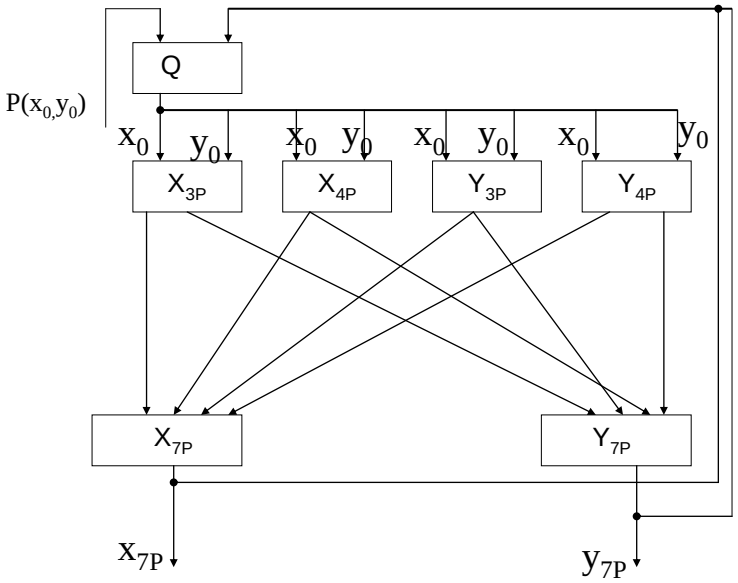


Figure 3.23: Parallel computation of coordinate components x, y

3.2.3.4.1 Estimated speed

If $t_{\Sigma}, t_{INV}, t_m, m$ are the RR7SQSD adder, multiplicative inverse, multiplexer circuits' MVL signal propagation times and the length of m in the RR7SQSD representation respectively then the speed of the point multiplication can be estimated as in equation (3.43).

$$T = 3t_{\Sigma} + t_{INV} + t_{\Sigma} + (m-1)\left[3t_{\Sigma} + t_{INV} + t_{\Sigma}\right]$$

$$= m\left[4t_{\Sigma} + t_{INV}\right] \qquad \textbf{(3.43)}$$

3.2.3.4.2 Total number of general purpose registers

The next question is the total number of general purpose registers required.. For purpose of this exercise the method outlined in [12] is followed and the worst scenario is considered. That is the entire variable are in separate registers.

a) The modified EC points addition and point doubling system of equation (3.33) require a total number of 10 registers i.e.; A,,B,C,G,H,x,y, $\Delta x, \Delta y, \nabla x$.

b) The number required for the one-stop multiply-by-7 EC point multiplication is determined as follows:

i) Selector module: the total number variables are made up of any of these combinations:

$$\begin{pmatrix} A_0 & A_1 & A_2 \\ B_0 & B_1 & B_2 \\ C_0 & C_1 & C_2 \end{pmatrix} \Rightarrow \begin{pmatrix} X_{UD} \\ Y_{UD} \\ W_{UD} \end{pmatrix} + \left\{\left[D_1 \quad E_1 \quad F_1\right] or \left[E_2 \quad H_2\right]\right\} = 3+3=6$$

ii) For one-stop multiply-by-7 EC point multiplication;

Let $\Psi_{ij} \in \{A_{ij}, B_{ij}, C_{ij}, D_{ij}, E_{ij,} F_{ij}, x, y\}$ **represent the variables, and** $T_{ij} \in \{T_{17}, T_{27}, T_{37,} T_{77,}\} \equiv \{2P, 3P, 4P, 7P\}$ **denotes the modules while** T_{Ud} **, denote the update module. Also let a ' + ' denotes that a particular variable operation exist and a ' –' denotes that it does not exist in the specified module of the one-stop multiply-by-7 EC point multiplication. Then the variable-module relation can be described by the presentation on table 3.12**

Table 3.12 Determination of number of registers required

	T_{17}	T_{27}	T_{37}	T_{77}	T_{UD}	$7P$
A_{ij}	+	+	+	+	+	–
B_{ij} j	+	+	+	+	+	–
C_{ij}	+	+	+	+	+	–
D_{ij}	+	–	–	–	+	–
E_{ij}	+	+	+	+	+	–
F_{ij}	+	+	+	–	+	–
H_{ij}	–	+	+	+	+	–
X	–	–	–	–	–	+
Y	–	–	–	–	–	+
Total number of registers for the main module $7x5-5+2=32$						

Hence while only 10 registers are required to compute point's addition or point doubling a total of $32+6=38$ registers are required to compute EC point multiplication using RR7SQSD multiply-by-7 addition/subtraction chain. The points addition/point doubling registers are free the moment a points addition or point doubling operation is completed hence, only a maximum of 38 registers are required for the VLSI circuit. Furthermore, the set of registers for the $2P$ module T_{17}, can be recycled for the update process set T_{UD} and with a base addressing mode of instructions the total number of registers could be reduced to one digit.

3.2.3.5 The block diagram of the one-stop multiply-by-7 point multiplication subunit.

From the data flow analysis of section 3.2.3.3 figure 3.23, three major functions indisputably stand out i.e.; the jump-start selection module generator, multiply-by-7 maker unit and the result update unit as shown in figure 3.24. The $1P,2P,3P$ generator unit determines the required multiple $\in(1P,2P,3P)$ of the base point $P_U(x_U,y_U)=P_G(x_G,y_G)$ based on the RR7SQSD character digit of the scalar k. The generated multiple serves as one input to the update unit. The $3P$ and $4P$ components for the multiply-by-7 calculation ($7P=3P+4P$) are

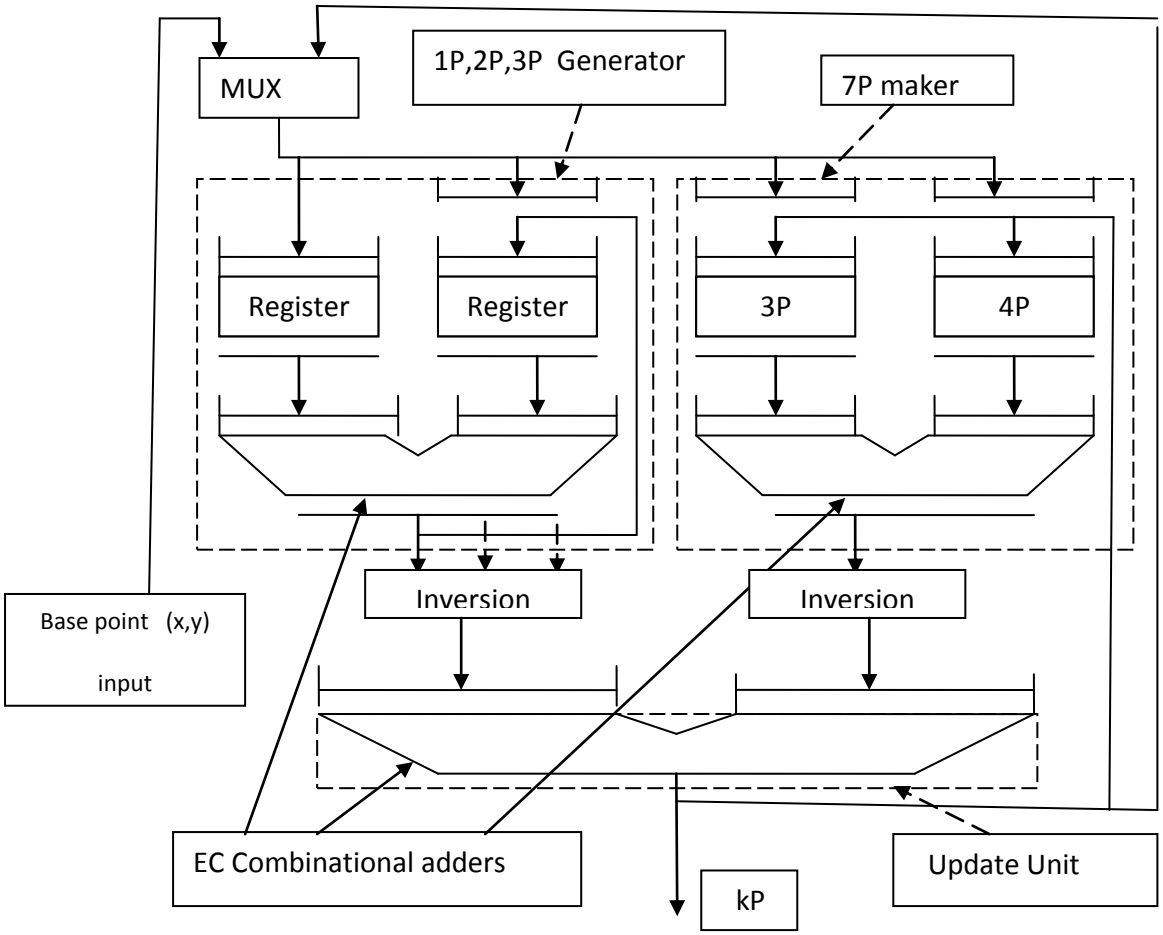


Figure 3.24 Block diagram of the point multiplication unit organization.

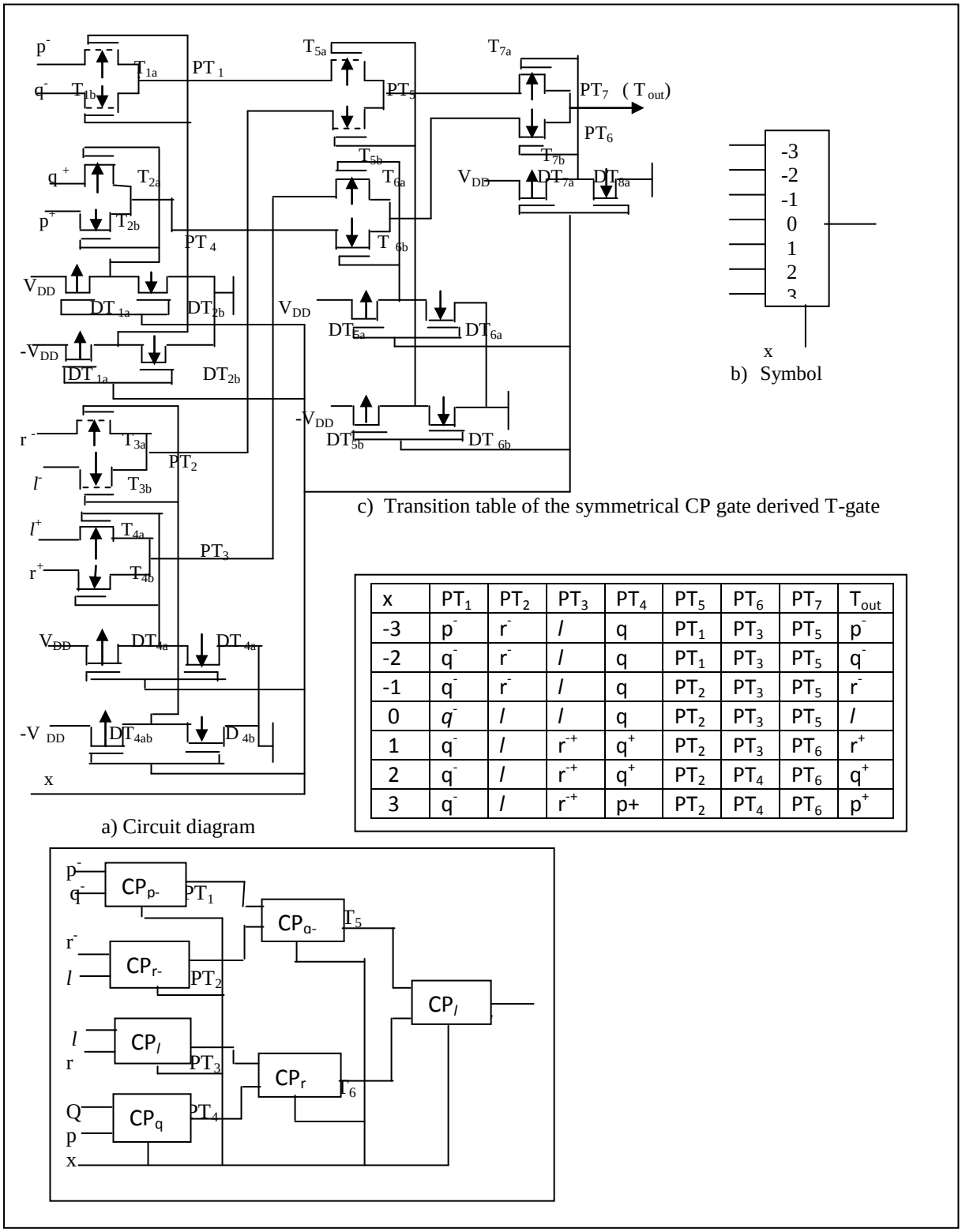
computed per ordinate (figure 3.24) in parallel in the *7P maker* unit whose output serve as the second input to the update adder $\sum_{ud}$ Feeding back the iterated output P^* of this unit through the MUX to the input points of $\sum_{ud}$ perform the $\left(\prod_{j=0}^{n-1} 7^j\right)P$ operation. The circuit require no structural modification to perform $\left(\prod_{j=0}^{n-1} 2^j\right)P$. Remembering that the modification of the traditional EC point's addition and doubling equations explicitly showed that: a) RR7SQSD EC point multiplication is better computed in affine coordinates. b) RR7SQSD addition/subtraction chain EC point multiplication scheme

arithmetic unit is better adder based. Consequently, the major the functional logic circuits for the subunits are: i) RR7SQSD parallel field adders, ii) RR7SQSD multiplexers, iii) RR7SQSD registers, iv) RR7SQSD sign detector/inverter and v) an $SGF(7^m)$ field element inverter. The mathematical model of the basic circuit for the design of these function logic circuits has been formulated in section 3.1.3.5. The design implementations of the $j-RR7SQSD$ finite field parallel adders, multiplexers and multiplicative inverters are based on this basic logic component. The practical realization of the mathematical model and thus its transistor circuit shall now be presented

3.2.3.6 The RR7SQSD T-gate transistor circuit diagram

The processing signal voltage profile has been demonstrated in section 3.1.3.2. Section 3.1.3.3 fully explained the possibility of implementing a practical circuit of a restricted radix-h symmetrical r-nary signed digit T-gate using CP- gates. The constituent components of the building block circuit are the positive literal and negative literal CP-gates presented circuits in section 3.1.3.4. Their parallel interconnection, giving rise to the RR7SQSD T-gate is given in section 3.1.3.5. All these showed that design implementation of an RR7SQSD T-gate circuit using CP- gates is a reality. Consequently, the circuit diagram, the CP-gates layout, the symbol and the truth table of the RR7SQSD T-gate are as shown in figure 3.25. In figure 3.25(a), the inputs p^-, q^-, r^-, l, r, q and p are symmetrical signed -digit functions corresponding to f(-3), f(-2),f(-1),

f(0),f(1), f(2) and f(3) while {PT_i } are the outputs of the various CP_i gates. The enhancement mode transistors T_{jm}, j = 2,4,6,7 and their literal DT_{jm} i = 1,3,4 while m = a or b, realizes the various positive CP gates. The depletion mode transistors T_{jm} circuits (j = 1,3,5) and DT_{im}, (i = 2,4,6) provide the negative CP gates. The voltages -3, -2, -1, 0,1, 2 and 3 represent the symmetrical quaternary logic values –3,-2,-1,0,1,2 and 3 respectively, while V_{DD} and $–V_{DD}$ correspond to +5 and –5 volts. A negative control signal x at the common gates of the literal circuits formed by transistors DT_{2a} and DT_{2b} will turn ON T_{1a} and OFF T_{1b}, if smaller in magnitude, than the threshold value i. T_{1a} is OFF and T_{1b} ON if x is greater in magnitude. Positive control signal x of similar magnitude to the literal circuits of DT_{1a} and DT_{1b} transistors turns ON T_{2a} and OFF T_{2b}, if x < i and T_{2a} OFF T_{2b} ON if x > i, thereby transmitting p^+ and q^+ to point PT_2. Corresponding control signals at the literals: DT_{3a}, DT_{3b} and DT_{4a}, $DT_{4}b$ will pass I^- ,r^- and I^+, r^+ to the points PT_3 and PT_4 respectively. Either of the signals at points $PT_{1\ or\ 2}$ and either of those at nodes $PT_{3\ or\ 4}$ will be pass to nodes PT_5 and PT_6 and subsequently to the T-gates output according to sign and magnitude of the signal at the various literal circuits. The multiple value logic signal propagation time for the RR7SQSD T-gate is $3t_{MOST}$ where t_{MOST} , is the switching speed of the MOSFET . In [40], signal propagation of a CP gate is placed at 100 ns and 200 [pW] - power consumption. This would mean a speed of 300 nanoseconds for the radix – 7 restricted symmetrical quaternary signed digit T-gate.. On transistor count, twenty-eight transistors are required to construct the module, with very remarkable reduced power consumption as against 34 transistors with normal CMOS circuits [102].



x	PT_1	PT_2	PT_3	PT_4	PT_5	PT_6	PT_7	T_{out}
-3	p^-	r^-	*l*	q	PT_1	PT_3	PT_5	p^-
-2	q^-	r^-	*l*	q	PT_1	PT_3	PT_5	q^-
-1	q^-	r^-	*l*	q	PT_2	PT_3	PT_5	r^-
0	q^-	*l*	*l*	q	PT_2	PT_3	PT_5	*l*
1	q^-	*l*	r^{-+}	q^+	PT_2	PT_3	PT_6	r^+
2	q^-	*l*	r^{-+}	q^+	PT_2	PT_4	PT_6	q^+
3	q^-	*l*	r^{-+}	p+	PT_2	PT_4	PT_6	p^+

Figure 3.25: Transistor circuit diagram of the RR7SQSD T-gate

3.2.3.7 Experiments to test the RR7SQSD T-gates parameters and multiplexing ability

Having formulated, designed and realized the basic building block for the function circuits of the arithmetic unit of a SMVL ECC VLSI circuit, this critical block was also tested for electrical parameter conformity. The corresponding results will be presented and discussed in section 4.6.

3.2.3.7.1 Testing for Electrical parameter conformity-Analogue simulation

In this experiment the transfer impedance, transfer function, level of transient, device parameter sweep or input signal propagation time, output signal distortion level measurements and frequency analysis were performed on the RR7SQSD T-gate circuit of figure 3.25. The experiment was conducted on Electronic workbench, Multisim 9 and for this purpose, a special circuit based on the following axiom, was again developed.

Axiom 3.13 **If the sets** $I = \{a,b,c,d\}$**,** $O = \{e, f, z, \gamma\}$ **and** $X = \{x_1, x_2, x_3, x_4\}$ **are some MVL input, output functions and control signal variables correspondingly, such that**

1. $\forall(a,b,c) \ \exists \ (\vec{a},\vec{b},\vec{c})$**,** $\forall \ (e, f, z) \ \exists \ (\vec{e}, \vec{f}, \vec{z})$ **and** $\forall \ (x_1, x_2, x_3) \ \exists \ (\vec{x}_1, \vec{x}_2, \vec{x}_3)$ **while the MVL elements of set** $U = \{d, \gamma, x_4\}$ **are unsigned then a 7-valued symmetrical quaternary-signed threshold MVL T-gate can be constructed to function according to equation:**

$$\gamma = T((\vec{z}, z); X(x_4)) = \begin{cases} \vec{z} & \text{; if } x \le \vec{x}_4 \\ z & \text{; if } x = x_4^{+} \end{cases} \qquad \textbf{(3.44)}$$

provided:

$$\left.\begin{aligned}
\vec{z} = T_{-2}((\vec{e},\vec{f}); X(x_2)) &= \begin{cases} \vec{e} & \text{; if } x \le \vec{x}_2 \\ & \text{; } z = T_1((f,e); X(x_2)) \\ \vec{f} & \text{; if } x = x_2^{+} \end{cases} \quad = \begin{cases} f & \text{; if } x \le \vec{x}_2 \\ & \\ e & \text{; if } x = x_2^{+} \end{cases} \\
\vec{e} = T_{-3}((\vec{a},\vec{b}); X(x_3)) &= \begin{cases} \vec{a} & \text{; if } x \le \vec{x}_3 \\ & e = T_2((b,a); X(x_3)) \\ \vec{b} & \text{; if } x = x_3^{+} \end{cases} \quad = \begin{cases} b & \text{; if } x \le \vec{x}_3 \\ & \\ a & \text{; if } x = x_3^{+} \end{cases} \\
\vec{f} = T_{-1}((\vec{c},d); X(x_1)) &= \begin{cases} \vec{c} & \text{; if } x \le \vec{x}_1 \\ & \text{; } f = T_0((d,c); X(x_1)) \\ d & \text{; if } x = x_1^{+} \end{cases} \quad = \begin{cases} d & \text{; if } x \le \vec{x}_1 \\ & \\ c & \text{; if } x = x_1^{+} \end{cases}
\end{aligned}\right\} \quad \textbf{(3.45)}$$

where $\forall \; T_j ; j \in \{-3,-1,0,2,-2,1,0\}$ **is the literal of corresponding complementary pass gate. Figure 3.26 shows the circuit diagram arising from this axiom constructed with real MOSFETs. The various complementary pass gates' literal circuits and pass transistors interconnections are realized as follows.** CP_{-3} **by transistors Q_1, Q_2,Q_9 and Q_{17};** CP_{-1} **,by transistors Q_{16}, Q_{18},Q_3, and Q_4;** CP_0 **by Q_{22}, Q_{11}, Q_{13}, Q_{24} ;and the set of transistors::Q_{27}, Q_{25}, Q_{28}, and Q_{26} at the output Similarly,** CP_2 **is realized by Q_{19}, Q_7, Q_8 , Q_{23}: and** CP_{-2} **by Q_{10}, Q_5, Q_6 and Q_{15}. DC sources are used to accomplish the necessary literals values of the complementary pass gates. The circuit was then tested for logical accuracy, Similarly, transfer function, transient , d.c transfer sweep , and ac analysis were conducted.**

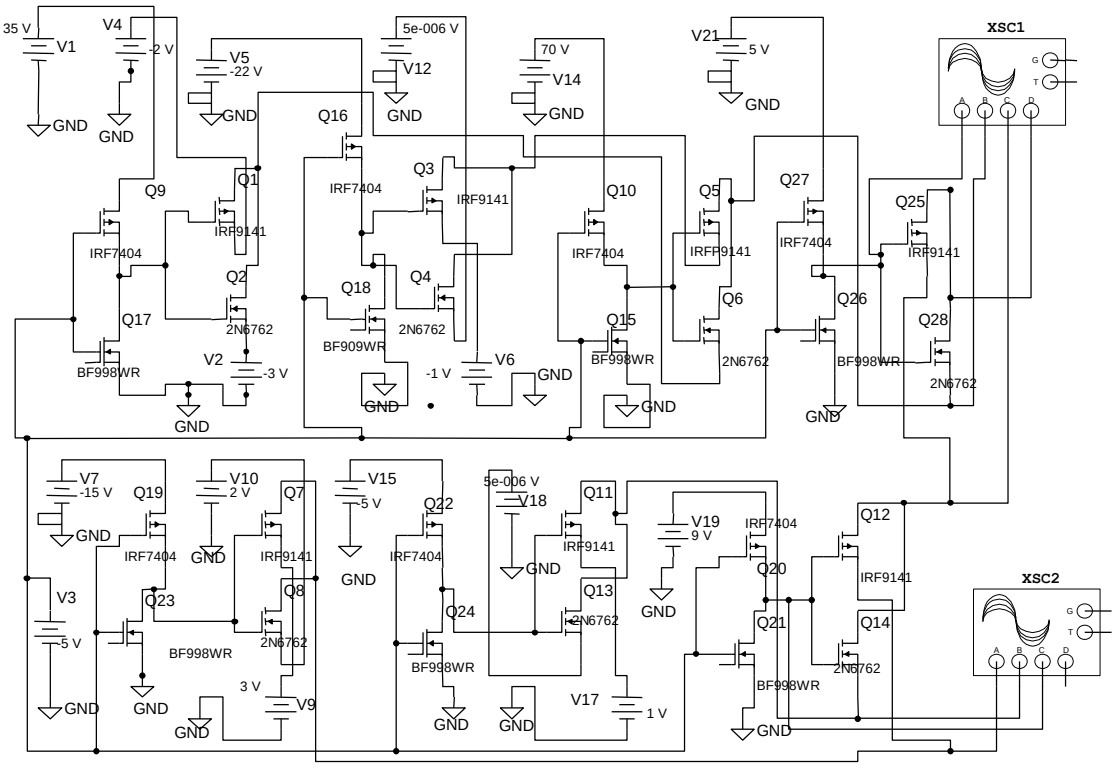
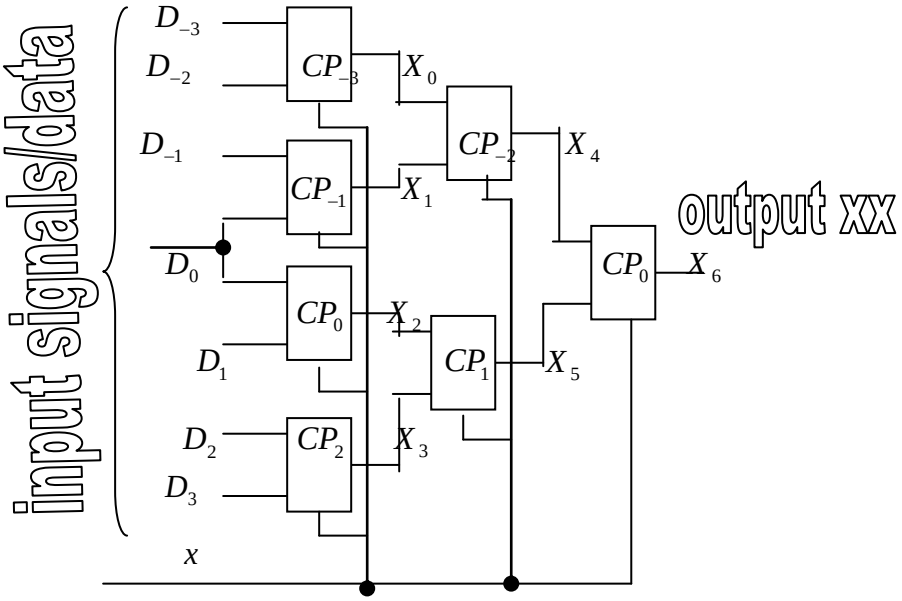


Figure 3.26: Circuit for electrical parameters simulation of the RR7SQSD T-gate.

3.2.3.7.2 Testing for the multiplexing property- Digital simulation

Figure 3.27a shows the schematic diagram of the RR7SQSD T-gate . Its ability to function as a SMVL signals multiplexing circuit was modelled on the hardware description language styled algorithm A2 of figure 3.27b and simulated with a QB 4.5 source code program. A large number, and of varying degrees of complexity test data and control signals: $\{d_i(x_j)\}$, $\{x_j\}$ samples were used in the investigation. The input data $D_i \in \{D_o, D_1, D_2, D_3, D_4, D_5, D_6\}$ selectivity property was monitored at the output node X_6. The simulation source code program is as presented in Appendix CH4. 7 while the results shall presented and discussed in section 4.6.2.



a) Schematic diagram

Input : $\{f_j(x)\}_{j\in\{8,7.6,\ldots,1\}}, x \in \{-3,-2,-1,0,1,2,3\}, lit(x) \in \{-3,-1,0,2,-2,1,0\}, h$

Output: xx

P=3:do while $hh = 8$ downto 1 : $d(p,hh) = f_{hh}(x) : lm(p,hh) = lit(hh)$:loop

do while $c = \left\lceil \frac{h+1}{2} \right\rceil - 1$ downto 1: $g = 2^{c-1}+1 : t = 2^{c-1} : k = 2^c$

NXT: $g = g-1$: if $g < 1$:then NTC:

else if $x < lm(x,g)$ then $d(c-1,t) = d(c,k)$: goto LL:

else $d(c-1,t) = d(c-1,k-1)$

LL: $k = k-2$: if $k < 1$ then NTC:

else $t = t-1$:if $t < 1$ then NTC

Goto NXT:NTC: loop

$x = d(0,1)$:return

b) The layout algorithm

Figure 3.27: Modelling the RR7SQSD T-gate

3.2.3.8 Synthesis of the functional logic circuits

The following functional circuits of the VLSI circuit were synthesized on RR7SQSD T-gate level and tested for specification conformity. Namely:

i) **Dual RR7SQSD addition/multiplication circuits**

ii) **Multiplexer**

iii) **RR7SQSD registers**

iv) **RR7SQSD sign inverter**

v) **RR7SQSD parallel addition/multiplication LSI circuit.**

Brief description of the design implementations and the testing by way of simulation is now presented.

3.2.3.8.1 R**R7SQSD addition/multiplication circuits**

a) **A dual half or one-RR7SQSD adder/multiplier circuit**

The formulation of this circuit has been presented in section 3.1.2.1, equation (3.1) . The circuit performs RR7SQSD $\alpha_i + \beta_i$ **or** $\alpha_i * \beta_i$ **operations on two one-RR7SQSD input operands** $\alpha_i, \beta_i \in \{-3,-2,-1,0,1,2,3\}$ **The operation of this circuit is as described by equation (3.46)**

$$z = \alpha \Theta \beta$$

$$\delta = \begin{cases} \bar{1} & \text{; if } \alpha\Theta\beta < \bar{a} \\ \mathbf{1} & \text{; if } \alpha\Theta\beta > \bar{a} \\ \boldsymbol{0} & \text{; if otherwise} \end{cases} \qquad \textbf{(3.46)}$$

$$\lambda = z - 7\delta$$

where $\delta \in \{-1,0,1\}$, $\lambda \in \{-3,-2,-1,-0,1,2,3\}$ **and** $\Theta \in \{'\oplus', '*'\}$**. The circuit require no separate control signal** x**, to select the either of the two operations the except appropriate input to the T-gates which are the entries of the corresponding truth tables as presented in tables 3.13 and 3.14.**

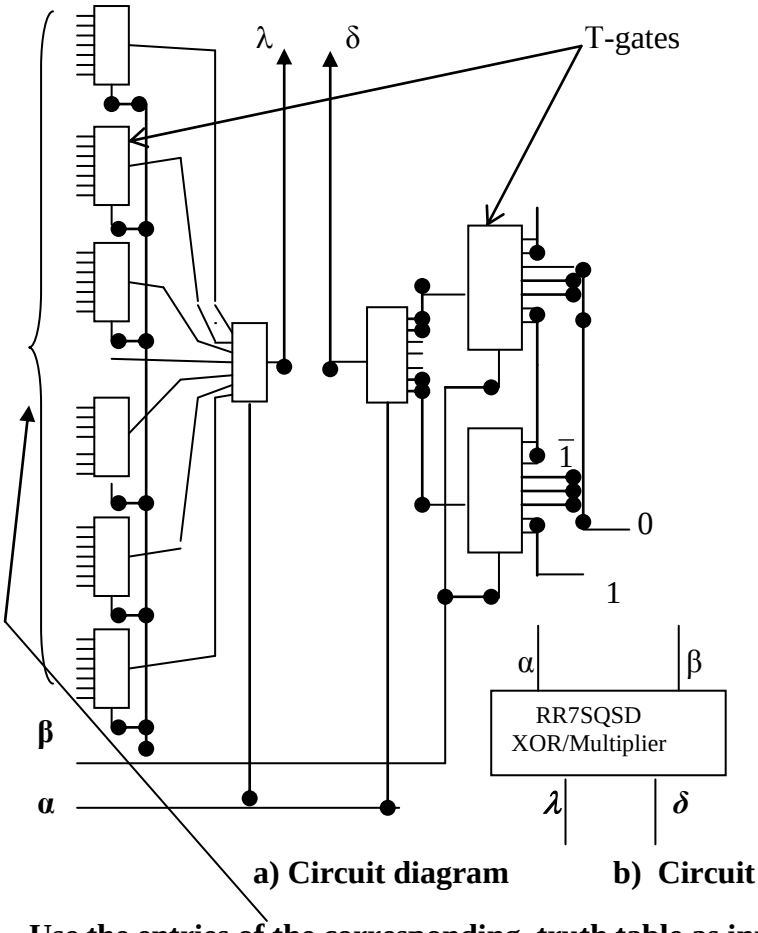
Table 3.13 Half or one- RR7SQSD adder truth table

α/β	Sum digit λ							Carry digit δ						
	-3	-2	-1	0	1	2	3	-3	-2	-1	0	1	2	3
-3	1	2	3	-3	-2	-1	0	-1	-1	-1	0	0	0	0
-2	2	3	-3	-2	-1	0	1	-1	-1	0	0	0	0	0
-1	3	-3	-2	-1	0	1	`2	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-2	-1	0	1	2	3	-3	0	0	0	0	0	0	1
2	-1	0	1	2	3	-3	-2	0	0	0	0	0	1	1
3	0	1	2	3	-3	-2	-1	0	0	0	0	1	1	1

Table 3.14 Transition table of the half or one-RR7SQSD multiplier

a/b	product digit λ							product Carry digit δ						
	$\bar{3}$	$\bar{2}$	$\bar{1}$	0	1	2	3	$\bar{3}$	$\bar{2}$	$\bar{1}$	0	1	2	3
$\bar{3}$	2	$\bar{1}$	3	0	$\bar{3}$	1	$\bar{2}$	1	1	0	0	0	$\bar{1}$	$\bar{1}$
$\bar{2}$	$\bar{1}$	$\bar{3}$	2	0	$\bar{2}$	3	1	1	1	0	0	0	$\bar{1}$	$\bar{1}$
$\bar{1}$	3	2	1	0	$\bar{1}$	$\bar{2}$	$\bar{3}$	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	$\bar{3}$	$\bar{2}$	$\bar{1}$	0	1	2	3	0	0	0	0	0	0	0
2	1	3	$\bar{2}$	0	2	$\bar{3}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	0	0	0	1	1
3	$\bar{2}$	1	$\bar{3}$	0	3	$\bar{1}$	2	$\bar{1}$	$\bar{1}$	0	0	0	1	1

The circuit realization of equation (3.38) on RR7SQSD T-gate level is as shown in figure 3.28.



a) Circuit diagram **b) Circuit symbol**

Figure 3.28 Profile of the half or one-RR7SQSD T-gate dual adder/multiplier

Only a total of 10 RR7SQSD T-gates are required for the circuit's realization. The input operands α and β for the operation are applied to the control points of the T-gates as shown and can be interchange. Inputs to the first 6 out of the 7 T-gates dedicated for the formation of the RRSQSD result digit λ, are the sum/product digit column entries from the respective

truth tables. The 7^{th} T-gate is a crystal selector. Similarly, the first 2 of the carry digit generator obtain their inputs from the respective truth table carry digit columns while the 3^{rd} is the crystal selector. The sum/product and carry digits are formed simultaneously. The signal delay time of the circuit was found to be *0.6μ sec.*

b) A full RR7SQSD dual adder /multiplier circuit

The operation of a full RR7SQSD dual adder/multiplier scheme is as presented by equation (3.2) of section 3.1.3.1. Tables 3.15 and 3.16 shows the corresponding truth tables of a full RR7SQSD adder and a full RR7SQSD multiplication schemes. In the addition mode the circuit is analogous to the full binary adder. The corresponding truth table similar to the truth tables of the half or one-RR7SQSD dual adder/Multiplier circuit, are also identical but differ only in the nature of the set of inputs operands therefore a single circuit on RR7SQSD T-gate level was realized as shown in figure 3.29 to implement the two operations. The implementation requires just 50 RR7SQSD T-gates: 25 for the formation of the sum/product $\gamma_i{}^f$ and 25 for the formation of the carry-out $\delta_i{}^f$. In the 'sum/product' formation section the first 8 T-gates per carry-in $\delta_{i-1} \in \{-1,0,1\}$ situation, giving a total of 24 RR7SQSD T-gates forms the corresponding intermediary 'sums/product digits' and the 25TH serve as the chip selector. The inputs to the first 21 of the 25 intermediary 'sums' forming T-gates is the add-end/multiplicand α_i. The outputs of these feeds into the next set of 3 T-gates: one per carry-in situation. The aug-end/multiplier β_i serves as the SMVL control signal for the entire 24 T-gates. The outputs of this set of 3 T-gates in turn feeds the {-1,0,1} inputs

of the 25TH gate that uses the carry-in signal as its control. Similar organization can be seen in the carry-out $\delta_i{}^f$ section except that the inputs for the first 21 gates as shown in design truth table comprises $-1,0,1$ only. These variables are generated simultaneously during the course of the addition operation. Experiments shall be conducted to confirm the circuit's functional capability later in the next section.

Table 3.15 **Truth table of the RR7SQSD full adder**

carry out / carry in	$\gamma^f{}_i$								$\delta^f{}_{i+1}$						
	β_i / α_i	-3	-2	-1	0	1	2	3	-3	-2	-1	0	1	2	3
$\delta^f{}_{i-1} = -1$	-3	0	1	2	3	-3	-2	-1	-1	-1	-1	-1	0	0	0
	-2	1	2	3	-3	-2	-1	0	-1	-1	-1	0	0	0	0
	-1	2	3	-3	-2	-1	0	1	-1	-1	0	0	0	0	0
	0	3	-3	-2	-1	0	1	2	-1	0	0	0	0	0	0
	1	-3	-2	-1	0	1	2	3	0	0	0	0	0	0	0
	2	-2	-1	0	1	2	3	-3	0	0	0	0	0	0	1
	3	-1	0	1	2	3	-3	-2	0	0	0	0	0	1	1
$\delta^f{}_{i-1} = 0$	-3	1	2	3	-3	-2	-1	0	-1	-1	-1	0	0	0	0
	-2	2	3	-3	-2	-1	0	1	-1	-1	0	0	0	0	0
	-1	3	-3	-2	-1	0	1	2	-1	0	0	0	0	0	0
	0	-3	-2	-1	0	1	2	3	0	0	0	0	0	0	0
	1	-2	-1	0	1	2	3	-3	0	0	0	0	0	0	1
	2	-1	0	1	2	3	-3	-2	0	0	0	0	0	1	1
	3	0	1	2	3	-3	-2	1	0	0	0	0	1	1	1
$\delta^f{}_{i-1} = 1$	-3	2	3	-3	-2	-1	0	1	-1	-1	0	0	0	0	0
	-2	3	-3	-2	-1	0	1	2	-1	0	0	0	0	0	0
	-1	-3	-2	-1	0	1	2	3	0	0	0	0	0	0	0
	0	-2	-1	0	1	2	3	-3	0	0	0	0	0	0	1
	1	-1	0	1	2	3	-3	-2	0	0	0	0	0	1	1
	2	0	1	2	3	-3	-2	-1	0	0	0	0	1	1	1
	3	1	2	3	-3	-2	-1	0	0	0	0	1	1	1	1

Table 3.16 **Truth table of the RR7SQSD quasi-multiplier**

α_i / β_k		Product digit $\rho^1{}_j$							Product carry-out $\rho^o{}_j$						
		-3	-2	-1	0	1	2	3	-3	2	1	0	1	2	2
$\rho^o{}_{j-1} = -1$	-3	1	-2	2	-1	3	0	-3	1	1	0	0	-1	-1	-1
	-2	-2	3	1	-1	-3	2	0	1	0	0	0	0	-1	-1
	-1	2	1	0	-1	-2	-3	3	0	0	0	0	0	0	-1
	0	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0
	1	3	-3	-2	-1	0	1	2	-1	0	0	0	0	0	0
	2	0	2	-3	-1	1	3	-2	-1	-1	0	0	0	0	1
	3	-3	0	3	-1	2	-2	1	-1	-1	-1	0	0	1	1
$\rho^o{}_{j-1} = 0$	-3	2	$\bar{1}$	3	0	$\bar{3}$	1	$\bar{2}$	1	1	0	0	0	$\bar{1}$	$\bar{1}$
	-2	$\bar{1}$	$\bar{3}$	2	0	$\bar{2}$	3	1	1	1	0	0	0	$\bar{1}$	$\bar{1}$
	-1	3	2	1	0	$\bar{1}$	$\bar{2}$	$\bar{3}$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	$\bar{3}$	$\bar{2}$	$\bar{1}$	0	1	2	3	0	0	0	0	0	0	0
	2	1	3	$\bar{2}$	0	2	$\bar{3}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	0	0	0	1	1
	3	$\bar{2}$	1	$\bar{3}$	0	3	$\bar{1}$	2	$\bar{1}$	$\bar{1}$	0	0	0	1	1
$\rho^o{}_{j-1} = 1$	-3	3	0	-3	1	-2	2	-1	1	1	1	0	0	-1	-1
	-2	0	-2	3	1	-1	-3	2	1	1	0	0	0	0	-1
	-1	-3	3	2	1	0	-1	-2	1	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
	1	-2	-1	0	1	2	3	-3	0	0	0	0	0	0	1
	2	2	-3	-1	1	3	-2	0	-1	0	0	0	0	1	1
	3	-1	2	-2	1	-3	0	3	-1	-1	0	0	1	1	1

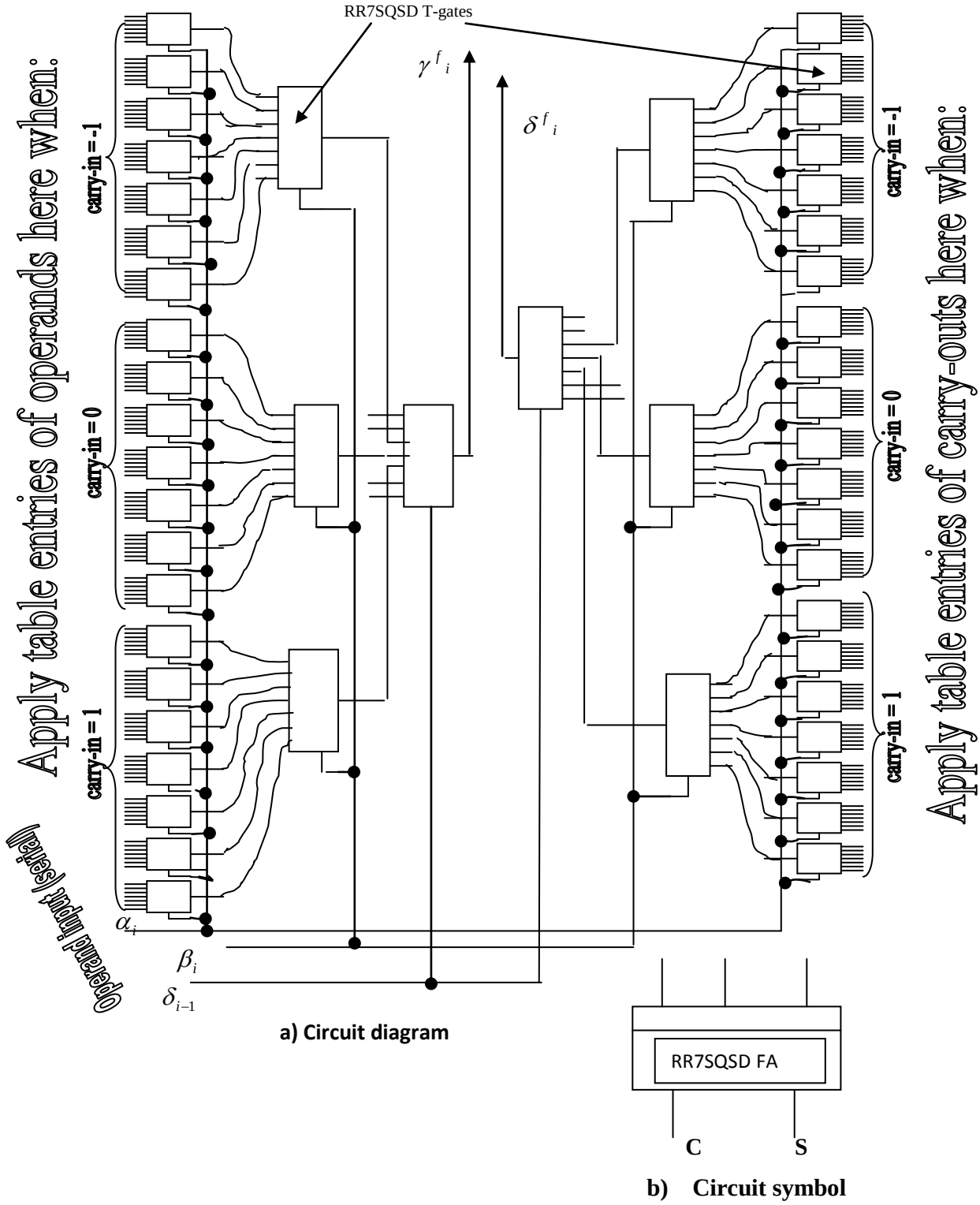


Figure 3.29: RR7SQSD T-gate dual full adder/multiplier

c) RR7SQSD non-finite field parallel addition LSI architecture

The RR7SQSD Half Adder (HA) otherwise the RR7SQSD XOR and the RR7SQSD Full Adder (FA) circuits have been presented in sections 3.6.1.1 and 3.6.1.2 respectively. A cascade of the RR7SQSD FA performs parallel addition but such circuits are always berserk with long carry propagation chain and the RR7SQSD parallel addition circuit is no exception. In extremely long word length operands the carry chain propagates horizontal and vertical such that the horizontal component results in device slow speed and the vertical component results in expensive hardware in order to sustain accuracy of computed results. In section 3.1.4, this problem was adequately analyzed and an efficient solution in the form of equation (3.3) was equally proffered. The numerical example 3.1 was also used to exemplify the algorithm's accuracy however; the hardware realization was then deferred. For the purpose of continuity the example is reproduced here in figure 3.30.

It can be seen that in figure 3.30 only 4 steps as against 9 in the previous case, are required to arrive at the final solution when using the algorithm of equation (3.3). From the right, the first two columns do not satisfy the specified conditions so equation (3.2) is used to compute their sums. However, a "1" carry generated from the 2nd column into the 3rd column where $\alpha_i + \beta_i = 3$ triggering a carry propagation chain, made the 3rd column to qualify for action. The same condition applies in 4 , 8, 9, 10, 14, 15 and 17. With a ratio of $4:8$ there is a saving of almost 50% in speed.

$\alpha = 2\bar{2},\bar{3}\bar{2}\bar{2},\bar{3}\bar{1}\bar{2},302,\bar{3}0\bar{2},332$

$\beta = 2\bar{2},0\bar{1}\bar{2},0\bar{2}\bar{2},132,0\bar{3}\bar{2},023$

$\bar{3}3,\bar{3}\bar{3}3,\bar{3}\bar{3}3,\bar{3}\bar{3}3,\bar{3}\bar{3}3,3\bar{2}\bar{2}$

$1\bar{1}0,0\bar{1}0,0\bar{1}1,010,0\bar{1}0,11$

$133,\bar{3}33,\bar{3}\bar{3}3,\bar{3}\bar{3}3,\bar{3}33,\bar{3}\bar{1}\bar{2}$

$0,\bar{1}00,\bar{1}00,\bar{1}10,100,\bar{1}01$

$033,3\bar{3}\bar{3},333,\bar{2}\bar{3}3,33\bar{3},\bar{3}\bar{1}\bar{2}$

$000\bar{1},00\bar{1},100,00\bar{1},01$

$32,332,\bar{3}\bar{3}\bar{3},\bar{2}\bar{3}\bar{3},33\bar{3},\bar{3}\bar{1}\bar{2}$

$000,001,000,0\bar{1}0,10$

$32,333,\bar{3}\bar{3}\bar{3},\bar{2}33,\bar{3}\bar{3}\bar{3},\bar{3}\bar{1}\bar{2}$

$000,000,000.\bar{1}01,00$

$32,333,\bar{3}\bar{3}\bar{3},\bar{2}3\bar{3},\bar{3}\bar{3}\bar{3},\bar{3}\bar{1}\bar{2}$

$000,000,000,010$

$32,333,\bar{3}\bar{3}\bar{3},\bar{3}\bar{3}\bar{3},\bar{3}\bar{3}\bar{3},\bar{3}\bar{1}\bar{2}$

$000,000,000,10$

$032,333,\bar{3}\bar{3}\bar{3},\bar{2}\bar{3}\bar{3},\bar{3}\bar{3}\bar{3}.\bar{3}\bar{1}\bar{2}$

Eight steps are required

(I) $\alpha = 2\bar{2},\bar{3}\bar{2}\bar{2},\bar{3}\bar{1}\bar{2},302,\bar{3}0\bar{2},332$

$\beta = 2\bar{2},0\bar{1}\bar{2},0\bar{2}\bar{2},132,0\bar{3}\bar{2},023$

$= \bar{3}3,\bar{3}\bar{3}3,\bar{3}\bar{3}3,\bar{3}\bar{3}\bar{3},\bar{3}\bar{3}3,3\bar{2}\bar{2}$

(II) $1\bar{1}0,0\bar{1}0,0\bar{1}1,010,0\bar{1}0,11$

$0\bar{3}0,0\bar{3}0,0\bar{3}3,030,0\bar{3}0300$

$103,\bar{3}03,\bar{3}00,\bar{3}0\bar{3},\bar{3}03,3\bar{1}\bar{2}$

(III) $\bar{1}00\bar{1}00\bar{1}10100\bar{1}0100$

$030,030,03\bar{3},0\bar{3}0,0\bar{3}0,000$

$033,333,3\bar{3}\bar{3},\bar{2}\bar{3}\bar{3},33\bar{3},\bar{3}12$

(IV) $00\bar{1},00\bar{1},100,00\bar{1},110,00$

$000,001,000,001,000,000$

$032,333,\bar{3}\bar{3}\bar{3},\bar{2}\bar{3}\bar{3},\bar{3}\bar{3}\bar{3},\bar{3}\bar{1}\bar{2}$

The process levels

Correction action

These "+1"s propagates to the next column

To neutralize the"-1"s

with the present algorithm only four steps are required

Figure 3.30: RR7SQSD multi-digit addition numerical example reproduced

The circuit designed to realize this achievement functioned, by analyzing two neighbouring pairs of input operands (a_i, b);(a_{i-1}, b_{i-1}) to either generate an expected carry out to ensure result accuracy or neutralize an undesired carry out leading to improves system speed. The

RR7SQSD conventional parallel addition LSI circuit architecture is as shown in figure 3.31. With two signed digits δ_i α_i per character digit ε_i the length of an operand is doubled and the necessary vertical stages of large operand parallel addition schemes increases. The circuit proposed arises from the dual RR7SQSD full adder/multiplier scheme presented in equation (3.2) hence in principle it is similar to the parallel transfer's schemes advocated in [103] and repeated transfer in [48]. It considers the three-in-one carry status of the previous operation: δ_{1i-1}, δ_{2i-1} and δ_{3i-1}, and generate their corresponding new status δ_{1i}, δ_{2i}, and δ_{3i} to the (i +1)th digit in the ith digit stage , while evaluating the ith digit grand final sum digit γ_i.

The scheme is implemented with four levels of addition operation in two phases. The first phase comprising levels 1 to 3, identify potential error generating input operand pairs and formulate correction signals sets $\{\tau'_{1i},\tau''_{1i}\} \in \{(a,\bar{a}),(\bar{a},a)\}$ on identification and {(0,0)} on a negative search. Identification of error prawn input pairs and the formulation of the appropriate correction set $\{\tau'_{1i},\tau''_{1i}\}$ are carried out in the Filtra block where the magnitudes of two neighbouring input operand pairs (α_i , β_i), (α_{i-1} , β_{i-1}) are analyzed simultaneously to generate the appropriate combination of $\{\tau'_{1i},\tau''_{1i}\} \in \{(3,\bar{3}),(\bar{3},3),(0,0)\}$. Adding τ'_{1i} to the interim sum τ_{1i} in the second phase to temporarily nullify the interim sum, generate and send a carry neutralizing signal to the next level to counter the already propagating carry δ_{1i}. The interim sum now appearing as τ_{3i} is re-established at the third level by the adding τ''_{1i} to τ_{2i}.

The final carry out $\delta^{f}{}_{i}$, at any *iteration* is the sum of the third level carry δ_{3i} and the ultimate carry-in $\delta^{f}{}_{i-1}$. Simultaneously, at this level the final sum $\gamma_i = \tau_{3i} + \delta^{f}{}_{i-1}$ is processed.

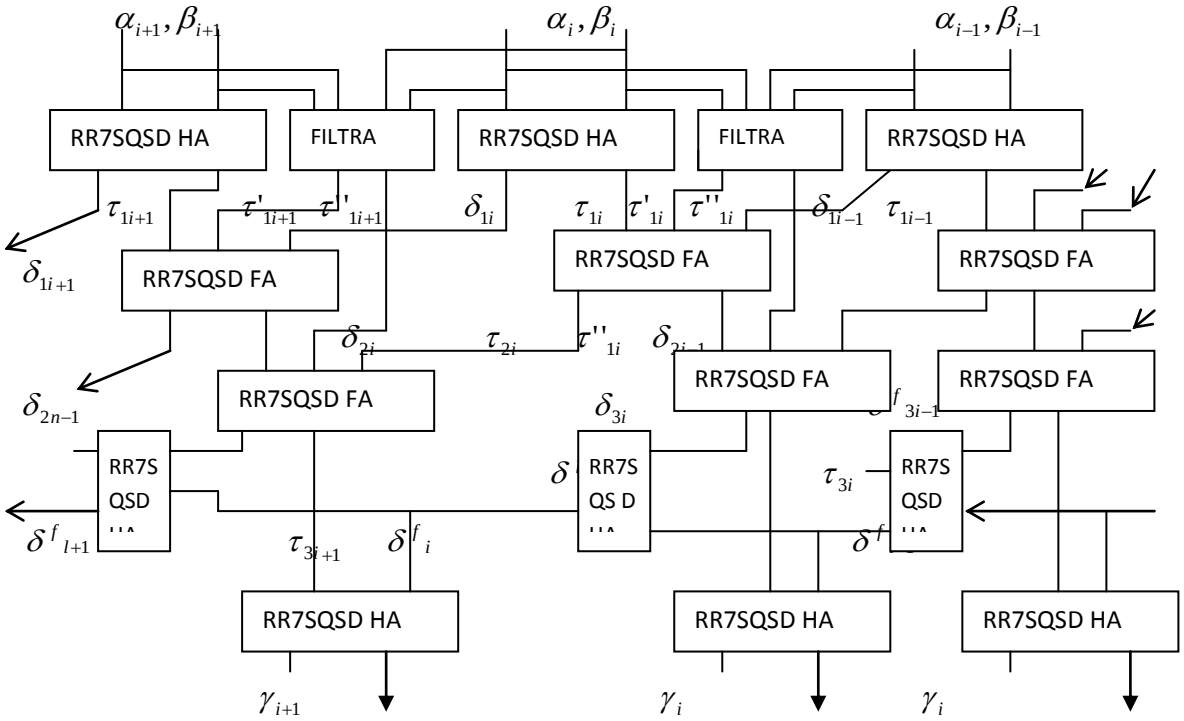
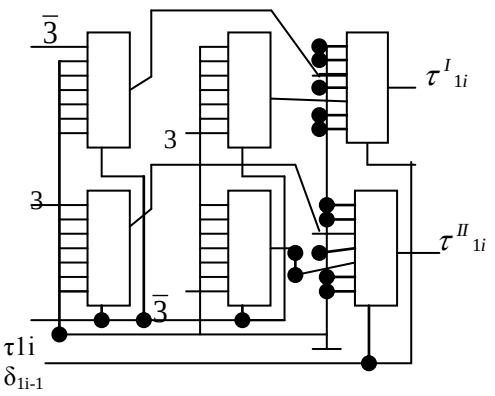


Figure 3.31: A 3-RR7SQSD parallel addition LSI circuit

$\tau^{I}{}_{1i}$

α_i / β_i	-3	-2	-1	0	1	2	3
-3	0	0	3	$\bar{3}$	0	0	0
-2	0	3	$\bar{3}$	0	0	0	0
-1	3	$\bar{3}$	0	0	0	0	0
0	$\bar{3}$	0	0	0	0	0	3
1	0	0	0	0	0	3	$\bar{3}$
2	0	0	0	0	3	$\bar{3}$	0
3	0	0	0	$\bar{3}$	3	0	0



i)Truth table **ii) Circuit diagram**

Figure 3.32: The filtra unit of an RR7SQSD parallel addition LSI circuit

Figure 3.32 shows the design of the LSI circuit's Filtra unit. The corrected carryout δ^{f}_{i-1} delay is *0.3 µsec.* The computed time to form a digit's final sum γ_i , is *1.8 µsec;* as against *0.9 x m sec* where *m*, is the count of the i^{th} digit from the right. Speed of an addition operation is given by (*1.8 + 0.3l_n) µsec*, where l_n , is the radix 7 SD input operands length.

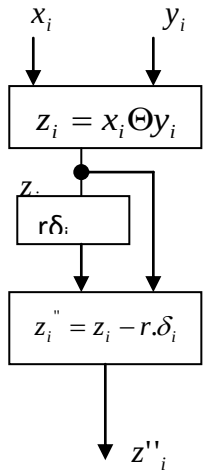
3.2.3.8.2 Dual one- $SGF(7)$ **field element addition/multiplication unit**

The half or one-RR7SQSD adder/multiplier scheme presented in section 3.6.1.1 is capable of performing a one- $SGF(7)$ **field element addition/multiplication operation provides the carry-out digit is ignored. Thus, in the addition mode the operation is essentially an RR7SQSD XOR analogous to the binary XOR operation. Consequently, the behaviour of the circuit can be once again summarized in the manner of equation (3.47) and the diagram of figure 3.33.**

$$z_i = x_i \Theta y_i$$

$$\delta_i = \begin{cases} \bar{1} & ; \text{if} \quad z_i < \bar{a} \\ 0 & ; \text{if} \quad |z_i| < a \\ 1 & ; \text{if} \quad z_i > a \end{cases} \qquad (3.47)$$

$$z_i^{''} = z_i - r.\delta_i$$



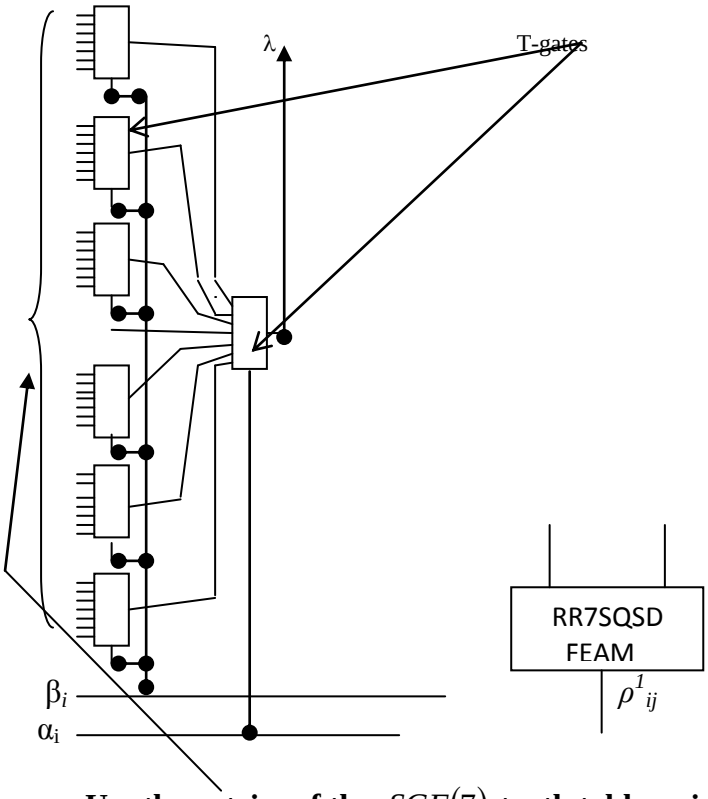
where $(a, r) = (3,7)$, $\oplus$ **denotes RR7SQSD XOR/multiplication and** $x, y \in SGF(7)$

a) Formular **b) schematic diagram**

Figure 3.33: $SGF(7)$ **field elements dual XOR/multiplication scheme.**

Table 3.17 $SGF(7)$ **field element addition/multiplication truth table.**

a_i / b_i	Addition operation							Multiplication operation						
	$\overline{3}$	$\overline{2}$	$\overline{1}$	0	1	2	3	$\overline{3}$	$\overline{2}$	$\overline{1}$	0	1	2	3
$\overline{3}$	1	2	3	-3	-2	-1	0	2	$\overline{1}$	3	0	$\overline{3}$	1	$\overline{2}$
$\overline{2}$	2	3	-3	-2	-1	0	1	$\overline{1}$	$\overline{3}$	2	0	$\overline{2}$	3	1
$\overline{1}$	3	-3	-2	-1	0	1	2	3	2	1	0	$\overline{1}$	$\overline{2}$	$\overline{3}$
0	-3	-2	-1	0	1	2	3	0	0	0	0	0	0	0
1	-2	-1	0	1	2	3	-3	$\overline{3}$	$\overline{2}$	$\overline{1}$	0	1	2	3
2	-1	0	1	2	3	-3	-2	1	3	$\overline{2}$	0	2	$\overline{3}$	$\overline{1}$
3	0	1	2	3	-3	-2	1	$\overline{2}$	1	$\overline{3}$	0	3	$\overline{1}$	2



Use the entries of the $SGF(7)$ **truth table as inputs here**

a) circuit diagram **b) add/Multiply circuit symbol**

Figure 3.34: $SGF(7)$ **elements add/multiply circuit.**

The combined truth table of the circuit is as shown in table 3.19 and the circuit's realization is as presented in figure 3.34. The circuit is a basic component for high speed RR7SQSD addition and multiplier LSI circuit's implementation.

3.2.3.9 High speed $SGF(7^m)$ field element addition and multiplication LSI circuits

3.2.3.9.1 Parallel $SGF(7^m)$ field element addition circuit

An $SGF(7^m)$ or multi-RR7SQSD finite field element adder is an $n-string\ SGF(7)$ field elements or word adder consequently, the $SGF(7^m)$ field element addition operation is a parallel RR7SQSD component-wise addition $\lambda_i = \alpha_i \oplus \beta_i$ operation and its realization is simply a row of *n* one-$SGF(7)$ field element dual addition/multiplication units or RR7SQSD XOR/multiplier modules connected in tandem with un-utilized carry-ins/out so that $SGF(7^m)$ field element addition operation can therefore be done at constant tine.

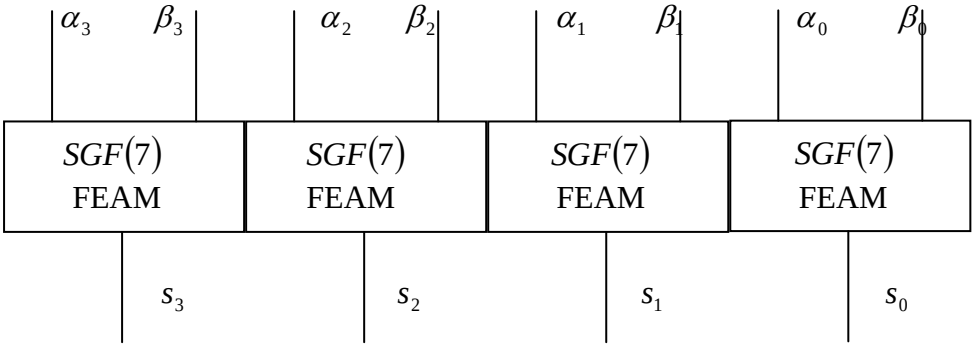


Figure 3.35: $SGF(7^m)$ field element addition circuit.

$s_i = \alpha_i \oplus \beta_i$ where $\oplus$ denotes RR7SQSD addition or XOR. Figure 3.35 shows the realization on the one-RR7SQSD Field Element Addition/Multiplication (FEAM) circuit

level. It can be observed that the signal delay period of the circuit is just twice the signal propagation time of an RR7SQSD T-gate.

3.2.3.9.2 The $SGF(7^m)$ field elements multiplication LSI circuit

It has been established that if α_j, β_j are two RR7SQSD such that $\alpha_j, \beta_j \in \{-3,-2,-1,0,1,2,3\}$ then the product $\lambda_j = \alpha_j \beta_j$, of their multiplication operation, consisting of two RR7SQSD digits $\rho^0_j \rho^1_j$ may be described as in equation (3.48)

$$\left.\begin{array}{l} \lambda_j = \alpha_j * \beta_j \\ \rho^0_j = \begin{cases} \bar{1} & \text{; if } \lambda_j < \bar{a} \\ 1 & \text{; if } \lambda_J > \bar{a} \\ 0 & \text{; if otherwise} \end{cases} \\ \rho^1_j = \lambda_j - 7\rho^0_j \end{array}\right\} \quad (3.48)$$

where $\rho^0_j \in \{-1, 0, 1\}$ and $\rho^1_j \in \{-3,-2,-1,0,1,2,3\}$. It is also a known fact it is true if $\alpha_j, \beta_j \in SGF(7)$ and extended the above method to $SGF(7^m)$. Hence, let the field elements $a(t), b(t) \in SGF(7^m)$ and the prime modulo $w(t)$ be represented in RR7SQSD polynomial basis i.e. $a(t) = a_3t^3 + a_2t^2 + a_1t + a_0$, $b(t) = b_3t^3 + b_2t^2 + b_1t + b_0$ and $w(t) = t^5 + w_4t^4 + w_3t^3 + w_2t^2 + w_1t + w_0$ besides $w(t) \in SGF(7^m)$. Then the field multiplication operation a(t) * b(t) can be represented by equation (3.49):

$$\text{a(t) * b(t)} = \{ \sum_{i=0}^{k+r-i} (\sum_{n=0,m=0}^{n=k-1,m=r-1} a_{k-n} b_{r-m} - 7p^0{}_{mn}) \text{ mod w(t)} \} \text{ mod w(t)}$$

$$= \{ \sum_{i=0}^{k+r-1} d_i t^i \text{ mod w(t)} \} \text{mod w(t)} \quad (3.49)$$

where $d_i = a_{k-n}b_{r-m} - 7\rho^0_h$ as earlier explained .

Basic steps involved in executing any multiplication operation are partial product generation and partial product accumulation [102]. Speed-up strategies particularly necessary for very long operands are reduction of number of partial products and/or development of very fast techniques to accumulate the partial products [63,104]. Hence, high-speed conventional multipliers circuits are either, parallel [104 - 105] or array [106-107], sequential [108-110] in organization. In implementing the RR7SQSD field elements multiplication architecture, a 'parallel multiply-simultaneous addition-and-subtraction' strategy is employed to avoid the expensive polynomial division. This is done by simply writing the coefficients of the multiplicand b(t) in rows that terminate with corresponding coefficients of the multipliers a(t). The process repeated for all multiplier coefficients forming a 2-dimensional array of k-rows and r+1 - columns as shown in equation (3.50) on figure 3.36.

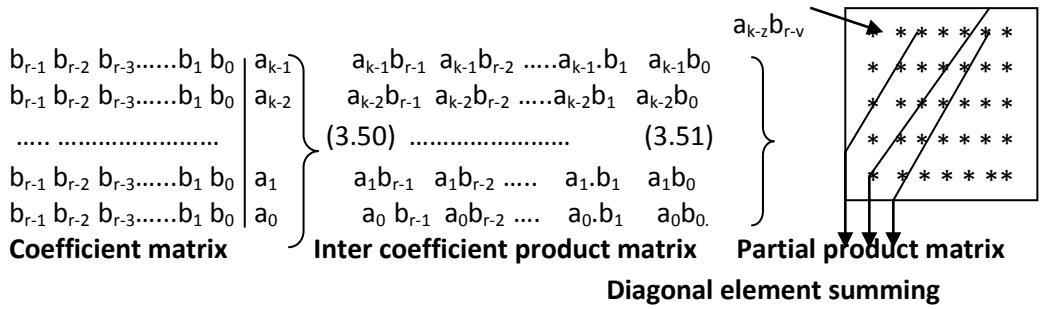


Fig.3.36 High speed $SGF(7^m)$ element multiplication operation organization

Next each multiplier-coefficient-column element $a_{k-z,}$ (z =1,2,…k) RR7SQSD-wise multiplies all its corresponding multiplicand-coefficient-row elements $\{b_{r-v}, \}_z$, v = (r,(r-1),(r-2),…,2,1).

Repeating this, for all the rows of equation (3.50) produces the second 2-dimensional partial product-coefficient matrix of equation (3.51). The interim – product coefficients p_{k+r-i}, are obtained by RR7SQSD XOR operation of the diagonal elements of the matrix in equation (3.46) from the left. The sum total is finally reduced modulo w(t) to obtain the product of *a(t) * b(t)* field multiplication operation product. The multiplier coefficient-column element positioning may start with the junior coefficient a_0 and end with the senior, a_{k-1}. In this case, the summation of diagonal elements starts from the right. However, in either case the same number of diagonal elements adds up to form any particular interim-product coefficient p_{k+r-i}, i.e.; $p_{k+r-i} = (\sum_{i=0} (a_{k-n}.b_{r-m} - 7p^{o}_{nm}))$

where *n = 0,1,2….k-1 and m=0,1,…r*-3 .(3.52)

Figure 3.37 shows the RR7SQSD element finite field parallel multiplication LSI circuit. In realizing this circuit the FEAM circuits are used either as field element multiplier or field element adders. The $SGF(7^m)$ element parallel addition circuits are used to implement the combinational field parallel addition block. A slide-and-repeated subtraction modulo reduction technique is employed in the design. This is accomplished by repeatedly RR7SQSD component-wise subtraction operation of the reduction modulo w(t), from the interim product coefficients in the power term of the independent variable *t*, of the interim product $IP(t)$ polynomial with gradual sliding of the reduction polynomial $w(t)$, to the right until $IP(t) < w(t)$. This unit realization is simply an arrangement of $n = k + r$ number of RR7SQSD field adders in a parallel $SGF(7^m)$ field element addition circuit. . The subtraction starts with the highest row with sum outputs

feed back into the third input such that the speed *T* of the operation depends greatly on the signal delay times of the adder t_{SD} and t_{MD}, *multiplier* units. *T* may thus be deduced using the expression $T = t_{mr} + t_{SD}((k-1)+u))$; where *u* is the number of subtractions, t_{mr} = t_{SD} = radix – 7 SD full adder/ multiplier delay period.

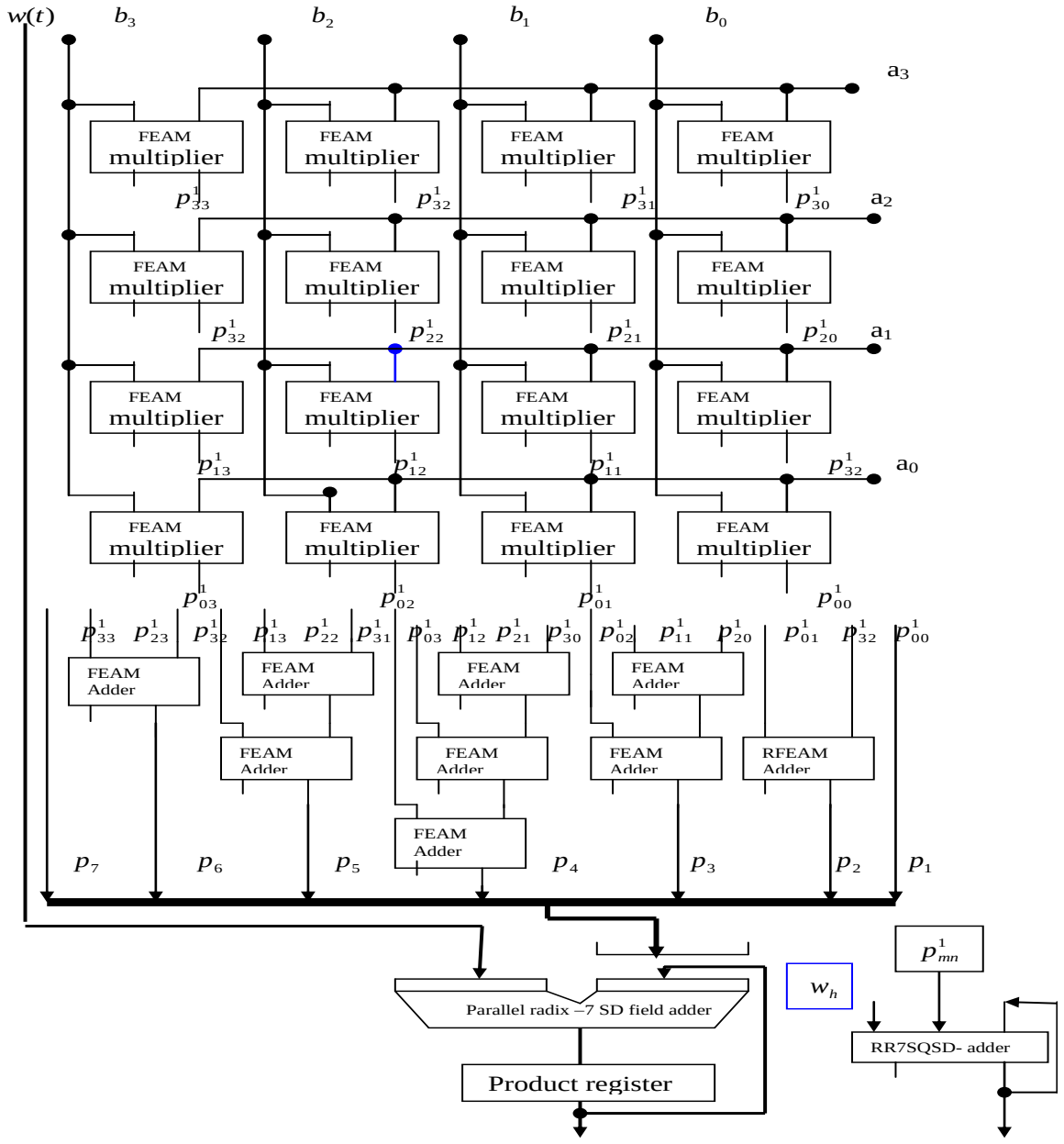


Fig.3.37. RR7SQSD element finite field parallel multiplication LSI circuit

3.2.3.10 RR7SQSD information transfer logic circuit

3.2.3.10.1 Multiplexer unit design

From the data flow diagram of figure 3.22 in section 3.5.1.2, the parallel inputs to the multiplexers are streams of RR7SQSD strings hence the circuit sought for can be considered as a data selector and be described by the functional diagram of figure 3.38 performing input data stream type selection and the transmission of the same to the output.

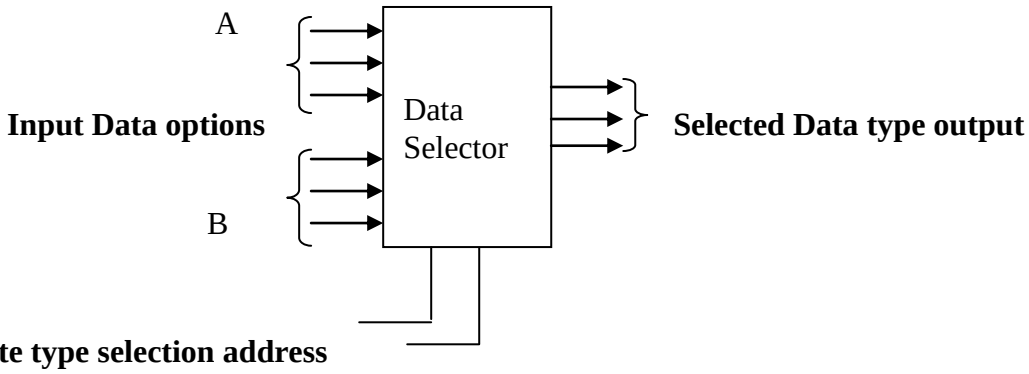


Figure 3.38: Functional block diagram of $SGF(7^m)$ multiplexer

An RR7SQSD multiplexer may be nomenclatured as a $7n-by-n$ multiplexer where n is the number of RR7SQSD consequently; the synthesis of a $21-to-3$ multiplexer may be described as follows. Let the RR7SQSD string input data type A, B and the selector output ψ be represented as

$$A=\alpha_{10},\alpha_{11},\alpha_{12},.....,\alpha_{1n-11},\ B=\beta_{20},\beta_{21},\beta_{22},.....,\beta_{2n-11},\ C=\psi_{30},\psi_{31},\psi_{32},.....,\psi_{3n-1}$$

Similarly let the input data stream and T-gate crystal selection addresses be

$x_0 \in \{\bar{3},\bar{2},\bar{1},0,1,2,3\}$ **and** $x_1 \in \{\bar{1},1\}$ **then the operation of the multiplexer can be described by equation (3.53)**

$$\psi = T(T(A;x_0),T(B;x_0);x_1)$$

$$= T(T(\alpha_{10},\alpha_{11},\alpha_{12},.....,\alpha_{1n-1};x_0),T(\beta_{20},\beta_{21},\beta_{22}.....,\beta_{2n-1});x_1) \quad \textbf{(3.53)}$$

And any particular output selection can be expressed in Boolean algebra as in equation (3.54)

$$\psi_{i,j} = x_1(x_0\alpha_{i,j} + x_0\beta_{i,j}) \qquad \textbf{(3.54)}$$

The RR7SQSD T-gate realized circuit of the multiplexer unit is as shown in fig. 3.39

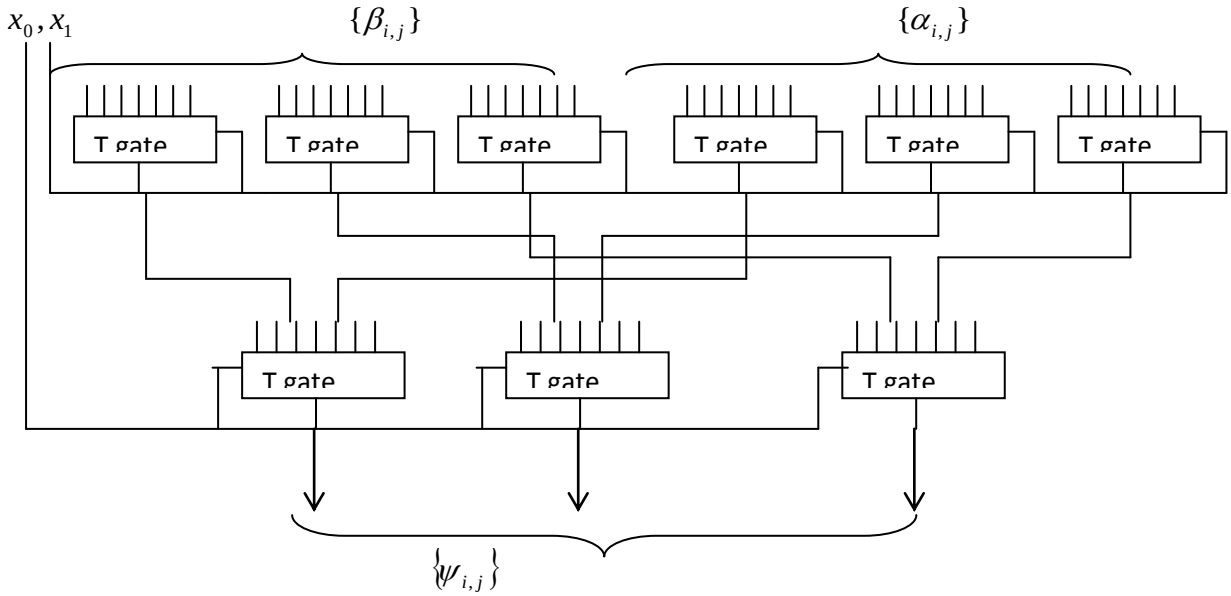
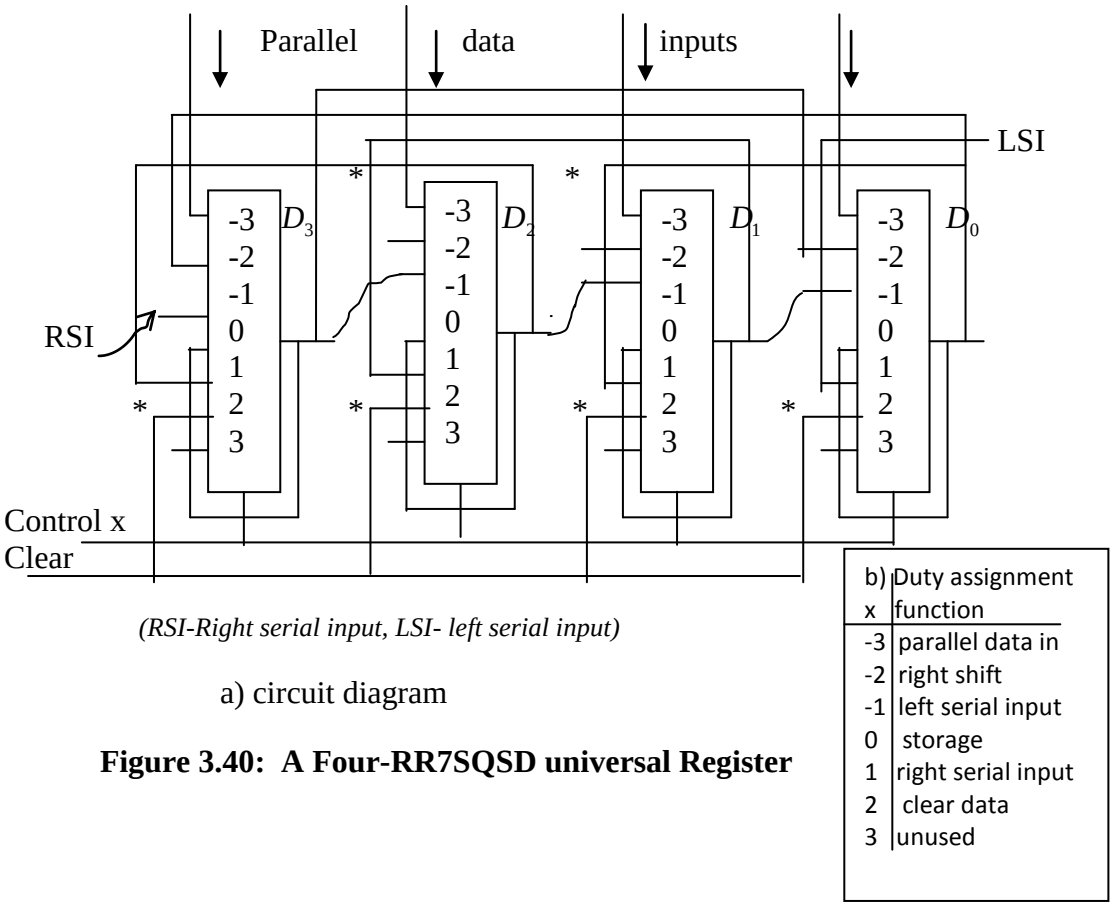


Figure 3.39: A $21-to-3$ **RR7SQSD T-gate based** $SGF(7^m)$ **multiplexer**

3.2.3.10.2 The RR7SQSD registers



(RSI-Right serial input, LSI- left serial input)

a) circuit diagram

Figure 3.40: A Four-RR7SQSD universal Register

The circuit diagram of a 4-RRSQSD left/right/cyclic shift register is as shown in figure 3.40a. The organization in principle is similar to the concept of designing a bilateral shift register using B master-slave Flip-Flap-Flop (FFF) presented in [39]. The T-gates are first wired to function as a D-type flip-flap-flop and then cascaded to form the register. The duty assignment of the control signal threshold values are as indicated in the duty assignment table of figure 3.40b.

3.2.3.11 RR7SQSD inverter

Figure 3.41 shows one-RR7SQSD sign inversion logic. Sign inversion occurs in gate G_1 With RR7SQSD data word $-3,-2,-1,0,1,2,3$ at the inputs of gate G_1, the data to be inverted is applied simultaneously to G_1, G_2. A "-1" gate control causes the RR7SQSD at the input to be inverted and sent to the output through gates G_1, G_2 and a "0" routes the input data through G_3, G_4 to G_5's output un-inverted.

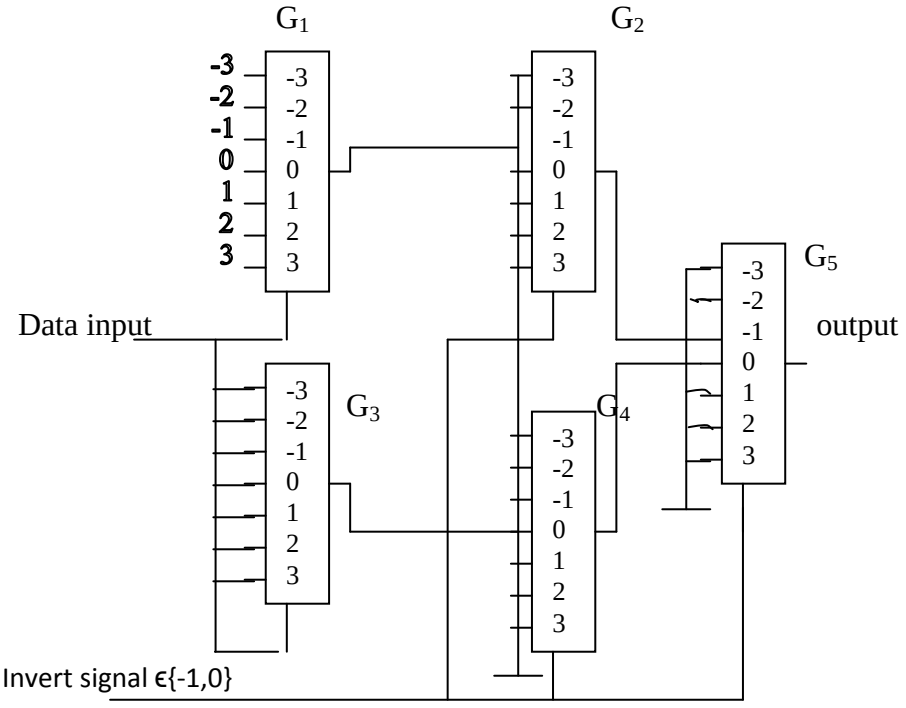


Figure 3.41: An RR7SQSD sign inverter.

3.2.3.12 Experiments on the designed RR7SQSD T-gate dual full adder/multiplier circuit

The operation of a full RR7SQSD T-gate based dual adder/multiplier operation circuit has been presented in section 3.1.2.1, equation (3.2) and later in section 3.6.1.2 fig 3.32. The goal of this experiment therefore is to confirm that figure 3.32(i)a will be capable of:

i) **adding two RR7SQSD α_i, β_i and the carry-in δ_{i-1} from the previous addition process together to produce a sum γ_i and carry out δ_i and**

ii) **Multiplying two RR7SQSD α_l, β_k together, reports a carry-out $\rho^o{}_{lk}$ to the next operation and adds the interim product digit λ_{lk} to a carry-in from previous operation digit δ_{lk-1} to obtain a final product digit $\rho^I{}_{lk}$ as described in equation (3.6.1.2).**

In the light of the above the device functional capability was experimented upon as follows:

3.2.3.12.1 Testing the RR7SQSD full Adder

Although the test specimen operands: $x_i = 02112000223\bar{1}2\bar{3}2\bar{1}012\bar{3}23$ **and** $y_i = 033022022\bar{1}3\bar{3}2120\bar{3}232\bar{3}02$ **are equally appropriate for a parallel addition mode however, the device was simulated on sequential addition mode as shown in figure 3. 42. The RR7SQSD pulse trains of the operands are applied simultaneously two first input terminals of the adder and resulting carry-out is fed into the third input terminal along with the next input RR7SQSD pulse train. Thus the i[th] sum $s_i = x_i + y_i + \delta_{i-1}$ digit and the corresponding carry-out δ_i , are available at the adder's output terminals at the extinct of each pulse.**

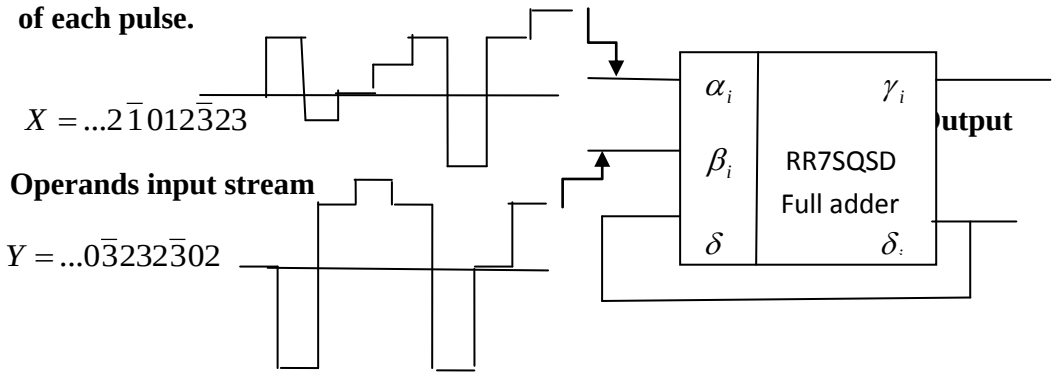


Figure 3.42 Testing the RR7SQSD full adder circuit for sequential addition mode

3.2.3.12.2 Testing the RR7SQSD quasi- multiplication ability

As earlier stated the different between an RR7SQSD fuller adder and a multiplier is in the voltage potentials sitting at the various RR7SQSD T-gate inputs, right at the point of fabrication. External input operands do no determine type of operation (addition or multiplication) thus figure 3.42 configuration can adequately satisfy a quasi-RR7SQSD multiplication operation provided the device is wired for multiplication. That is, what is done here is component-wise multiplication with addition of reported carry at each pulse train. The input operands used for RR7SQSD multiplication experiment are: $x = 32\bar{3}210\bar{1}2\bar{3}2\bar{1}\bar{3}2200021120$ **and** $y = 20\bar{3}232\bar{3}0212\bar{3}3\bar{1}2\bar{2}022033$

Both experiments were carried out by modelling the device functional capability using QB 4.5 hardware-description-language styled source code programs as presented in appendices CH 3.8 for the addition mode and CH 3.9 for the multiplication mode. The results are shall be presented and discussed in chapter 4.

CHAPTER 4

RESULTS AND DISCUSSION

The raw results, as presented in Appendices CH4.1 to CH4.4 and Appendices CH4.5 to CH4.8, comprise:

i) **Application of RR7SQSD arithmetic for computing product of operands with lengths by far greater than the computing systems word length.**

ii) **Computing** $SGF(17011)$ **element multiplicative inverses using the extended Euclidean algorithm.**

iii) **Method of embedding plain text on the points of The EC** $y^2 = x^3 - 53x + 218 \bmod 17011$

iv) **Computing elliptic curve point multiplication using the one-stop multiply-by-7 addition/subtraction chain algorithm.**

v) **Art of message encryption and decryption on EC defined over RR7SQSD finite fields.**

vi) **Modelling the electrical parameters of the RR7SQSD CP-gate derived T-gate.**

vii) **Simulation of the correct operation of the T-gate**

viii) **Simulating Dual RR7SQSD full adder/multiplier unit on T-gate level**

It is observed that the volume of raw results generated in each of these experiments is quite enormous particularly for cases (i) to (v). However, it is also worthy of note that only exact number of character digits of a given code word makes meaning in cryptography. Similarly, it is an exact length without approximation of a given cipher that can present the true

information at a deciphering stage. Thus there is the tendency to present a large volume and intimidating report with the appendices occupying a greater percentage. To avoid this problem, only the first three and the last three pages plus certain particular nodes of interest are presented in the appendices.

4.1 RR7SQSD big integer multiplication

This all-very important arithmetic operation in image processing and cryptography when executed in general purpose computing systems most often is berserk with result truncation or presentation in scientific notations, either of which is not cryptography compliance. This simulation experiment was conducted in section 3.2.1.2 to show that RR7SQSD arithmetic can be used to solve this problem. The assembled program of the experiment based on algorithm AM2 of figure 3.12 automatically converts input operands of any length to the RR7SQSD form and accumulates the multiples of the multiplicand based on the RR7SQSD scanned. Appendix CH 4.1 shows the raw results of the multiplication operation for operand lengths: 14 by 14 , 32 by 32, 88 by 88 and 160 by 160 decimal digits (limit of Basic language environment). Summary of the program output is presented in figure 4.1 for discussion. In specimen i) operands word length is within the systems data word length limit hence the operation is not subject to round-up error or truncation however, in specimens ii), iii) and iv) operands lengths are by far longer than the processor data word length. Executing the 160 by 160 decimal digit multiplication took less than 1 second in the Pentium IV processor. The experiment proved that multiplication of integer operands of any length can be executed in a non-expensive general purpose computing system truncation of results.

i) $(7^{16}+74)x(7^{16}+74) \equiv (3323293.5696775)x(3323293.5696775)$

$= 11044276742488391200296056 25$

ii) $(7^{37} = 1856,2115,9210,1757,4302,4531,6367,1207)$ by

$(7^{37} = 18562115,9210,1757,4302,4531,6367,1207)$ (32 x 32 digits)

$= 344552147465294110719732986332367243247925798357729806000836849$

iii)
1856,2115,9210,1757,4302,4531,6367,1207,7895,2313,7945,8679,4030,4589,6354,4756,8523,9546,8671,7477,7532,1596 (88 decimal digit) by

1856,2115,9210,1757,4302,4531,6367,1207,7895,2313,7945,8679,4030458963544756852395468671 7417 7532 1596 (88 decimal digits)

=
344,5521,4746,5294,1107,1973,2986,3323,9655,3687,9435,5186,7658,4286,6565,2934,3207,3506,8585,9607,0473,0306,0181,6756,6776,9375,8457,1450,2068,9394,9751,1918,8714,0811,9348,5623,9305,3194,3585,6606,5560,8492,2398,7216

iv) 1856 2115 9210 1757 4302 4531 6367 1207 7895 2313 7945 8679 4030 4589 6354 4756 8523 9546 8671 7417 7532 1596 7894 4562 3216 4756 2346 7894 4562 7417 3216 1596 7895 4896 5643 1459 2587 3694 1002 7777 (160 digits) X

1856 2115 9210 1757 4302 4531 6367 1207 7895 2313 7945 8679 4030 4589 6354 4756 8523 9546 8671 7417 7532 1596 7894 4562 3216 4756 2346 7894 4562 7417 3216 1596 7895 4896 5643 1459 2587 3694 1002 7777 (160 digits)

=

344 5521 4746 5294 1107 1973 29863323 9655 3687 9435 5183 7628 4286 6565 2394 3207 8585 9687 0475 0306 3112 4319 0206 6212 5169 21470 9578 9340 2381 4333 3222 2442 1109 5351 1824 9035 1320 7474 1145 8850 2040 8256 0203 6672 1000 6553 9676 9795 6445 5619 8679 6448 7754 4747 9881 7224 4197 4580 4089 6943 3712 1435 0234 1419 5569 2433 4493 4172 7302 1661 0250 3598 1966 8034 6820 2039 1156 1729.

Figure 4.1 Summarized result of the RR7SQSD big integer operand multiplication .

4.2 Results of the experiments to test the hardware implement friendly algorithm

4.2.1 Results of $SGF(7^2)$ construction

The elements of this field may be viewed as the elements of the restricted radix - 49 symmetrical radix - 25 signed digit number system. Figure 4.2 shows the *21* degree - *2 monic irreducible polynomials* in their coefficient value form and the corresponding finite field elements, in vector form of the field SGF(7^2), using the irreducible polynomial $x^2 - 3x - 2$. The following numerical example certifies the accuracy of the computation

For example, $\mu^{23} * \mu^{32} = \mu^{55 \bmod 48} = \mu^7$ **and** $\mu^{23} * \mu^{32}$ **is also** $= 2*(3\mu - 2) = -\mu + 3 = \mu^7$**.**

Similarly, $\mu^{27} + \mu^{19} = (3\mu + 1) + (2\mu + 3) = \overline{2}\,\mu + \overline{3} = \mu^{43}$**.**

Monic degree 2 irreducible polynomials for sign digit element *characteristic 7* finite field

1 –1 -3	1 0 -3	1 1 -3	**1 –3 -2**	1 –2 -2	1 2 -2	1 3 -2
1 –3 -1	1 –1 -1	1 1 -1	1 3 -1	1 –3 1	1 0 1	1 3 1
1 –2 2	1 0 2	1 2 2	1 -2 3	1 -1 3	1 1 3	1 2 3

Elements of SGF (7) in their vector form of representation using the basis {1, μ)

(0 1)	(1 0)	(3 2)	(-3 -1)	(-3 1)	(-1 1)	(-2 -2)	(-1 3)	(0 -2)
(-2 0)	(1 3)	(-1 2)	(-1 -2)	(2 -2)	(-3 -3)	(2 1)	(0 -3)	(-3 0)
(-2 1)	(2 3)	(2 -3)	(3 -3)	(-1 -1)	(3 - 2)	(0 -1)	(-1 0)	(-3 -2)
(3 1)	(3 -1)	(1 -1)	(2 2)	(1 -3)	(0 2)	(2 0)	(-1 -3)	(1 -2)
(1 2)	(-2 2)	(3 3)	(-2 -1)	(0 3)	(3 0)	(2 -1)	(-2 -3)	(-2 3)
(-3 3)	(1 1)	(-3 2)	(0 1)					

Figure 4.2 Irreducible polynomials and field elements of the signed digit $SGF(7^2)$

4.2.2 Results of the multiplicative inverse computation

Field element multiplicative inverse computation as discussed in section 3.1.1.6 is a complex yet indispensable operation in all finite field arithmetic operations. With the EEA in section 3.1.2.3(e) as an efficient and hardware implement friendly method of computing multiplicative inverse of field elements, the experiment in section 3.2.2.2 was set to ascertain the accuracy of EEA computation even with the special primes from the number sequence $H \in \{7^m + \alpha\beta i\}$ **and in RR7SQSD element finite fields as discussed in section 3.2.1.1 axiom 3.9. The experiment was conducted in two areas. The EEA was used to compute (manually) the inverse elements of** $SGF(7^2)$ **in the RR7SQSD domain. Then, a QB 4.5 source code program as shown appendix programs CH3.5 was assembled to compute specimen data set** $H \in \{37,79,241,17011,117721,.....\}$ **in the decimal number system domain.**

4.2.2.1 The inverse element computation of $SGF(7^2)$

The result of inverse element computation for $SGF(7^2)$ is as shown in figure 4.3.

S/N0	$b(x)$	$b(x)^{-1} \in SGF(7^2)$
1	1	1
2	x	$3x^2+2x+3$
3	3x+2	$-3x^2+3x$
4	-3x-1	$-2x^2+3x$
5	-3x+1	$-2x+3$
6	-x+1	$-2x^2-3x+1$
7	-2x-2	$2x^2+3x+2$
8	-x+3	$-3x^2-2x-1$
9	-2	$-3x^2+2x+2$
10	-2x	$3x^2+3x$
11	X+3	$-3x^2-2x+2$
12	-x+2	$2x^2-x-2$
13	-x-2	$x+2$
14	2x-2	$2x^2+2x+1$
15	-3x-3	$-3x^2+x+3$
16	2x+1	$3x^2+1$
17	-3	$-2x^2-x-1$
18	-3x	$3x^2-x-2$
19	-2x+1	$-3x^2-3x+1$
20	2x+3	$2x^2+2x+2$
21	2x-3	3x-1
22	3x-3	$-3x^2-2x$
23	-x-1	$-3x^2-x-3$
24	3x-2	$-x$
25	-1	x^2-3x-2
26	-x	$3x^2+x-1$
27	-3x-2	$-3x^2+x-2$
28	3x+1	$-3x^2-2x+3$
29	3x-1	$2x^2+3x$
30	x-1	$-2x^2+x$
31	2x+2	$2x^2-x-3$
32	x-3	$-2x^2+3x-3$
33	2	$-3x^2+2x+3$
34	2x	$3x^2+2$
35	-x-3	$3x^2+2x-2$
36	x-2	$2x^2+3x+1$
37	x+2	$2x^2+1$
38	-2x+2	$-2x^2-2x-1$
39	3x+3	$-2x+1$
40	-2x-1	$3x^2+3x+1$
41	3	$2x^2+x+1$
42	3x	$3x^2-3x-3$
43	2x-1	$-3x^2-3$
44	-2x-3	$-2x^2-2x-2$
45	-2x+3	$-3x+1$
46	-3x+3	x^2+x-3
47	x+1	$-2x^2+2x-1$
48	-3x+2	$3x^2-x+1$

Figure 4.3: $SGF(7^2)$ **multiplicative inverses using the irreducible polynomial** $w(x)=x^2-3x-2$

4.2.2.2 Indirect RR7SQSD domain mode

The goal of this experiment as earlier stated in section 3.2.2.2(ii) is to consolidate the fact that EEA based inverse calculation can be accurately executed for any prime modulo in the number sequence

$H \in \{7^m + 12j\}$ **where** $m, j \in \{1,2,3,..\}$ **. The raw results for** $H \in \{37,79,241,691\}$ **and fragments of** $H = 17011$ **are presented in appendix CH 4.2. The entry in any even column represents the multiplicative inverse of the corresponding odd column entry. The accuracy of the results can be easily verified using the fact that** $b(x) * b(x)^{-1} \bmod p = 1$ **for values of** $x \in p$ **as shown in fig. 4.4 for some random values of x in prime moduli 37,79,241, 691 and 17011.**

Prime modulo *P*	*b(x)*	$b^{-1}(x)$	$b(x).b^{-1}(x) \bmod P$
37	8	14	$(8*14) \bmod 37 = 1$
	23	29	$(23*29) \bmod 37 = 1$
	30	21	$(30*21) \bmod 37 = 1$
79	11	36	$(11*30) \bmod 79 =$
	39	77	$(39*77) \bmod 79 = 1$
	64	21	$(64*21) \bmod 79 = 1$
241	75	45	$(75*45) \bmod 241 = 1$
	145	123	$(145*123) \bmod 241 = 1$
	211	8	$(211*8) \bmod 241 = 1$
691	53	339	$(53*339) \bmod 691 = 1$
	301	427	$(301*427) \bmod 691 = 1$
	599	353	$(599*353) \bmod 691 = 1$
17011	9506	13602	$(9506*13602) \bmod 17011 = 1$
	14636	6919	$(14636*6919) \bmod 17011 = 1$
	16995	11695	$(16995*11695) \bmod 17011 = 1$

Figure 4.4: Checking accuracy of RR7SQSD multiplicative inverse computation

It is seen that the result is accurate as the product of $b(x)*b(x)^{-1}=1$ in all values of x. The experiment thus confirmed that the RR7SQSDNS supports multiplicative inverse computation.

4.3 Embedding plain text message on elliptic curves defined over RR7SQSD finite fields experiment.

The experiment generates the quadratic residues, computes the elliptic curve group rational points and embeds the given message points into this group of rational points. The experiment, using the established procedure described in section 3.5.4 codes the message in the given alphabets, generated 21223 **rational points of the elliptic curve group** $E_{17011}(-53,218)$ **to embed the coded message. To accomplish this process the program using the Legendre symbol identified the set of quadratic residues for the prime modulo** 17011 **and with Fermat's little theorem computed the corresponding square roots that constitutes the rational points. The QB 4.5 source code of the experiment is as presented in appendix program CH 3.4. The raw result of the experiment showing the 21223 rational points and the embedded message is as presented in Appendix output CH 4.3. A summary of the result in part I, II and III is presented in figure 4.5. Part I shows the encoded message. Part II shows a random sample of the generated curve rational point and part III shows the embedded message. All the** 22 **blocks of the specimen plain text message i.e. "The quick brown fox jumps over the lazy dog "in decimal numeric coded were fully embedded in to the rational group of points of the elliptic curve** $E_{17011}(-53,218)$**.**

```
              THE PLAN TEXT MESSAGE TO BE SENT IS

      'THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG'

           THE CORRESPONDING NUMERIC CODE IS    (Part I)

1507          2704          2625          1201          527           2006
1827          2205          2715          1619          2113          2710
1524          2706          2314          1815          2702          311
2109          2717           605            20

    AND THE ELLIPTIC CURVE CONSTANTS ARE-53  218  17011

              WHILE THE RATIONAL CURVE POINTS ARE  (Part II)

 2  4835        4 16541        7  3799        8  2632       10  9691
12 10031       13 16051       15 12516       16 12088       18 13186
19  9968       21  8173       22  7965       24 12986       25 12332
27 12618       29  4603       30 11849       33  5600       37  3030
42  1188       43 14253       44 10474       45 15296       46  4691
47  9130       55  7007       56   143       58 13948       59  5048
60  4832       62 10629       63   841       66  8655       67 15587

1490 14972     1491   387     1494 13071     1495  5549     1498  1641
1507 16417     1509  4862     1510 13380     1514 13193     1515 10031
1517  2462     1519  8714     1525  4099     1529  8000     1530 11752
1532 15911     1537  8718     1540   513     1543  2573     1544 15455

16936  7524    16938  5608    16939 10327    16943 14764    16945 14621
16946   126    16948  6523    16953  4931    16954  3916    16955   816
16958  5004    16960 10592    16961  1553    16967  3442    16976  2660
16979  9335    16980 11794    16982   772    16983 11642    16985  2714
16987 16091    16989 12166    16990   792    16996  3300    17001  7136
17003 16350    17004 14102    17005 15111    17008   620    17010 14564

              THE ORDER OF FIELD =     21223
```

(Part III)

THE PLAN NUMERIC TEXT EMBEDDED INTO THE ELLPTIC CURVE GROUP OF POINTS

S/N0	RAW MESSAGE	CURVE POINTS EMBEDDED	Reference values(4^{th},column)
1	1507	1507 16417	
2	2704	2704 393	
3	2625	2628 2950	
4	1201	1204 15324	
5	527	528 12647	
6	2006	2006 9629	
7	1827	1828 1936	
8	2205	2205 14215	
9	2715	2715 13753	
10	1619	1619 915	
11	2113	2113 10358	
12	2710	2710 401	
13	1524	1525 4099	
14	2706	2708 11651	
15	2314	2314 12327	
16	1815	1816 6343	
17	2702	2702 11279	
18	311	311 4188	
19	2109	2109 15967	
20	2717	2718 3716	
21	605	610 3309	
22	20	21 8173	

s/n	text	code	(x, y)
1	OG	1507	(1507, 16417)
2	SpD	2704	(2704, 393)
3	ZY	2625	(2628, 2950)
4	LA	1201	(1204, 15324)
5	Esp	0527	(528, 12647)
6	TH	2006	(2006, 9629)
7	Rsp	1827	(1828, 1936)
8	VE	2205	(2205, 14215)
9	SpO	2715	(2715, 13735)
10	PS	1617	(1619, 915)
11	UM	2113	(2113, 10350)
12	SpJ	2710	(2710, 401)
13	OX	1524	(1525, 4095)
14	SpF	2706	(2708, 11651)
15	WN	2314	(2314, 12327)
16	RO	1815	(1816, 6343)
17	SpB	2702	(2702, 11279)
18	CK	0311	(0311, 4188)
19	UI	2109	(2109, 15967)
20	SpQ	2717	(2718, 3716)
21	HE	0605	(610, 3309)
22	T	0020	(0021,08175)

Figure 4.5 Summarized result of the $E_{17011}(-53,218)$ message embedment

The box on the right contains the reference data manually computed with the embedded points in the 4^{th}, column. The simulation outputs coincide with the reference data. In both cases it is also observed as experienced in the manual checking, 12 message code points such as 20,605 and 1815 are not direct curve points and have to be stream lined i.e. 21,610 and 1816 into points of the EC.

4.4 Summarized result of the one-stop multiply-by-7 EC point multiplication $W = kP$, experiment

The purpose of the experiment is to test the accuracy of the multiply-by-7 EC point multiplication algorithm as presented in figure 3.15 described in section 3.4.3. The result presented consists of two parts: the simulation output and the order of computation complexity analysis of the algorithm. The inferences from the two parts simply indicate that the algorithm is accurate and efficient.

4.4.1 Simulation output

The QB 4.5 source code program is as presented in appendix CH 3.6 and the corresponding raw output is as presented in appendix 4.4. The traditional method of computing point multiplication that is $kP = P + P + P + ... + P$ **was incorporated as an autonomous reference module for instant comparison of the computed results. The reference module used equation (3.34) for the computation. About 21 different values of** k **to match** $19 \in \{E_{17011}(-53,218)\}$ **curve points as shown in Appendix Output CH 4.4 were used as specimen test data in this experiment. Although the QB 4.5 compiler could not permit the use of near practical sizes of k to enable determine meaning full speed recording, the various combinations of** k **and** $p(x, y)$ **as input operands all returned very accurate results. Table 4.1 shows in part, the points of the computation for when** $k = 3414$**. The algorithm was subjected to a large number of test data but the purpose of clarity, 15 output nodes of the 21 values of**

$k \in \{29,156,375,1379,2340,2983,3399,3408,3414,1289,873,6085,2713,619,1191\}$ **for the curve point** $P(1,14564)$ **are reproduced in table 4.2. It can be seen that all computed**

values fully coincide with the corresponding reference program values. This shows that the algorithm can be used to comfortably compute point multiplication on elliptic curves defined over RR7SQSD element finite fields.

Table 4.1 Points of the elliptic group $E_q(a,b) = E_{17011}(-53,218)$

k	$P(x,y)$	k	$P(x,y)$	k	$P(x,y)$	k	$P(x,y)$
1	(1, 14564)			….	……	….	……
2	(15649, 10379)			….	……	….	……
3	(13200, 6763)	.	.	408	(13767,16437)	1510	(15050, 16697)
4	(14518, 16806)	.	.	409	(977,10989)		
5	(7344, 5815)	…	……..	….	……		
6	(11612, 10163)	….	……..	….	……	1512	(1617, 16470)
7	(2941, 1752)	….	…….	….	……	….	……
8	(14555, 11667)	59	(13987, 16567).	….	……	….	……
9	(14085, 13514)	.60	(3436, 3266)	425	(2093,12621)	….	……
10	(12998, 8269)*	61	(1456, 14033)	426	(16076, 13970)	….	……
11	(11434, 3219)	62	(6065, 2275)	427	(9723, 12634)	2456	(3109, 12190)
12	(9652, 1119)	63	(11571, 2044)	428	(12311, 13510)	2457	(8323, 14060
13	(10461, 4649)	64	(68, 11841)	429	(8479, 9817)	….	……
14	(9312, 4353)	65	(12667, 13995)	430	(4441, 10796)	….	……
15	(2104, 13362)	….	……	…..	……	2983	(227,7210)
16	(12674, 11834)	68	(10817,13328)	….	……	….	……
17	(3594, 1974)	69	(5142,11411)*	485	(6407,16992)	….	……
18	(2229, 1784)	70	(10591,12081)	486	(3929,15851)*		

Table 4.1 Points of the elliptic group $E_q(a,b)$ = $E_{17011}(-53,218)$ (continued)

19	(9209, 13394)	….	……	487	(2549,3083)		
20	(12477,10210)	….	……	….	…….		
21	(16851, 6472)	99	(3377,397)				
22	(16035, 6472)	100	(809, 10554)				
23	(15650, 10141)	101	(2281, 8878)				
24	(2689,2111)	102	(14669, 6139)	….	…….		
25	(8334,784)	103	(13047,1857)				
26	(14325, 13210)	….	……			….	…….
27	(10758, 14373)	….	……				
28	(1367, 7829)	298	(8840,1480)	….	…….		
29	(3788, 10521)					….	…….
30	(7726, 14993)	300	(16505, 2930)	….	…….	….	…….
31	(9970, 7654)					3398	(11069,13047)
32	(3680, 2702)			823	(6663, 16229)	3399	(12674,5172)*
						3400	(1337, 92)
				829	(4812, 16961)		
				….	…….		
				….	…….	3414	(1,2447)

Table 4.2 One-stop Multiply-by-7 Simulation Algorithm

SCALAR k		POINTS				
Dec	**RR7 SQSD**	**COORDINATES OF THE POINTS OBTAINED FROM THE SIMULATION PROGRAM USING THE ONE-STOP MULTIPLY-BY-7 ALGORITHM**				
29	131	$1P$ $(1,14564)$	$(10P)$ $(12998,8269)$	$29P$ $(3788,10521)$	----------	----------
156	312	$3P$ $(13200,06763)$	$22P$ $(16035,6472)$	$156P$ $(14475,13287$	----------	----------
375	$11\bar{2}\bar{3}$	$1P$ $(1,14564)$	$8P$ $(14555,11667)$	$54P$ $(1573,9892)$	$375P$ $(2950,1521)$	----------
1379	$1\bar{3}010$	$1P$ $(1,14564)$	$4P$ $(14518,16806)$	$28P$ $(1367,7829)$	$197P$ $(2116,2787)$	$1379P$ $(6624,12964)$
2340	$10\bar{1}\bar{2}2$	$1P$ $(1,14564)$	$7P$ $(2941,1752)$	$48P$ $(4261,16193)$	$334P$ $(1823,4794)$	$2340P$ $(8419,13367)$
2983	$12\bar{2}\bar{1}1$	$1P$ $(1,14564)$	$9P$ $(14085,13514)$	$61P$ $(1456,14033)$	$426P$ $(16076,13970)$	$2983P$ $(227,7210)$
3399	$13\bar{1}3\bar{3}$	$1P$ $(1,14564)$	$10P$ $(12998,8269)$	$69P$ $(5142,11411)$	$486P$ $(13929,15889)$	$3399P$ $(12674,5177)$
3408	$130\bar{3}\bar{1}$	$1P$ $(1,14564)$	$10P$ $(12998,8269)$	$70P$ $(8796,6309)$	$487P$ $(7105,3192)$	$3408P$ $(2941,15259)$
3414	$130\bar{2}\bar{2}$	$1P$ $(1,14564)$	$10P$ $(12998,8269)$	$70P$ $(8796,6309)$	$488P$ $(5750,5025)$	$3414P$ $(1,2447)$
1289	$1\bar{3}\bar{2}21$	$1P$ $(1,14564)$	$4P$ $(12998,8269)$	$26P$ $(1692,2470)$	$184P$ $(9428,12144)$	$1289P$ $(2360,5160)$
873	$3\bar{3}\bar{1}\bar{2}$	$3P$ $(3879,14802)$	$18P$ $(7621,12502)$	$125P$ $(14738,29)$	$873P$ $14025,9684)$	----------

Table 4.2 One-stop Multiply-by-7 Simulation Algorithm (contuned)

6085	$3\bar{3}\bar{2}12$	$3P$ $(3879,14802)$	$18P$ $(7621,12502)$	$124P$ $(6209,7547)$	$869P$ $(4390,10292)$	$6085P$ $(5194,15749)$
2713	$11\bar{1}3\bar{3}$	$1P$ $(1,14564)$	$8P$ $(14555,11667)$	$55P$ $(4258,10914)$	$388P$ $(8846,8783)$	$2713P$ $(12276,11096)$
619	$2\bar{1}\bar{3}3$	2P $(3004,11971)$	$13P$ $(15817,3948)$	$88P$ $(9808,6045)$	$619P$ $(14735,8922)$	----------
1191	3332	$3P$ $(3879,14802)$	$8P$ $(7621,12502)$	$24P$ $(8789,6539)$	$1191P$ $(9471,6243)$	----------

4.4.2 The order of complexity of point multiplication on EC defined over RR7SQSD finite fields using the one-stop multiply-by-7 algorithm

The operation execution complexity analysis was carried out on the multiply-by-7 algorithm by computing the order of complexity of all the variables. Table 4.3 shows the summary of the computation

Table 4.3 Summary of the order complexity for the One-stop multiply-by-7 algorithm

Arithmetic Functions	Select Module			Main module						Total
	$1P_{sel}$	$2P_{sel}$	$3P_{sel}$	$2P$	$3P$	$4P$	$7P$	*Update*	$x_w y_w$	
A	-	5	4	5	6	4	5	6	2	37
M	-	4	7	4	8	8	10	13	6	60
S	-	3	4	4	2	1	3	5	2	24
I	-	-	-	-	-	-	-	-	2	2
Total	-	12	15	13	16	13	18	24	12	123

It can be seen from this summary that, whereas in the selection module:

i) 5 addition, 4 multiplication and 3 squaring are required to form $2P_{sel}$

ii) 4 addition, 7 multiplications and 4 squaring are required for the $3P_{sel}$.

Similar, the main module required 28 additions, 49 multiplications, 17 squaring and 2 inversion operations to arrive at a final result. This gives a total of 37 additions, 60 multiplications, 24 squaring and 2 inversions or a grand total of 123 arithmetic operations as against the 137 in section 3.3.3.2, which means an improvement of about 11 % in speed.

4.5 Message encryption/decryption procedure experiment

The QB 4.5 source code and the raw results of the simulation obtained are as presented in Appendices CH 3.7 and CH4.5 respectively. The main objective of this simulation experiment is to ascertain the non-existence of message /data repudiation using the multiply-by-7 algorithm on elliptic curves defined over SGF (7^m). The results terminate with a tabular output to facilitate comparative analysis. The output of the experiment shows the coded message blocks as input data, the embedded message, and its encryption at the sending end and the decryption at the receiving end as the ultimate output. This portion is reproduced in Table 4.4. It can be seen that the input and output information coincide. Thus confirming the non- repudiation of the transmitted message and that a message encrypted using RR7SQSD field principles can be fully recovered.

Table 4.4 Example of encryption and decryption of message

S/N0	Input symbols	Input Numeric Code	Plain text Code	Cipher text Code	Deciphered text Code	Output message symbols
1	OG	1507	1507,16417	7842,2346	1507,16417	OG
2	SpD	2704	2704,0393	11750,16	2704,0393	SpD
3	ZY	2625	2628,2950	1608,12565	2628,2950	ZY
4	LA	1201	1204,15324	12555,9185	1204,15324	LA
5	Esp	0527	0528,12647	6926,12655	0528,12647	Esp
6	TH	2006	2006,9629	13202,9812	2006,9629	TH
7	RSp	1827	1828,1936	8918,7380	1828,1936	RSp
8	VE	2205	2205,14215	1253,4343	2205,14215	VE
9	SpO	2715	2715,13735	13617,9536	2715,13735	SpO
10	PS	1619	1619,0915	14606,2515	1619,0915	PS
11	UM	2113	2113,10358	954,8446	2113,10358	UM
12	SpJ	2710	2710,0401	9041,12792	2710,0401	SpJ
13	0X	1524	1525,4099	12250,259	1525, 4099	0X
14	SpF	2706	2708,11651	10761,42	2708,11651	SpF
15	WN	2314	2314,12327	8380,10376	2314,12327	WN
16	RO	1815	1816,6343	3664,11782	1816,6343	RO
17	SpB	2702	2702,11279	11786,16088	2702,11279	SpB
18	CK	0311	0311,4188	4520,7286	0311,4188	CK
19	UI	2109	2109,15907	12668,11909	2109,15967	UI
20	SpQ	2717	2718,3716	7545,8423	2718,3716	SpQ
21	HE	0605	0610 3309	1926,8348	0610,3309	HE
22	T	0020	0021 08173	13933,2298	0021,8173	T

4.6 The results of the experiment on the RR7SQSD T-gate

4.6.1 Electrical parameters

The main objective of this experiment is to confirm that this MVL circuit basic building block fully conforms to the design constrains i.e. it is signal voltage loss, oscillations, distortion - free and functions as a multiplexer circuit during operation. The circuit diagram of figure 3.25 as explained in section 3.2.3.7.1 was analogue simulated on transistor level using Multisim Electronic workbench that automatically converts all readings into a graphical outputs as presented in figures 4.6 to 4.10. The plots for transfer impedance and transfer function are as shown in figure 4.6 portraits the signal voltage loss-free nature. The next eight plots presented in figure 4.7 are the results of the transient analysis by tracking the $-3, -2, -1, 0, 1, 2$ **and** 3 **information input signals from the input to the output of the T-gate. The** 8^{th} **plot on this row is the case of no information signal input. The plots figures 4.8 and 4.9 are the results of the frequency and distortion tests. The last seven plots in figure 4.10 are the measurements of the time it takes an input signal to travel from any of the 7 input points to the output of the T-gate. It is observed that at any of the inputs, the threshold logic value voltage instantly rises to its nominal value (the y-axis) at less than zero microseconds. Frequency drifts and output signal distortion are practically absent as the output signal remains at** 0^{o} **phase as frequency increases from zero Hertz to GigaHertz. Transients are non-existent except at no load (the case of no input signal) condition. It is seen that this was as a result of transistor type miss-match. The noise level stood at 9.44p db at all inputs and 994.44 m db at the output.**

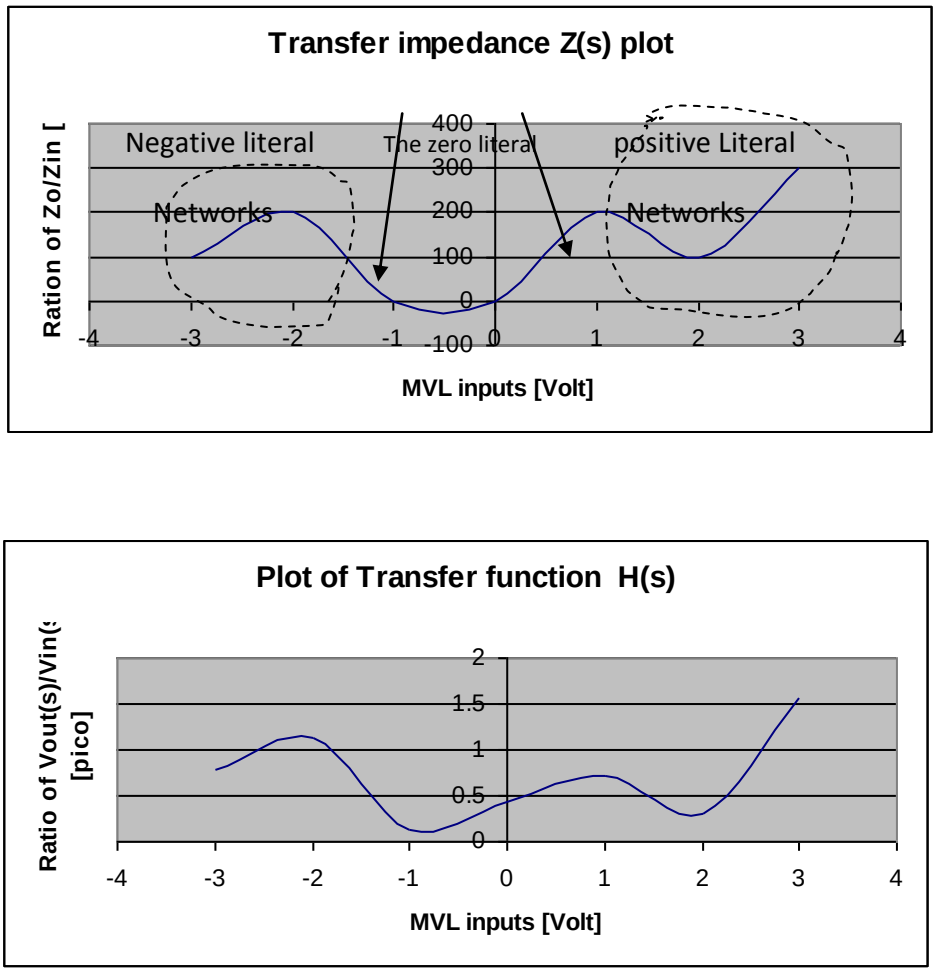
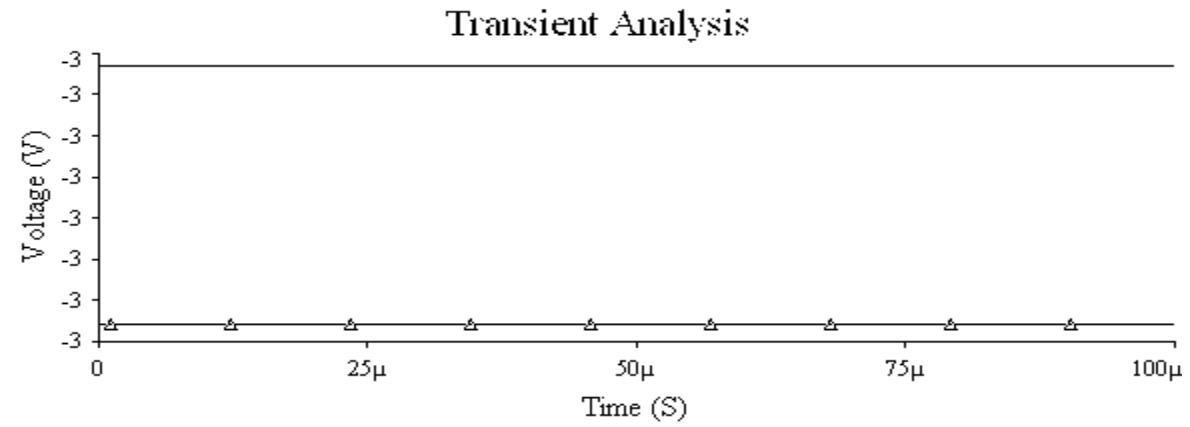
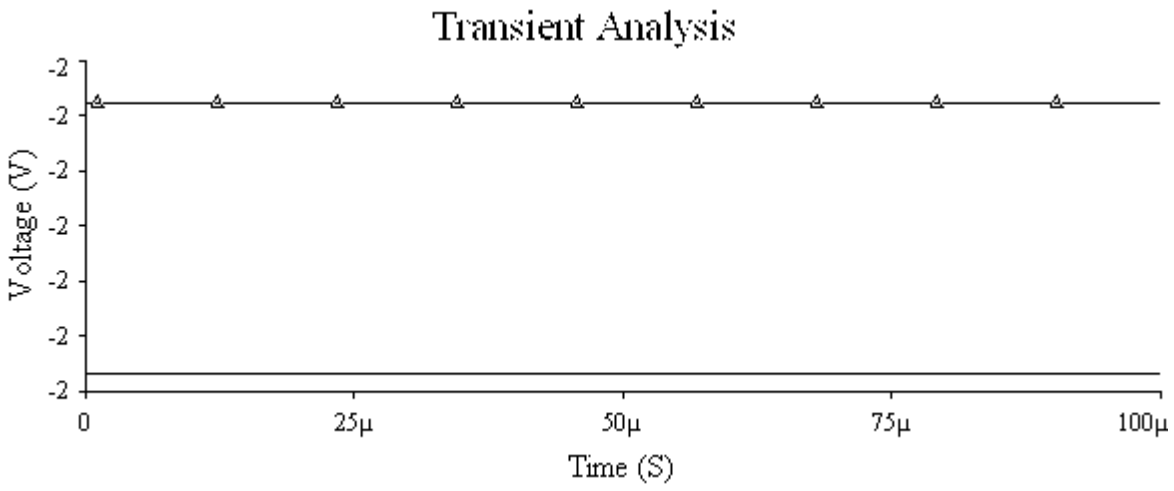


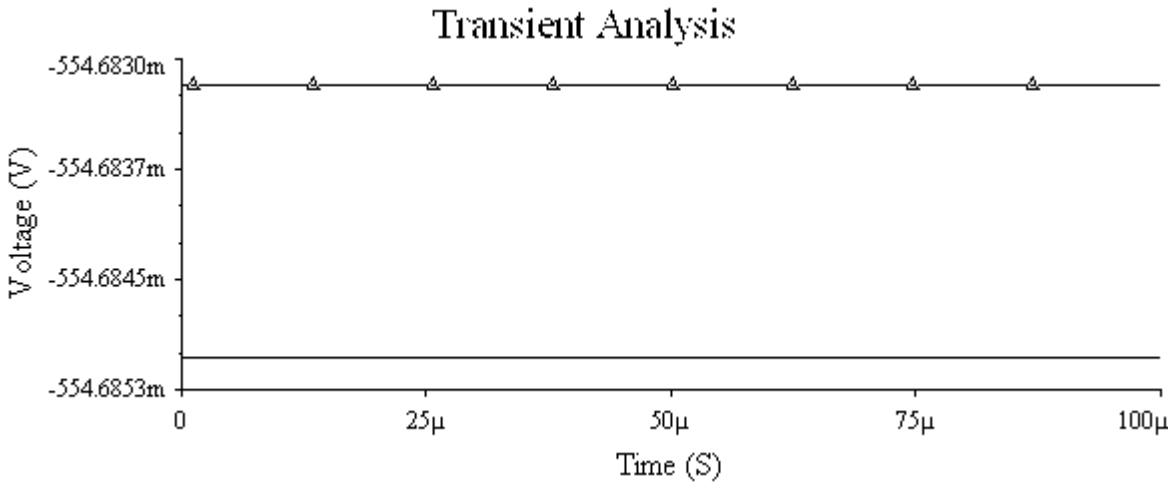
Fig. 4.6 Transfer functions analysis parameters



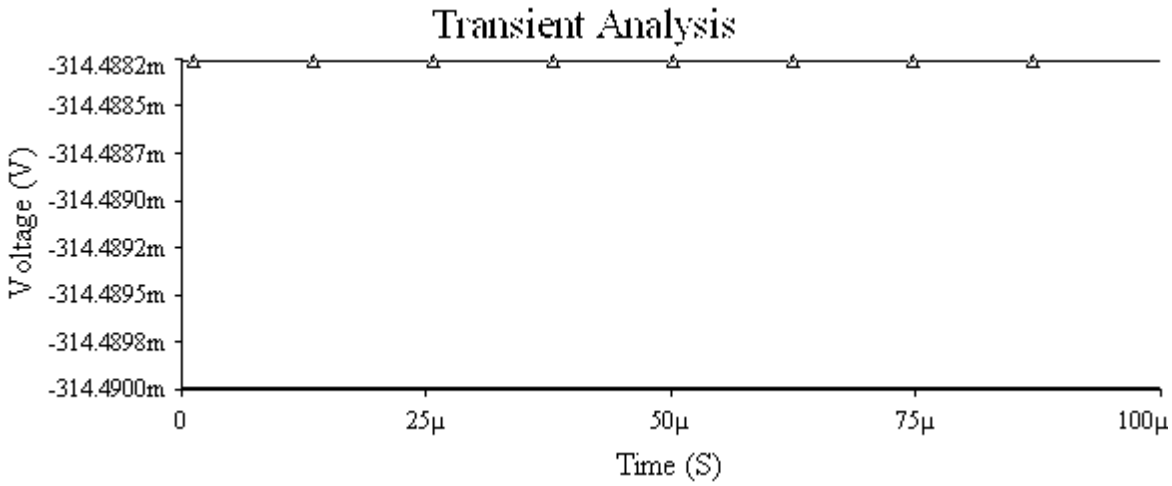
i) Tracking from "-3" input to T-gate output (bottom line:- "-3" input)



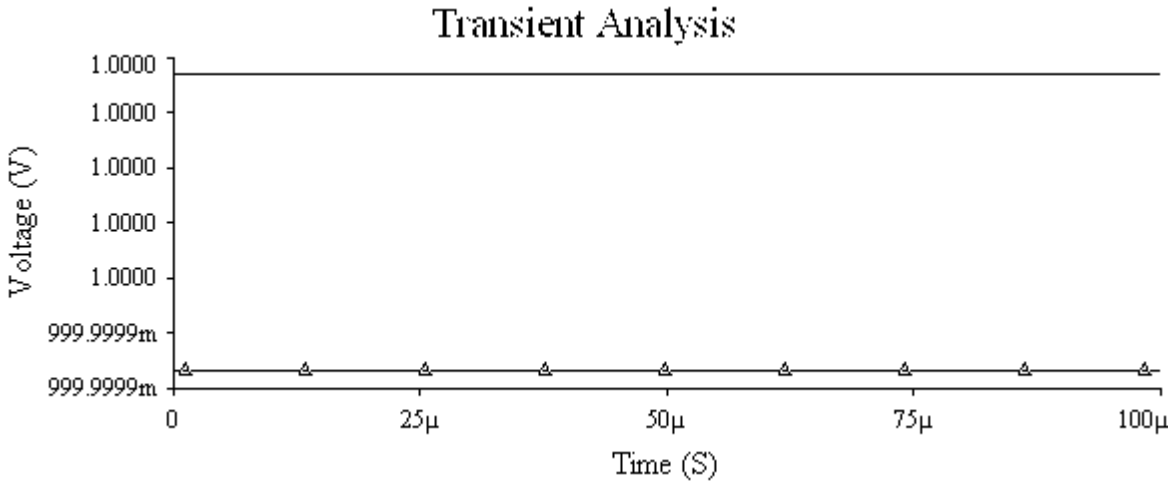
ii) Tracking from "-2" input to T-gate output (bottom line:- "-2" input)



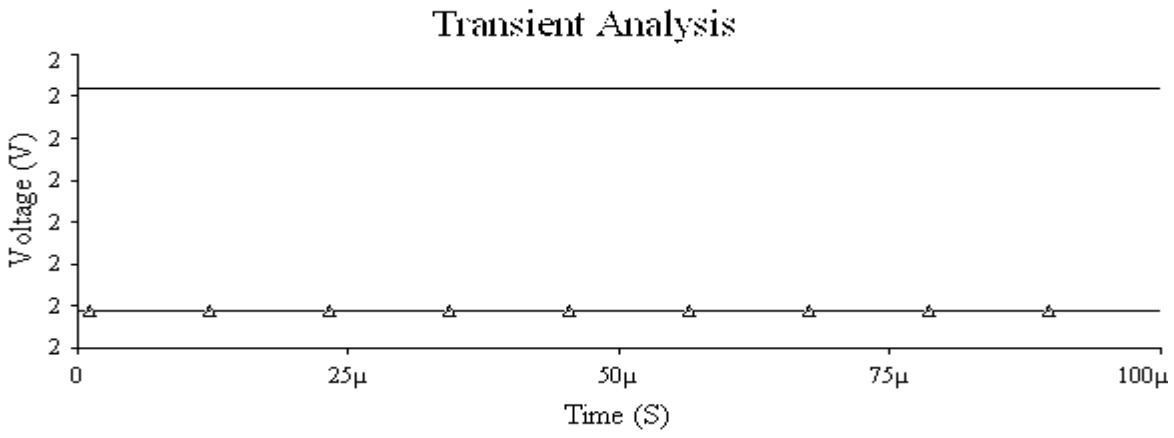
iii) Tracking from "-1" to T-gate Output (bottom line: - "-1" input)



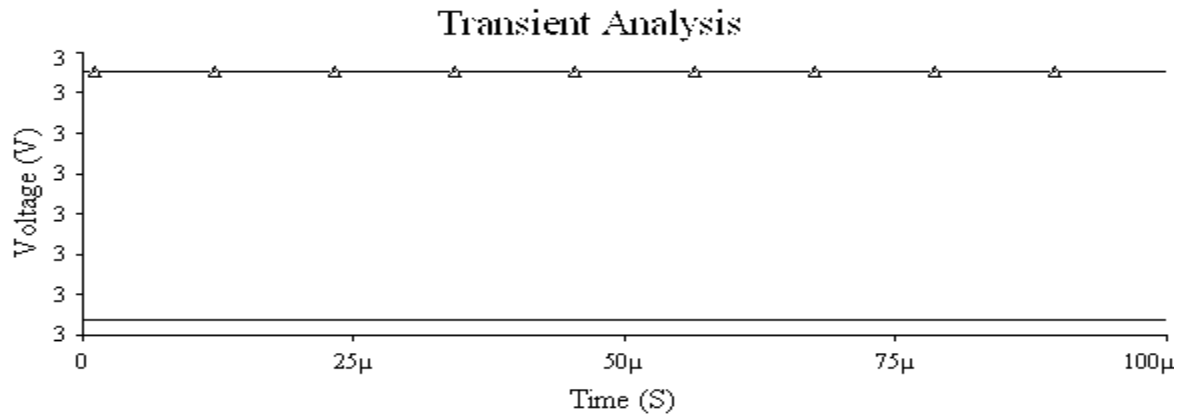
iv) Tracking from "0" input to T-gate output (bottom line coinciding with X-axis "0" input)



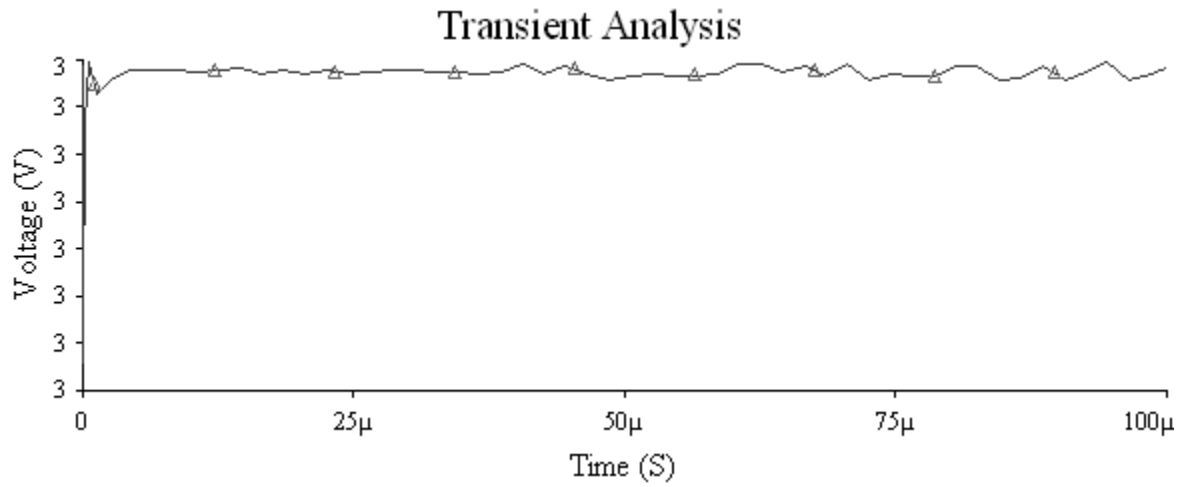
v) **Tracking from "1" input to T-gate output (bottom line:- "1" input)**



vi) Tracking from "2" input to T-gate output (bottom line: - "2" input)



vii) Tracking from "3" input to T-gate output (Top line:- "3" input)



Case at "0" input tag

Fig 4.7 Transient Analysis of the information inputs of RR7SQSD T-gate

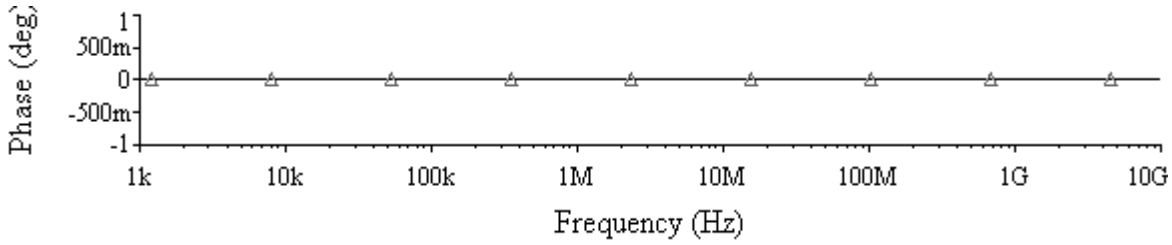


Fig. 4.8 Ac analysis

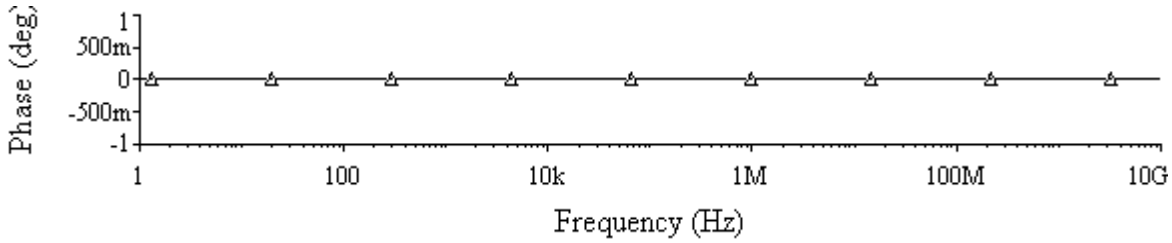
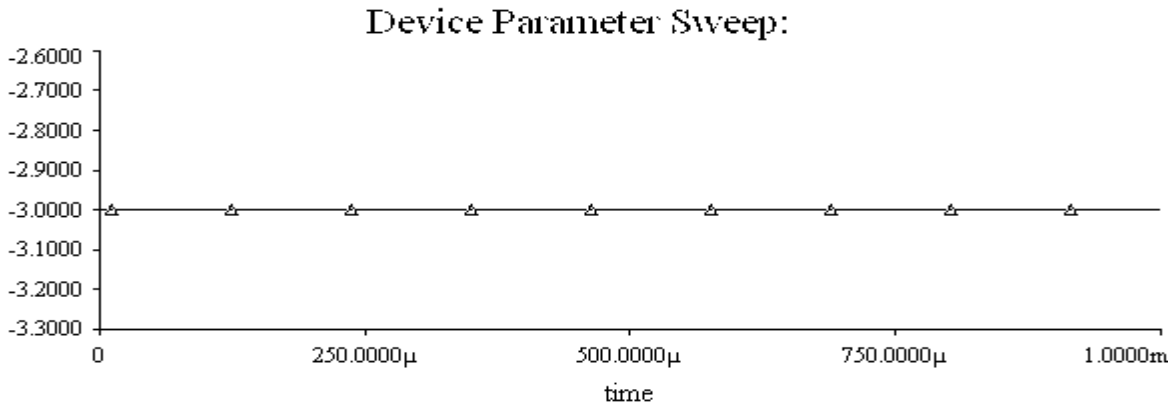
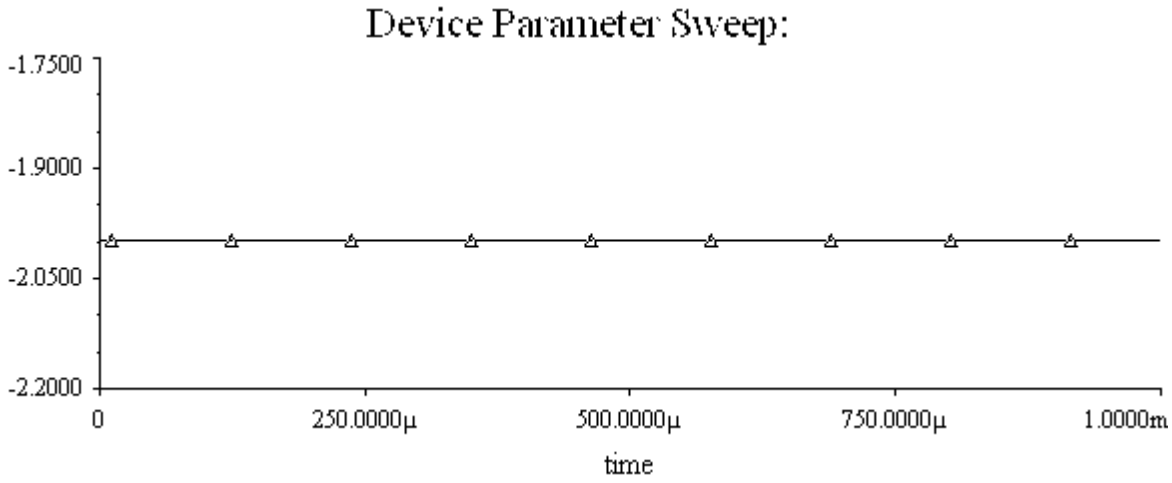


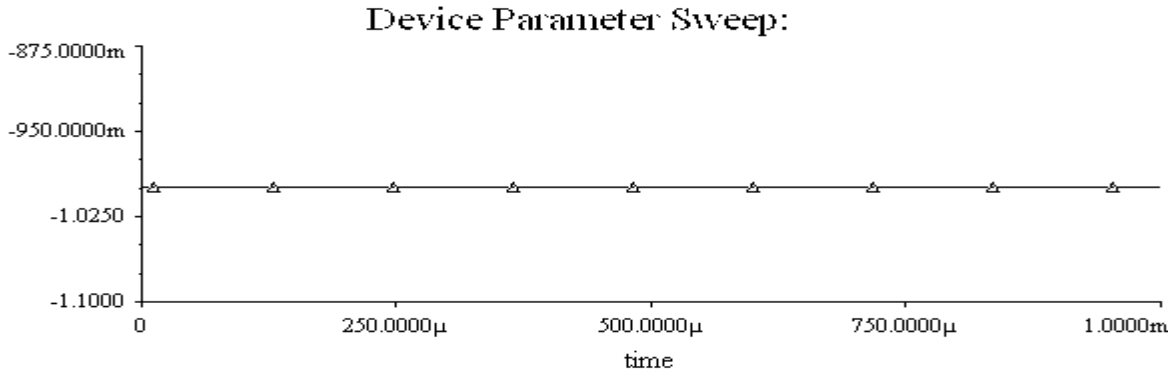
Fig. 4.9 Signal distortion plot



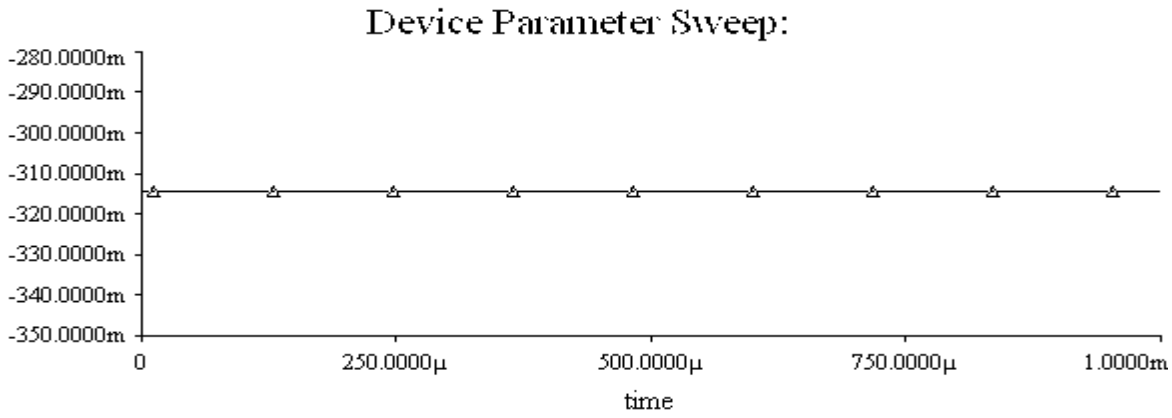
Plot at "-3" input



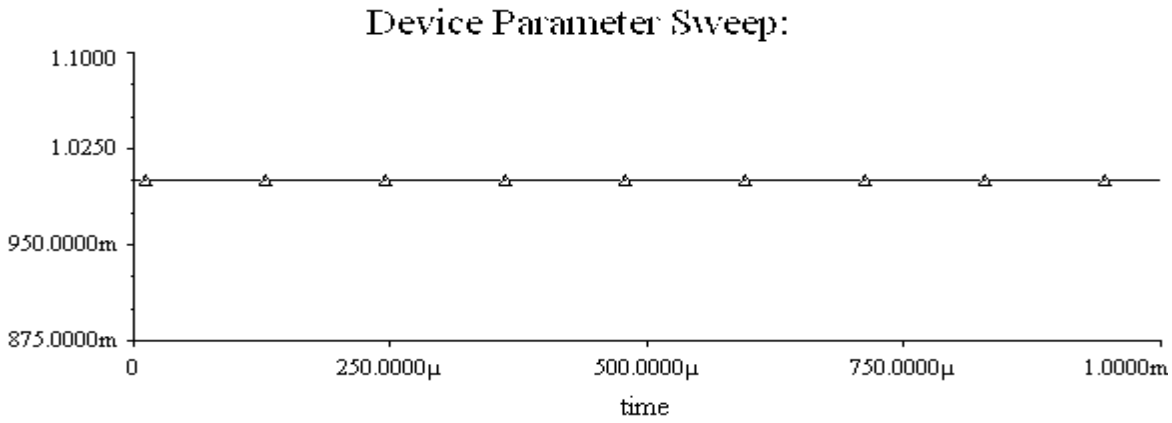
Plot at "-2" input



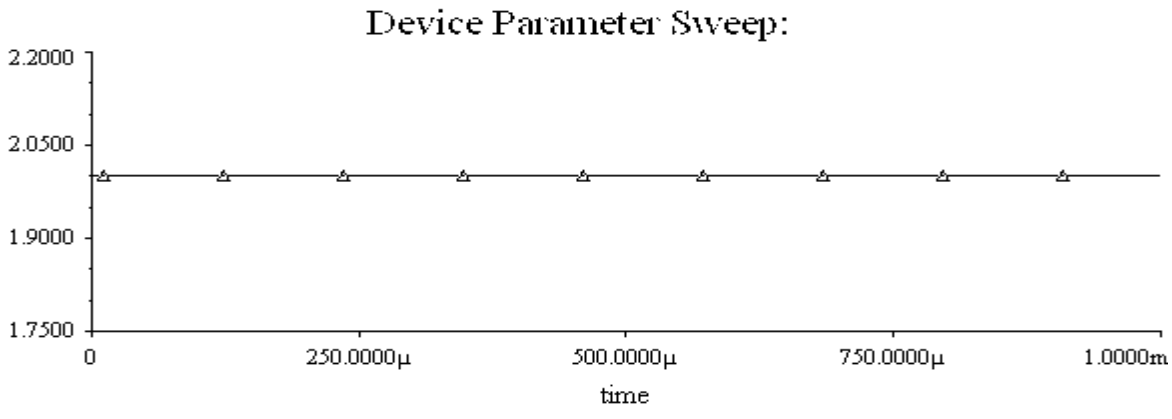
Plot at "-1" input



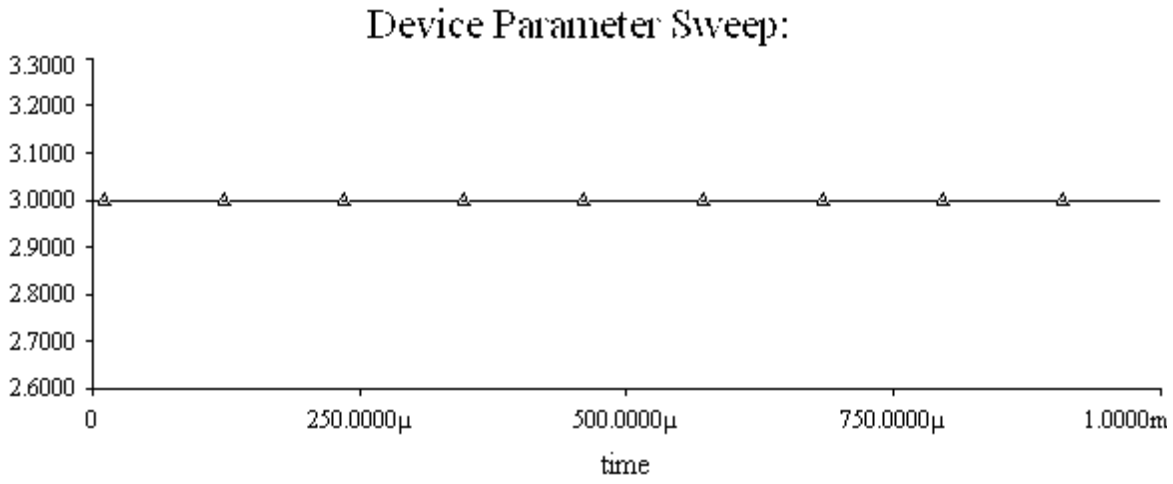
Plot at "0" input



Plot at "1" input



Plot at "2" input



Plot at "3" input

Fig 4.10 Devices parameter sweep – time duration from input to output

4.6.2 Results on the proof of multiplexing ability test

The corresponding raw results are presented in Appendix CH4.6 as the program's discrete outputs were also transferred to Microsoft excel column charts for the various inputs, control signal x, and the T-gate output. The charts are summarized by randomly selecting seven nodes for discussion. The T-gate dynamic output (Appendix CH 4.6) was equally videoed with an N6151 Nokia mobile phone camera and is presented in figure 4.11. In the column chart plots while the first seven column plots show the test data pulse train at inputs {-3,-2,-1,0,1,2,3}, Plots 8 and 9 are the control x and T-gate final output. At least 17 input pulse trains of different profiles are sent to the seven input pins simultaneously. A separate pulse train of different profile was also placed at the control input pin. Table 4.5 below shows the output of the seven randomly selected moments of the input pulse train RR7SQSD T-gate for some randomly selected control signal instants.

Table 4.5 Proof of multiplexing ability of the RR7SQSD T-gate

Pulse Train	Control X	Information signal at the inputs							T-gate Output	
		-3	-2	-1	0	1	2	3	observed	Desired
1	3	2	0	1	0	2	-2	1	1	1
8	0	-3	2	-1	2	-3	3	3	2	2
3	1	0	2	1	1	-1	-3	0	1	1
10	3	1	1	-1	2	-2	2	3	3	3
5	0	1	0	-1	2	-2	1	-3	2	2
2	-1	-3	0	-1	2	3	1	-2	-1	-1
15	-3	2	3	-1	1	2	2	1	2	2

For example, at pulse train count '1' the control signal x is "3" and the information signals: $2,0,1,0,2,-2,1$ **are sitting at the** $-3,-2,-1,0,1,2,3$ **terminals respectively. Since control x is "3", the information signal at the input terminal "3" which is "1" is selected and sent to the output of the T-gate , hence observed is "1"and desired is "1". At pulse train count '8',** $x='0'$**, and information signals -3,2,-1,2,-3,3,3 sits at the terminals corresponding to -3,-2,-1,01,2,3 and the selected output is '2'**.

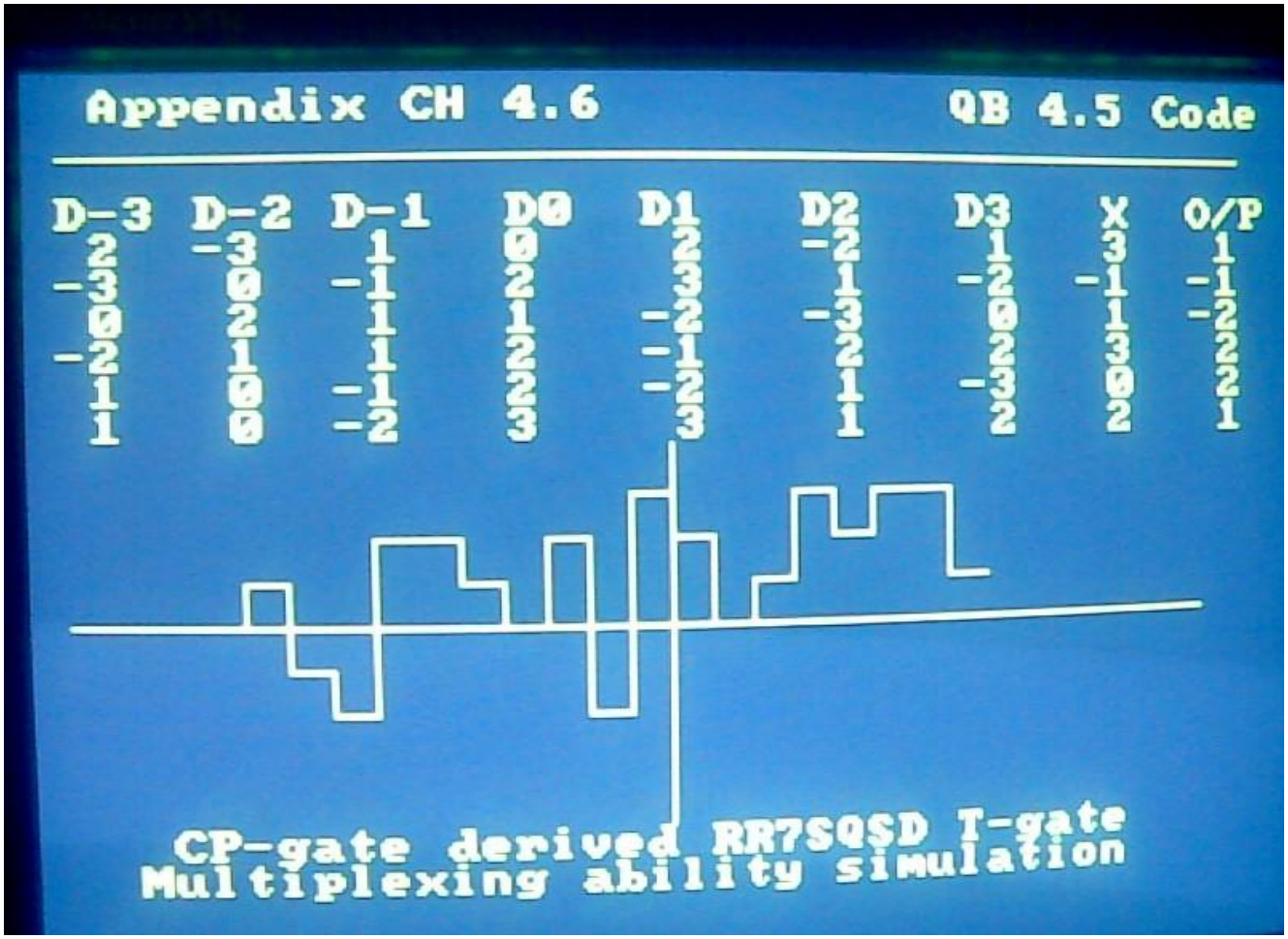


Figure 4.11: Video snapshot of RR7SQSD CP-gate derive T-gate output

The same output behaviour of the RR7SQSD multiplexer circuit is seen in the video image thus once again confirming that the circuit of figure 3.25 perfectly exhibits the multiplexing property.

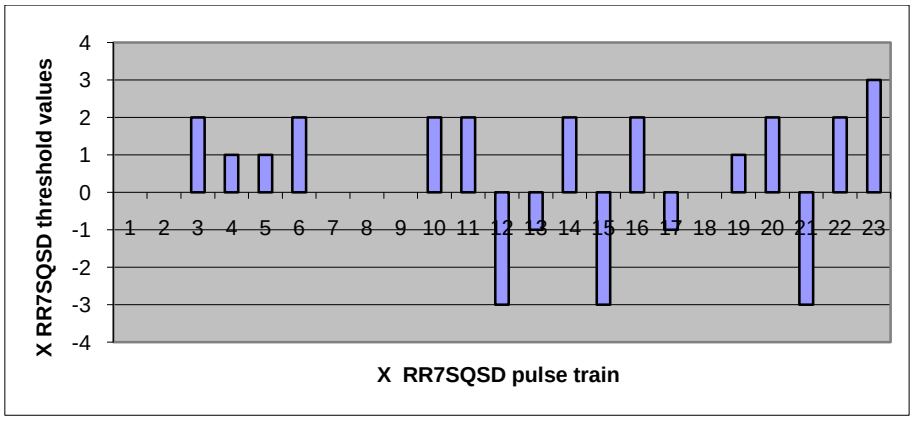
4.6.3 Results of the experiments on the RR7SQSD T-gate based full adder/multiplier

4.6.3.1 Addition mode

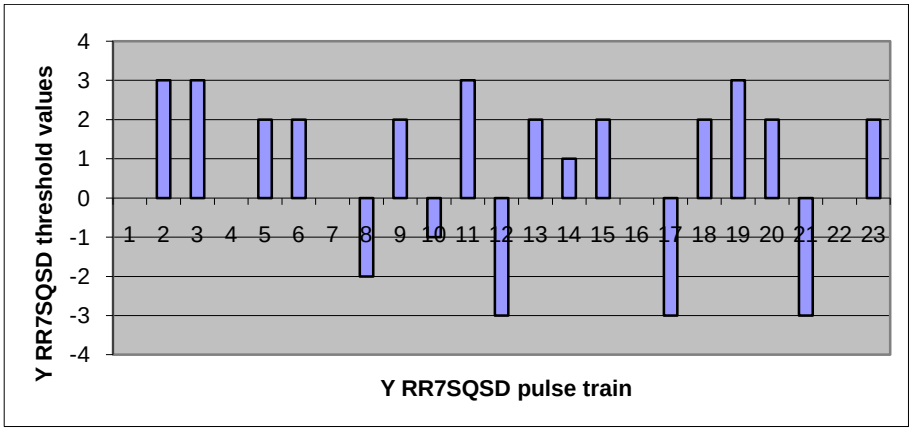
The objective of the experiment is to confirm that the circuit of figure 3.26 performs RR7SQSD addition operation as explained in section 3.2.3.7.2 based on the input data to the various T-gates in circuit. The circuit was simulated using the hardware styled QB4.5source code

program of appendix CH3.9. The Microsoft excel column charts plots of the input operands and the adders output versus input pulse trains are as shown in figure 4.12 while the video image snapshot with an N6151 Nokia mobile phone camera of the adder's output photo is as shown in figure 4.13

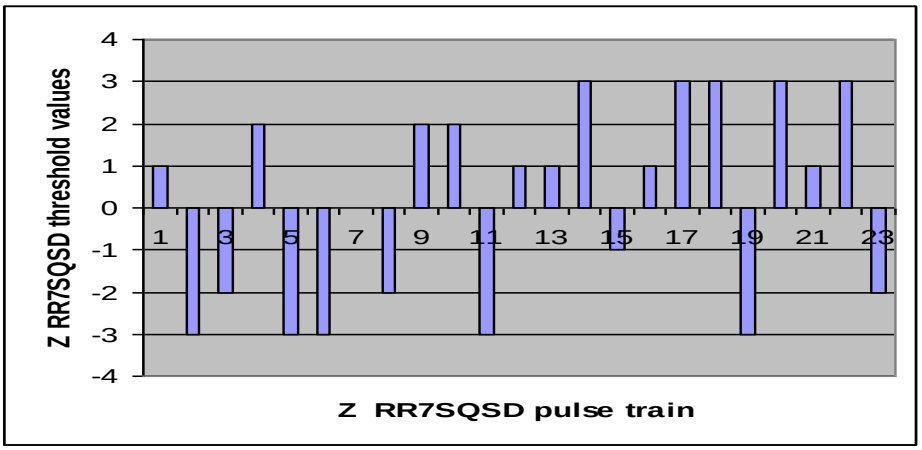
In the plots, the X-axis shows the input operands: x and y pulse trains and the Y-axis show the magnitude of the logic threshold. The third plot shows the sum of the RR7SQSD addition operation as enunciated in section 3. 6.1.2 Equation (3.47). Twenty-three pulse trains are shown. Pulse 23 is the first input pulse and pulse 1 is the last input pulse. Six random column readings are presented in table 4.6 for discussion. In the 23rd (the first from the right) input pulse train which is the least significant digit, the values of x and y are respectively 3 and 2. There is no carry-in from a previous addition so δ_{i-1}= 0 and by equation (3.4) z = -2. At the 20th pulse train x = y = 2, δ_{i-1} = 0; z = -3. In the 18th pulse train x = 0 y = 2 and δ_{i-1}=1,hence z = 3. At pulse train 10: x = 2 y = -1 and δ_{i-1}=1 to give z = 2. In the 2nd pulse train x = 0 y = 3 and δ_{i-1} =1 therefore z = -3. At the 1st pulse train (MSD) x = y = 0. δ_{i-1}=1 and z = 1.



$$X = 0,0,2,1,1,2,0,0,0,2,2,-3,-1,2,-3,2,-1,0,1,2,-3,2,3$$



$$Y = 0,3,3,0,2,2,0,-2,2,-1,3,-3,2,1,2,0,-3,2,3,2,-3,0,2$$



$$Z = x_i + y_i + \delta_{i-1} = 1,-3,-2,2,-3,-3,0,-2,2,2,-3,1,1,3,-1,1,3,3,-3,3,1,3,-2$$

Figure 4.12: Input /output pulse trains of the RR7SQSD dual full adder column charts

Table 4.6 Sample results of the RR7SQSD full adder circuit simulation

variables / value	Pulse train count					
	1st	2nd	10th	18th	20th	23rd
X	0	0	2	0	2	3
Y	0	3	-1	2	2	2
δ_{i-1}	1	1	1	1	0	0
$Z=x+y+\delta_{i-1}$	1	-3	2	3	-3	-2

$$x_i = 02112000223\bar{1}2\bar{3}2\bar{1}012\bar{3}23$$

$$y_i = 033022022\bar{1}3\bar{3}2120\bar{3}232\bar{3}02$$

$$z = 1\bar{3}\bar{2}2\bar{3}\bar{3}0\bar{2}22\bar{3}113\bar{1}133\bar{3}313\bar{2}$$

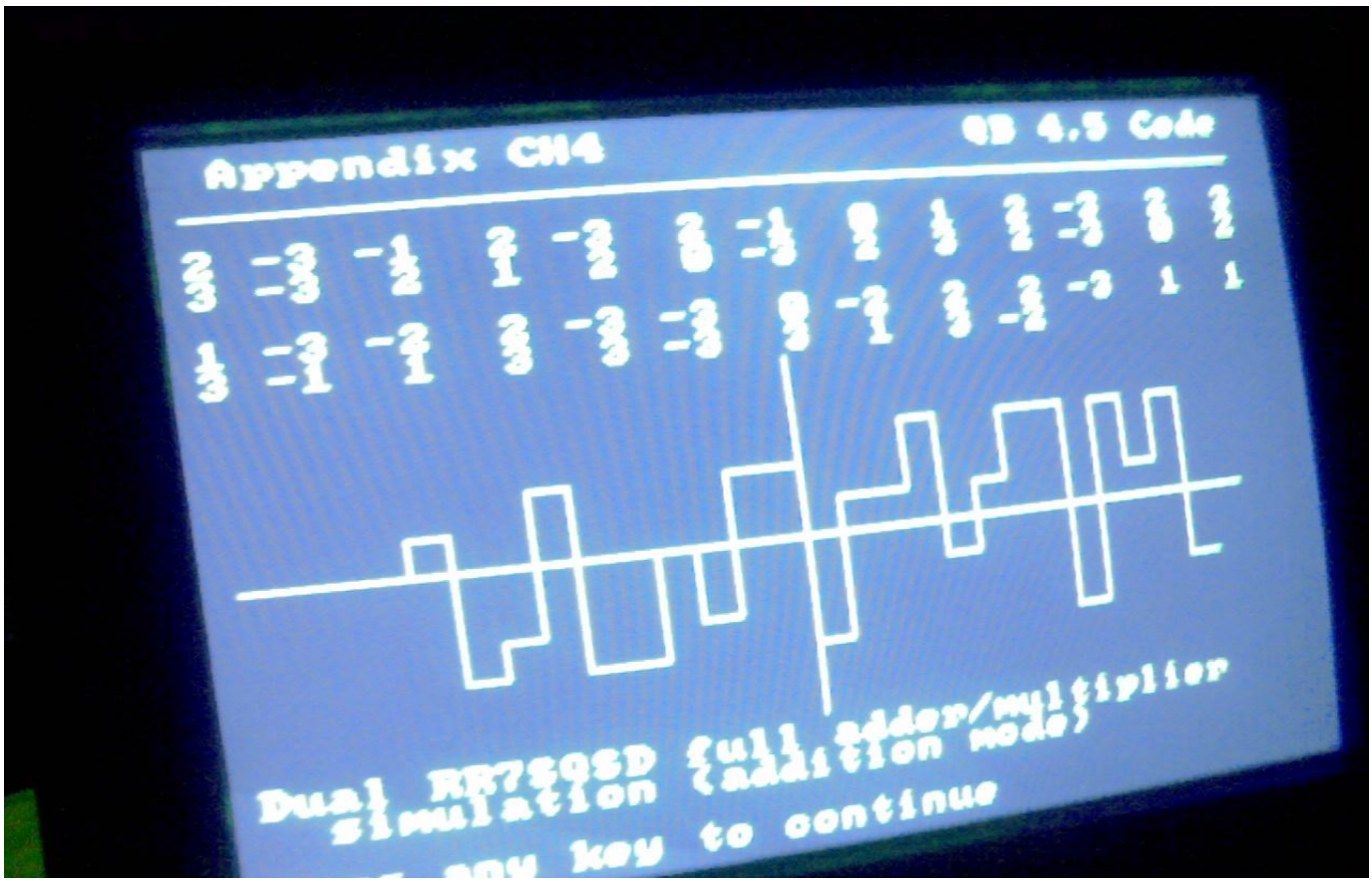


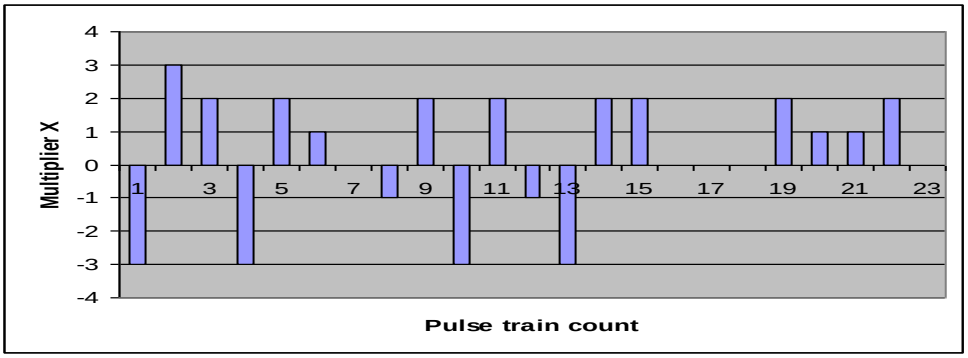
Figure 4.13: Video snap shot of RR7SQSD full adder output

The above explanation is also collaborated by the video of the circuits output as shown in figure 4.13. The result fully coincides with expectation hence it can be seen that the device function perfectly as an RR7SQSD full adder circuit.

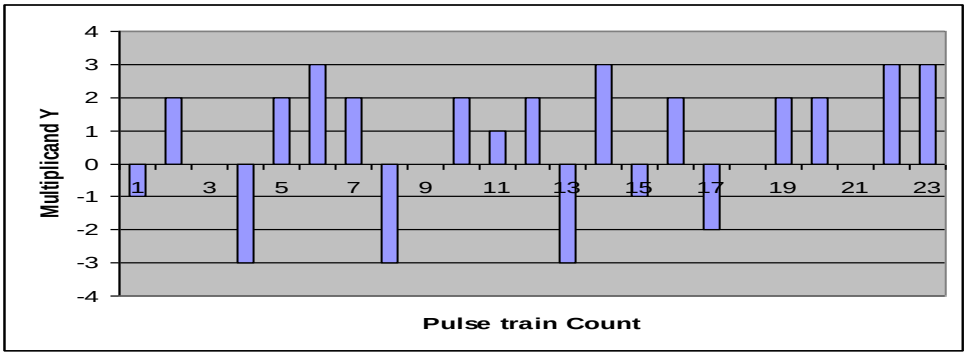
4.6.3.2 The Multiplication mode

In this mode, the device multiplies two RR7SQSD α_l, β_k together, reports a carry $\rho^o{}_{lk}$ to the next operation and adds the interim product digit λ_{lk} to a carry-in from previous operation digit δ_{lk-1} to obtain a final product digit $\rho^I{}_{lk}$. The tabular description of the operation of the device in this mode has been presented in the truth of table 3.16 and in figure 3.28 of section 3.2.3.8.1. Replacing the threshold voltages serving as input to the 21 sum/product digit forming T-gates with the appropriate entries of this table switches the device in figure 3.28 from addition mode to the multiplication mode. The circuit therefore computes the component-wise product of input operand pairs. The device was similarly simulated using the hardware styled QB 4.5 source code program of appendix program CH 3.10. The result of the simulation was transformed into Microsoft excel column chart plots are as shown figure 4.14 but summarized and presented in table 4.8. Figure 4.15 shows the dynamic trace of the same result as was video-captured with a Nokia N6115 mobile phone. The three column charts correspond to the two independent variables X_l, Y_k and the dependent output $z_i = x_j * y_k + \delta_{i-1}$.When x = -3 and y = -1, the interim product RR7SQSD is +3 and with a previous product carry-in of "1" , the final product becomes -3 and the carry-out is 1. Similarly, when x = 1 and y = 3, the RR7SQSD product of x and y for a zero previous product carry is 3. At pulse train 14 when x = -3 and y = -3 their RR7SQSD multiplication produces an interim product digit of 2 and with the previous operation carry forward digit of 1 the final product digit after the 14th pulse train is 3. The results confirm with expectations. Consequently, the figure 3.18 performs component-wise

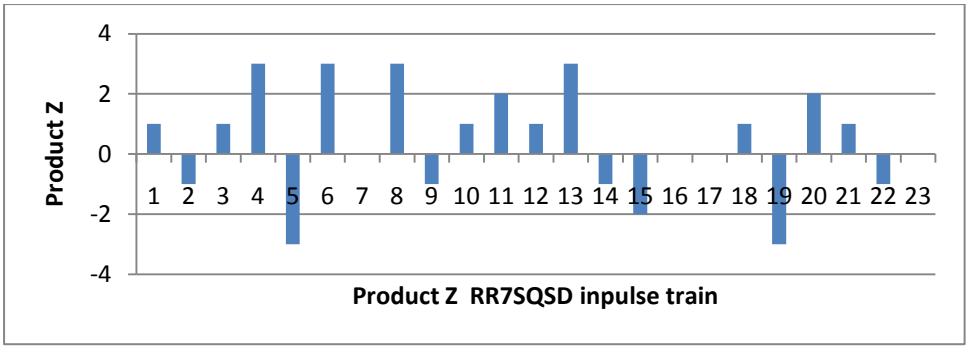
quasi-RR7SQSD full multiplication as shown both in the captured video output of figure 4.15 and the Microsoft excel column charts.



$$X_i = -3,3,2,-3,2,1,0,-1,2,-3,2,-1,-3,2,2,0,0,0,2,1,1,2,0$$



$$Y_k = -1,2,0,-3,2,3,2,-3,0,2,1,2,-3,3,-1,2,-2,0,2,2,0,3,3$$



$$Z = x_i * y_i + \delta_{i-1} = 1\bar{1}3\bar{3}303\bar{1}1213\bar{1}\bar{2}001\bar{3}21\bar{1}0$$

Figure 4.14: Input/output pulse trains of the RR7SQSD quasi-serial multiplier column charts

Table 4.7 Sample output of the quasi component-wise RR7SQSD multiplier .

$variables/value$	Pulse train count					
	2^{nd}	7^{th}	11^{th}	14^{th}	20^{th}	23^{rd}
X_l	-3	1	-3	-3	2	2
Y_k	-1	3	2	-3	2	3
δ_{i-1}	1	0	0	1	0	0
$z_i = x_j * y_k + \delta_{i-1}$	-3	3	1	3	-3	-1

$x = 32\bar{3}210\bar{1}2\bar{3}2\bar{1}\bar{3}2200021120$

$y = 20\bar{3}232\bar{3}0212\bar{3}3\bar{1}2\bar{2}022033$

$z_i = 1\bar{1}13\bar{3}303\bar{1}12\bar{1}3\bar{1}\bar{2}001\bar{3}21\bar{1}0$

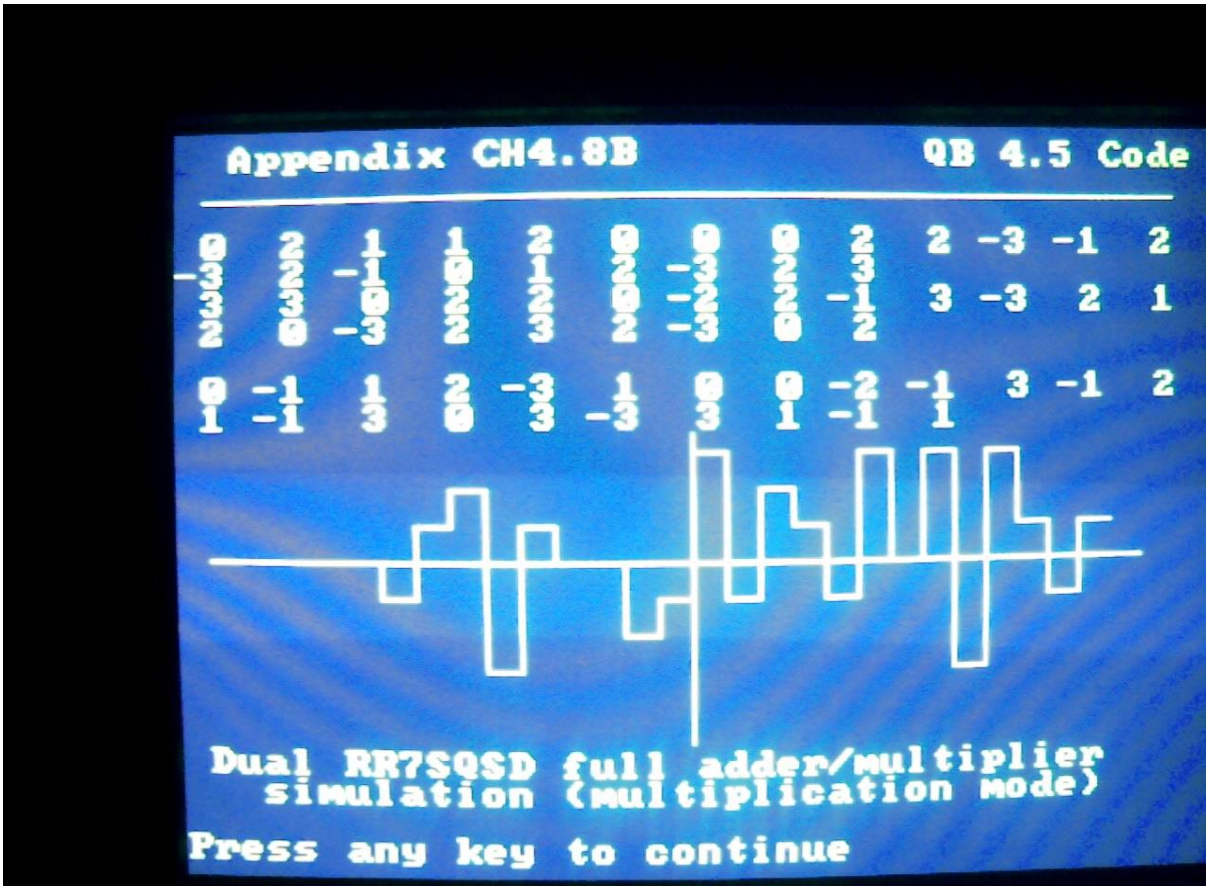


Figure 4.15: Video snapshot of RR7SQSD quasi multiplier output

CHAPTER 5

SUMMARY OF THE RESEARCH OBJECTIVES AND FINDINGS, CONCLUSION, CONTRIBUTIONS TO KNOWLEDGE AND RECOMMENDATIONS

This final chapter of the study presents: a summary of the research objectives, a summary of the research findings, concluding remarks, contributions to knowledge of importance and possible extensions of the work.

5.1 Summary of the research objectives

5.1.1 Research objective 1

Formulate appropriate strategy for performing Restricted Radix-7 Symmetrical Quaternary Signed Digit (RR7SQSD) based big integer operands parallel addition and multiplication operations.

Based on the reasons advanced in section 3.1.1.2 on the choice of radix-7 , the following numerical examples were presented:

i) **numerical examples 3.1 in section 3.1.1.5 on RR7SQSD parallel addition operation with large integer operands**
ii) **numerical example 3.2 in section 3.1.1.6 on multiplicative inverse computation in the RR7SQSD domain**
iii) **numerical example 3.3 in section 3.1.1.8 on RR7SQSD addition chain calculation and**
iv) **numerical examples 3.4 and 3.5 in section3.2.1 on RR7SQSD modular exponentiation computation;**

These collectively exemplify clearly the potentials of RR7SQSD arithmetic. The result of the RR7SQSD multiply-by-7 addition/subtraction chain big integer multiplication experiment presented in section 4.1 figure 4.1 also clearly consolidate the realization of the research objective 1.

5.1.2 Research objective 2

Establish the necessary basis of efficient arithmetic operations on EC defined over RR7SQSD element finite fields suitable for secure communication systems and prove that RR7SQSD element field based ECC is message degradation and repudiation free using a particular cryptosystem protocol as a case study.

This second objective is partially established with the help of solution to the:

i) **numerical example 3.6, figure3.14 in section 3.2.2.2 on RR7SQSD finite field element inverse computation and**
ii) **numerical example 3.7 figure 3.15 in section 3.2.2.3.4 and example 3.8 table 3.6 in section 3.2.2.3.5 all on integrity of the points' addition and point doubling operations on EC defined over RR7SQSD finite fields.**

The objective is fully established with the results of the:

a) **SGF(7^2) construction in section 4.2.1 figure 4.2, field inverse element computation in section 4.2.2.1 figure 4.3 and section 4.2.1.2 figure 4.4.**
b) **Experiments on: message embedment in section 4.3 figure 4.5, point multiplication using the one-stop multiply-by-7 addition/subtraction chain algorithm on EC defined**

over RR7SQSD finite fields in section 4.4 table 4.19 and table 4.2. Others are the: computation order of complexity calculation in section 4.4.2 table 4.3 and the combined message encryption and decryption experimental result in section 4.5 table 4.4.

5.1.3 Research objective 3

Establish the architectural organization of the RR7SQSD based ECC arithmetic unit VLSI circuit.

This objective is collectively established by the followings:

i) **Solution to numerical example 3.9 on the cost of forming 314159P using the multiply-by-7 algorithm in section 3.2.1.1 table 3.11 and the resultant data flow diagram in section 3.2.3.3 figure 3.22.**

ii) **The operation speed and total number of the co-processor's general purpose registers estimation in section 3.2.3.4 table 3.12 and**

iii) **The ensuing final block diagram of the point multiplication VLSI circuit in section 3.2.5 figure 3.24.**

5.1.4 Research objective 4

Design the VLSI circuit's basic building block, derive the exact LSI circuit diagram of the main functional logic units using the basic building block and show where possible, how they operate to design specification.

The realization of this objective consists of three parts namely: i) building block design, ii) synthesis of the functional logic units and iii) the conduction of tests to confirm their

proper function. The design of the building block (RR7SQSD T-gate) started with its formulation in section 3.1.3. Through section 3.1.3.1 on design specification, section 3.1.3.5 on RR7SQSD CP-gate derived T-gate equation to the realization of RR7SQSD T-gate transistor circuit diagram in section3.2.3.6.

Derivation of the VLSI's various functional logic units' LSI circuit diagrams is accomplished in section 3.2.3.8. For example, in section 3.2.3.8.1 the synthesis of the one- RR7SQSD dual XOR/multiplier circuit tables 3.13 , 3.14 and figure 3.28 metamorphosed into the RR7SQSD full adder circuit design as depicted in tables 3.15 , 3.16 and figure 3.29. The section ends with the presentation of the RR7SQSD parallel addition LSI circuit in figure 3.31 to 3.32. Similarly, synthesis of Field Element Addition Multiplication circuit (FEAM) in table 3.17 and figure 3.34 facilitated the design implementation of the LSI circuits: $SGF(7^m)$ field element parallel adder as shown in fig. 3.35 section 3.2.3.9.1 and the $SGF(7^m)$ field elements parallel multiplication as shown in figure 3.38 in section 3.2.3.9.2. The RR7SQSD: multiplexer LSI circuit, 4-RR7SQSD universal register and RR7SQSD sign inverter MSI circuits are presented in sections 3.2.3.1 to 3.2.3.3.

The experiments in section 3.2.3.7.1 and 3.2.3.7.1 tested the building block's conformity to design specifications and operational expectations. Similarly, the experiments in sections 3.2.3.12.1 and 3.2.3.12.2 respectively tested the accuracies of the RR7SQSD full adder circuit and quasi-multiplier in serial modes. The plots of fig. 4.6 to fig. 4.10 in section 4.6.1 are the results of the building block's electrical parameters test. The results in table 4.5 and the trace of fig. 4.11 are proves of its multiplexing ability.

confirmed the building block's operating to specification. Also the results discussed in section 4.6.3.1 and 4.6.3.2 separately showed the accuracies of the RR7SQSD full adder and multiplier.

5.2 Summary of findings

The main findings of the research efforts described in this presentation are as follows:

i) **The RR7SQSD number system which has been shown here to support fast and efficient integer arithmetic has also been shown to have the capability of performing big integer operand multiplication operation in word length-compute- constrained environments without truncation.**

ii) **High speed point multiplication and thus cryptographic protocol operations can be executed on EC defined over RR7SQSD element composite finite fields using a multiply-by-7 addition/subtraction chain.**

iii) **The one RR7SQSD addition operation logic circuit is the equivalent of RR7SQSD XOR operation and thus can be used to design RR7SQSD field element parallel adders and multiplication circuits.**

iv) **The RR7SQSDNS is a SMVL system - appropriate as an alternative processing threshold logic system to the binary logic. Consequently, a SMVL ECC arithmetic unit VLSI circuit can be implemented using RR7SQSD element finite field inverters, registers, and parallel adders and multiplexers LSI circuits. The basic building block of these circuits can be designed to operate directly on the RR7SQSD processing logic signal threshold.**

5.3 Concluding Remarks

The results of this research study have brought into the fore the potentials of the RR7SQSDNS for application in conventional and RR7SQSD element finite field ($SGF(7^k)$) arithmetic and thus for hardware implementation of elliptic curve cryptosystem protocols. The results also serve as a proof that the RR7SQSDNS is an appropriate SMVL processing threshold signal profile as the implemented arithmetic LSI circuits demonstrated the ability to execute specified operations accurately and independent of operands' word length. Consequently, RR7SQSDNS based high performance processors can be designed and implemented for the emerging secure and trusted communication gadgets.

5.4 Contributions to knowledge

The main contributions claimed by the thesis to knowledge of importance are as follows

During the period of this research activity the following contributions to knowledge are made.

i) **Development for the first time, of two efficient RR7SQSD number system based big integer arithmetic operation algorithms: one for performing carry-free large integer parallel addition operation and the other, for multiplying big integer operands in word length compute constrained environments without truncation.**

ii) **The novel design of an efficient multiply-by-7 addition/subtraction chain based hardware-implementable algorithm, for performing fast point multiplication operation on EC defined over RR7SQSD element finite fields.**

iii) **Acceptance by the International Scientific Community, of a completely new algebraic structure, the RR7SQSD element finite field, its attendant arithmetic**

and the realization of a RR7SQSD T-gate based $SGF(7)$ field element dual addition/multiplication (FEAM) circuit as the basic component for performing $SGF(7^m)$ field element parallel addition and multiplication operations.

iv) Demonstration of the:

a) viability of the RR7SQSDNS (otherwise herein referred to as the SMVL system) as an appropriate alternative to the binary processing logic threshold system,

b) formulation and design implementation, using CP-gate derived 7-valued (or RR7SQSD T-gate) as a basic building block for SMVL ECC arithmetic unit VLSI circuits design, and

c) methods of synthesizing the VLSI's constituent arithmetic, data transfer and information storage LSI circuits on the building block's level/.

5.5 Possible extensions

The unavailability of staple digital design flow component tools, namely:

i) The Very High Speed Integrated Circuit Description Language (VHDL) and HSPICE softwares

ii) Parallel computing facility

made it correspondingly difficult to:

a) Simulate the more complex functional logic circuits of the SMVL ECC arithmetic unit VSLI in parallel mode.

b) Perform real time analogue simulation of the basic building block circuit.

c) Use practical or large modulo prime values of $p = 7^m \pm \psi_i$ for example,

$$p = 785963102379428822376694789446897396207495868951$$

It is therefore recommended that:

i) **The VLSI circuit is VHDL simulated with a view to evaluating specific performance.**

ii) **carry out real time analogue simulation using HSPICE, of the RR7SQSD T-gate circuit for a bench-marked result analysis.**

iii) **Conduct crypto analysis with more practical values of p that will enable the measurement of coding breaking times of messages encrypted on elliptic curves defined over RR7SQSD finite fields.**

CHAPTER 6

REFERENCES

---

[1] Elizabeth Oswald, "Introduction to ECC" Institute for Applied Information processing and Communication "A-8010infledgasse 16a Gra 3 Austra, July 29, 2005. pp 1-20. http://www.iaik.tugraz.at/aboutus/people/Oswald/introduction to ECC.pdf

[2] Antonov Alex, Frank Gounelas, Jeffrey Kauppila. "Cryptography" http://people.maths.ox.ac.uk/gounelas/project/suprema-cryptography.pdf

[3] Wendy Chou", Elliptic curve cryptography ËCC and its Application to mobile Devices". University of Maryland , College park. Department of Mathematics. http://www.cs.umd.edu/Honor/report/ECCPAPER.PDF, 2009

[4] Dimitrov,V.S.; Imbert, L.; Mishra,P.K.; "Fast Elliptic curve point multiplication using Double-base Chains"; University of Calgary, 2500 University drive NW Calgary AB, T2n In4, Canada 2005 pp1-13 http://eprint.iacr.org/205/069.ps.

[5] Jean-Yves Chouinard "Design of Secure Computer Systems CAS14138/CEG4394 Notes on Elliptic Curve Cryptography", University of Ottawa, September 2002. http://www.site.aottwa.ca/~chouinar/publication.

[6] Chang N. Zhang and Hua Li. "A fast VLSI algorithm for multiplication on elliptic curves", Proceedings of the 2001 IEEE workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 5-6 June 2001, pp 175 -178.

[7] Kwang Ho Kim, SoIn Kim and Ju Song Choe, "New Fast Algorithm for Arithmetic on Elliptic Curves over Finite fields of Characteristic Three"; Department of Algebra,

Institute of Mathematics national Academy of Sciences, D.P.R. of Korea, 2007. kimkhhj1980@yahoo.com.cn.

[8] Atsuko Myyaji, Takatoshi Ono and Henry Cohen, "Efficient elliptic curve exponentiation", Multimedia development centre. Matsushita Electric Industrial co. Ltd. Matsushita Information Systems Research laboratory Nagoya Co., Ltd. Universite Bordeaux, 2000, 10 pages .http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.25.1196[1].pdf.

[9] Yasuyuki Sakai and Koushi Sakurai " Efficient scalar multiplication with direct computation of several Doublings" :IEICE Trans. Fundamentals, Vol. E84-A N01 January 2001 pp 120-128.

[10] Yuliag Zheng and Hideki Imai, "How to construct efficient signcryption schemes on elliptic curves", Information Processing Letters 68(1998) issue 5, pp 227-233. http://portal.acm.org/citation.cfm?id=306053.

[11] Anoop MS "Elliptic Curve Cryptography - An Implementation guide". 2008. http://www.tataelxsi.cer/whitepapers/ECC_Tut_v1_0.pdf_id

[12] Mathehieu Ciet, Marc Joye, Kristin Lauter L. Montgometry "Trading inversion for multiplication" Microsoft research, One Microsoft Way, Redmond, Wa 98052, USA. http://eprint.iacr.org/2003/257.

[13] Toshio Hasegawa, Junko Nakajima and Mitsuru Matsuf. " A small and fast software implementation of elliptic curve cryptosystem over GF(p) on a 16-bit

microprocessor", IEICE Trans., Fundamentals, Special section on cryptography and information security; Vol. E82-A, No 1 January 1999. pp 98 – 105.

[14] Jean-Claude Bajard, Laurent Imbert and Christophe Negre (2006). "Arithmetic Operations on finite fields of medium prime characteristic using the Lagrange representation"; IEEE Transaction on computers, Vol. 55, No 9, September 2006. pp 1167 – 1177.

[15] "Certicom ECC Challenges". Ecc.project@edu.informatik.tu-darmstadt.de. 2003. http://www.certicom.com/index.php/the-certicom-ecc-challenge

[16] Naoya Torii, Kazuhiru Yokoyama. "Elliptic curve cryptography", FUJUTSU Sci. Tech. J. 36,2 December 2000 pp 140 – 146 .

[17] "An intro to Elliptic Curve Cryptography-Deviceforge.com ….tomorrow's device technology today", http://www.DeviceForge.com/articles/AT4234154468, July 20, 2004..

[18] David Malan, "Crypto for Tiny Objects", Computer Science group , Harvard University Cambridge, Massachusetts , 2004. http://www.cs.havard-edu/malan/publication/tr-04-04.pdf.

[19] Andre Weimerslarch, Chrislot Paar, Sheuding Chang Shankz, " EC on Palm OS Devices". Worcester Polytechnic Institute U.S.A Third conference on Network and security Research, Sun Microsystems laboratory. USA. 2001.

[20] Leif Ussadel,Axel Poschmann and Christof Paar " An efficient general purpose ECC module for Ubiquitous Sensor network" Horst Gortz Institute for I.T security,

Communication security Group, Ruhr-Univeritat Bochum Germany.2008. http://www.cosic.esat.kuleuven.be/publications/article-1015.pdf.

[21] Scott A. Vanstore. "Next generation Security for wireless elliptic curve cryptography" Certicom. 2003. http://www.compseconline.com/hottopic20-8/next.pdf.

[22] Kefa Rabah ; "Elliptic curve cryptography over binary finite fields GF(2^m)" Asian network for Scientific Information. ISSN 1812-5638. Information Technology Journal 5(i) : 204 -229, 2006,

[23] Vanston, S.A, "Next generation security for wireless: elliptic curve cryptography", EVP strategic Technology, certicom. 0167-4048/03ElsevierLtd, 2003. http://www.compseconline.com/hottopic120-8/next.pdf.

[24] Melari Nano; "Roadmap for Nanoelectronics"; European Commission 1st programme. Future and Emerging Technologies Microelectronics Advanced Research Initiatives, 2008. http://cordis.europa.eu/esprit/src/melna-rm.htm.

[25] Ton Van oon and Ray Marston; "Transistor tutorials part 3". 2008. http://www.sentex.ca/~mech1995/tutoria,xtor3

[26] Robert R. Krchnavek "Nanoelectronics "Rowan university spring 2006.

[27] Giovanni De Micheli "Nanoelectronics: Challenges and Opportunities", Centre Systèmes Intégrés, EPA. http://infoscience.epfl.ch/record/90763/files/patmoso6.pdf

[28] Paul Beckett and Andrew Jennings. "Toward Nanocomputer architecture" , Australian Computer Society (2002). Proceedings of the seventeenth Asia-Pacific

Computer systems Architecture Conference (ACSAC "2002), Melbourne, Austria Conference in Research and Practice in Information Technology Vol. 6, Reipe and John Morris Eds.

[29] Emre Ozer, Resit Sengag and David Gregg; "Multiple-valued logic buses for reducing Bus size, transitions and power in deep sub micron Technologies" . Advanced Network and Communication Hardware workshop(ANCHOR) in conjunction with international Symposium on Computer Architecture (ISCA-32). June 2005. Department of Electrical and Computer Engineering, University of Rhode Island , U.S.A. http://www.elc.uri.edu/faculty/sendag/research htm.

[30] John Michell "Cryptography, briefly" 802.11b slides from Dan Boneh, 2009. http://crypto.stanford.edu/CSISS old-spring 02/lecture 8.pdf.

[31] Shoji Kawahito, Michitaka Kameyama , Tatsuo Higuchi and Haruyasu Yamada,"A High Speed Compact Multiplier Based on Multiple-Valued Bi-directional Current-Mode Circuits"; Proceedings of the Seventeenth International Symposium on Multiple-Valued Logic , May 26 – 28 1987 Boston, Massachusetts, pp 172 - 180.

[32] Michitaka Kameyama and Tatsuo Higuchi, "Multiple-valued logic and special-purpose processors: Overview and future" Proceedings of the twelfth international symposium on multiple-valued logic, May 25-27 1982 . Paris, France . pp 289 – 292.

[33] Michitaka Kameyama and Tatsuo Higuchi, "Design of Radix 4 Signed-Digit Arithmetic Circuits for Digital Filters" ; Proceedings of the Tenth International

Symposium on multiple-valued logic, Northwestern University Evanston, June 10th, 1980, Il pp 272 - 277.

[34] Katsuhiko Shimabukuro, Michitaka Kameyama and Tatsuo Higuchi, "Design of a Multiple-Valued VLSI Processor for Digital Control"; Proceedings of the Twenty-second International Symposium on Multiple-Valued Logic, pp322 -329, May 27 – 29 1992; Sendai, Japan.

[35] Makolo Honda, Michitaka Kameyam and Tatsuo Higuchi;" Residue Arithmetic Based Multiple-Valued VLSI Image Processor", Proceedings of the Twenty-second International Symposium on Multiple-Valued Logic, pp330-336, May 27 – 29 1992; Sendai, Japan.

[36] T Higuchi and M Kameyama, " A new digital Image Processor using MVL ". Proc. Twelfth international symposium on MVL. Pp. 8 – 16, May 1982.

[37] Michitaka Kameyama and Tatsuo Higuchi, "Synthesis of Multiple-valued logic network based on Tree-type Universal logic module". IEEE Transaction on computers Vol. C26, N0 12, December 1977; pp 1297 – 1302.

[38] Tatsuo Higuchi, Michitaka Kameyama. "Static-hazard-free T-gate for ternary memory element and its application to ternary counters", IEEE Transactions on Computers, Vol. C-26, N0 12, December 1977. pp 1212 – 1221.

[39] Michitaka Kameyama ,Takahiro Hanyu and Tatsuo Higuchi ,." Design and implementation of quaternary NMOS integrated circuits for pipelined image processing." IEEE Journal of solid- state circuits. Vol. Sc 212 N01. pp. 20 – 27 1987.

[40] Kazuhiro Horie, Kyu-ik Sohng, Michitaka kameyama and Tatsuo Higuchi. "Design of a complementary pass gate network for a multiple-valued logic system"; Proceedings of the seventeenth international symposium on multiple-valued logic , May 26 -28 1987 Boston, USA. Pp 142 -149.

[41] Israel Koren " Digital Computer Arithmetic ECE 666 Part 2, Unconventional Number System" . University of Massachusetts, Dept of Electrical of Computer Engineering, MA. spring 2004 pp. 1 - 30, MA USA.

[42] A. Aviziens, "Signed digit number system representations for fast parallel arithmetic ", IRE Trans. Elect. Comput., EC – 10, pp 389 – 400, Sept 1961.

[43] Daikpor, M.N and Oluwole Adegbenero. " Synthesis of a Decimal Adder Using 7-Valued Complementary Pass gate Derived T-gate". Proceedings of the International Conference in Engineering on Technology and Resource Management. 23 –26 May 2005, Lagos, Nigeria, pp 383 - 398.

[44] Chung-Kuan Cheng;"CSE 246: Computer Arithmetic Algorithms and Hardware Design. Lecture 2 Redundant and Residue Number Systems 2006.

[45] Anders Lindstrom, Michael Nordseth Lars bengsson and Amos Omondi;" Arithmetic circuits combining residue and signed-digit representations"; Proceedings of the eighth Asia-Pacific Computer systems Architecture ,conference (ACSAC'2003) Aizu-Wakamaisu City, Japan Sept . 2003; Notes in computer science (LNCS) Vol. 2823, 11 pages.

[46] Anders Lindstrom, Michael Nordseth and Lars Bengtsson;"VHDL Library of Nonstandard Arithmetic Units"; Department of Computer Engineering, Chalmers University of Technology, Technical Report 03-01, August 2003, pp 1-16.

[47] Dhananjay S. and Israel Koren ; " Hybrid Signed-Digit Number Systems: A Unified Framework for Redundant Number Representation with Bounded Carry propagation Chains. "IEEE Transaction on Computer, August 1994,Vol 43, No.8 , pp 880-890. MA USA.

[48] Shao-Hui Shieh and Chieng-Wen Wu; "Asymmetric High-Radix Number System for Carry free Addition"; Journal of Information Science and Engineering 19, Taiwan (2003), pp 1015-1039.

[49] Raymond Hill " A first Course in Coding Theory" Oxford Applied Mathematics and Computing Series, 1990.

[50] Wesley Peterson, and Weldon; E.J (Jr), "Error Correcting Codes". 2nd Edition. The MIT Press, Cambridge, Massachusetts and London, England. 1971.

[51] Keown; R." An introduction to Group Theory" Academic Press London, 1975.

[52] "Introduction to finite fields". File://A:\Galois% 2002 \Galois 2005 htm.

[53] George Barwood, "Elliptic Curve Cryptography FAQ v1.12 22nd December 1997 http://ds.dial/pipex.com/george.barwood/ec.faq.txt.

[54] Adan Mohammed I. Abu Mahfouz"EC cryptosystem over optimal extension fields for computationally constrained devices". M.Sc Thesis. University of Pretoria. Pretoria, Sept 2004.

[55] John Keri "Computation in finite field". Arizona State University and Lockheed. Martin Corporation. April 2004. Pp 1-83.

[56] V. Daniel, C Bailey and C. Paar, "Optimal extension fields for fast arithmetic in public key algorithms", Advances in cryptography – CRYPTO '98, Lectures in computer science(LNCS), Vol. 1462, pp. 472 – 485, Springer-Verlag, Aug 1998.

[57] Daikpor, M.N and O. Adegbenro; " Radix – 7 Signed Digit Element finite field Arithmetic"; Proc of the 2007 IEEE International Conference on Signal Processing and Communication (ICSPC07) 24-27 November 2007, Dubai UAE.

[58] Krisin Cordwell, Cheng Zhau, Stephen Scum, William Cordivett."Finding inverses in a finite field", New Mexico supercomputing Challenge . Final Report April 2005.

[59] Katsuki Kobayasi, Naofumi Takagi, and Kazuyoshi Takagi, ""Algorithm for inversion in suitable for implementation using a polynomial multiply instruction on ". Department of Information Engineering, Graduate School of Information Science , Nagoya University, Furo-cho, Chkusa-ku, Nagoya, 464-8603 Japan. www.computer.org/portal/web/usdl/101/10/101109/Arith. 2007.9

[60] Itoh T. and Tsuji,"A fast algorithm for computing multiplicative inverse in GF(2m) normal basis". Information Computing Journal, 78; pp. 171 -177, 1988.

[61] White paper; "Elliptic curve cryptography", Thales e-Security, INC white paper 2200 N Commerce parkway Western Florida 33326 USA. http://www.thales-esurity.com

[62] T. Shaska, "Introduction to Cryptography- Lecture Notes". Webpage http//www.oakland.edu/~shaka. 2008.

[63] Zhaohui Cheng. "Simple tutorial on Elliptic curve cryptography",2004. . http://www.eis.mdx.ac.uk/staffpages/m-cheng/link/ecc-simple.pdf.

[64] Joe Hurd, "Course notes on Elliptic curve cryptography A case study in formulization using a higher order logic theorem prover" Oxford University August 2005.

[65] T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithm" IEEE Trans, Information Theory, Vol.IT-31, 1985. pp 469 – 472.

[66] Joseph H. Silverman, "An introduction to the theory of Elliptic Curves". Summer School on Number Theory and Applications to Cryptography; University of Wyoming, June –July 2006. http://www.math.brown.edu/~jhs/presentation/wyomingellipticcurve'pdf

[67] Avikak "Lecture notes on 'Introduction to computer security". Avinagh Kak Purdue University 2010. http://cobweb.ecn.purdue.edu/~kak/compsec/newLectures/lecture14.pdf.

[68] Neal Kobllitz, Alfred Menezes and Scott Vanstone, "State of Elliptic Curve Cryptography", Design Codes and Cryptography, Kluwer Academic Publishers, Boston. N0 19, 2000 pp 173 – 193.

[69] Aleksander Jariska and Alfred J. Menezes "Elliptic curve and cryptography ". Certicom Corp (Canada), juristisic@certicom.com. http://www.math.uwaterloo.ca/àjmeneze/research 2009.

[70] Igbal H. Jubril, Rus Salleh and Al-Shawabkeh M. "Efficient algorithm in projective coordinates for ECC over " Internal Journal of Computer, internet and management. Vol. 15 N0 1 April 2007 pp 43 – 50.

[71] "Lecture notes 1: Affine Planes", http://w.w.w-math.cudener.edu/~wcherowi/courses/m4220/hg21cc1,4,5.htm 2007

[72] Cetin Kaya Koc "Elliptic curve cryptosystem" Dept of Electrical and Computer engineering, Oregon State University, Oregon 97331, 2004.

[73] Nicolas Meloni " fast and secure EC Scalar multiplication over prime fields using special Addition chains"; Institute de Mathematique et de modisatic de Montpeller, UmR 5149, Montpeller France. Citeseerx.ist.psu.edu/viewdoc/download?doi=10.61.5861[1] pdf.

[74] Brain King, "Implementation issues when using Elliptic curve Cryptography", Security Technology laboratory, Motorola Labs. 2006. http://www.iccip.csl.uiuc.edu/conf/ceps/2000/king.pdf.

[75] J.C Bajard, L. Imbert, C Negre and T. plan tard, "Efficient multiplication in GF(P^k) for elliptic curve cryptography Laboratoire de' Proceedings of the 16th IEEE Symposium on computer Arithmetic. Informatique de Robotic et de Micro electronique de Montpellier LIRMM, 161 rue Ade 34392 Montpellier adex 5 France. 2003, pp181 - 187.

[76] A. Murat Fiskiran and Ruby B. Lee, " Workload Characterization of Elliptic Curve Cryptography and other Network Security Algorithms for Constrained

Environments". Proc. of the 5th International workshop on Workload Characterization WWC-5 2002 pp 127-137.

[77] Yeh; C.S. Irving S. Reed and Troug "Systolic multipliers for finite fields GF(22)". IEEE Transactions on Computers, vol. c-33, N0. 4, pp 357- 360 APRIL 1984

[78] C.C Wang " An algorithm to Design Finite field Multipliers Using a self-dual Normal Basis". IEEE Trans. on computers, vol. 38, N0. 10 pp 1457- 1460 Oct. 1989.

[79] B. Sunar and C.K. Koc "An efficient Optimal Normal Basis Type II Multiplier" IEEE Transactions on Computers 50(1):83 -87, January 2001.

[80] Masakatu Morii and Masao Kasahara Efficient Construction of gate circuit for computing Multiplicative Inverses over GF(22)", .IEEE Transaction on IEICE, vol. E72, N0. 1 pp 37 – 42, January 1989.

[81] G.L.Feng 1383 "A VLSI Architecture for fast Inversion in ",IEEE Transactions on computers, vol. 38 N0. 10 pp 1383 – 1386, Oct. 1989.

[82] Johann Grobschadl and Guy-Armand Kamendje "" Low-power design of a function unit for arithmetic in finite field and ", WISA 2003, LNCS 2908, pp. 227 – 242, 2003.

[83] B. Hochet, P. Quinton and Y. Robert, "Systolic Gaussian Elimination over with partial pivoting", IEEE transactions on computers Vol. 38 N0 9, 1998.pp1321 – 1332.

[84] Martin Furer and Kurt Mehlhorn , " AT^2 - Optimal Galois Field Multiplier for VLSI" , IEEE transactions on computers vol. 38 N09, 1998, pp 1333 – 1336.

[85] Johann Grobschadl "Ä low-power bit-serial multiplication for finite Fields GF(22)". Proceedings of the 34th IEEE International Symposium on circuits and systems (ISCA 2001), vol. IV, pp 37-40.

[86] M. Jung, F. Madlener, M. Ernst and S.A Huss "Reconfigurable Coprocessor for finite field multiplication in " Integrated circuits and systems lab. Computer science department, Darmstadt University of Technology, Germany, 2002.

[87] M. Bednara, M. Daldrup, J. Von zur Gathen, J. Schokrollahi, J. Teich "Reconfigurable implementation of elliptic curve crypto algorithm" University of Paderborn, Paderborn, Germany, 2003.

[88] Jorge Guajardo, Bainer Bliimel, Uwe Krieger, and Christof Paar "Efficient Implementation of Elliptic curve Cryptosystems on the T1 MSP430x33x family of microcontrollers", In K. Kim (Ed.) : PKC 2001, LNCS 1992, pp. 365 – 382, Korea, February 2001.

[89] Dusanka Bundalo, Zlatko Bundalo and Branimir Dordevic; " Multiple-Valued regenerative CMOS Logic Circuits with High-Impedance Output State"; FACTA UNIVERSITATIS (NIS) Ser.: ElEC., ENERG. Vol. 18, No. 1 , April 2005, pp 45 – 56.

[90] Masamich Akazawa, Kentarou Kanaami, Takachi Yamada and Yosihito Amemiya; "Multiple-Valued Inverter Using Single-Electron- Tunneling Circuit"; IEICE. Trans . Electron., Vol. E82-C, N0 9 September 1999. pp. 1607 – 1613

[91] Akira Mochizuki and Takahiro Hanyu;" Low-power Multiple-Valued Current-mode Logic Using substrate bias Control"; IEICE. Trans . Electron., Vol. E87-C, N0 4 April 2004. pp. 582 – 588.

[92] Alejandro F. Gonzalez, and Pinaki Mazumder.;"Multiple-Valued Signed-digit Adder Using negative Differential-Resistance Devices". IEEE Transactions on Computers, Vol. 47, N0 9, September 1999.

[93] Sangmi Shim, Seungwoo Park and Seunghong Hong;" Design of Advanced multiple-valued D-FF Using Neuron-MOS"; UCSNS International journal odf Computer Science and Network Security , Vol., 6 No. 9B September 2006. pp 118 – 122.

[94] Elena Dubrova;" Multiple-valued Logic in VLSI; Challenges and opportunities"; Electronic system Design Laboratory; Department of Electronics, Royal institute of Technology , Kista, Sweden 2004; http://citeseerx.1st.psu.edu/viewdoc/summary?doi=10.1.1.8.6751

[95] Hiromitsu Kimura, Takahiro Hanyu and Michikata Kameyama, "Multiple-Valued Logic-n-Memory VLSI Using MFSFET and Its Applications" Research School of Information Science, Tohoku University Japan April 2002, Pp 23 – 42.

[96] Ali Sheikholeslami, Ryuji Yoshimura and P. Glenn Gulak;" Look-Up Tables for Multiple-valued, combinational logic". Department of Electrical and Computer Engineering , University of Toronto, Ontario Canada. 2004. Pp 1-6.

[97] Nareli Cruz-Cortes, Francisco Rodriguez-Henriquez, Raul Juarez-Morales and Carlos A. Coello; "Finding optimal Additions Chains Using a generic Algorithm Approach", Y. Hao et al (Eds.), CIS 2005, Part I LNAI 3801, pp. 208-215, 2005.

[98] Daniel J. Bernstein, "Differential addition chains" djb@cr.yp.to Danmarks Tekniske Universitet, Febuary 2006. http://cr.yp.to/ecdh/difchain.2006019.pdf.

[99] Nobororu Kunihiro and Hirosuke Yamamoto, "New methods for generating short addition chains", IEICE Trans. Fundamentals, Vol., E83-A, N01,January 2000, pp 60 - 66.

[100] Christophe Claver and Marce Joye ,"Universal Exponentiation Algorithm- A first step towards provable SPA-resistance", C.K KOC, D. Naccche, and C. Paar, Eds., Crptographic Hardware and embedded systems – CHES 2001, vol. 2162 of lecture Notes in computer science, pp. 300 – 308, spring-Verlag, 2001

[101] Oakley, C.O "Analytic Geometry" Barnes and Noble New York, College Out Line series 61.50 N0 68 1957.

[102] T. Vineswaran, B. Mukundhan, and P. Subbarami Reddy;" A novel Low Power, High Speed 14 Transistor CMOS Full Adder Cell with 50% Improvement in Threshold Loss Problem"; World Enformatika Society, Enformatika V13 2006 ISSN 1305 – 5313 Pp 81 - 85.

[103] B. Parhami, "Implementation alternatives for generalized signed-digit addition," in Proceedings of IEEE 28th Asilomar Conference on signals, Systems, and Computers, 1994, pp. 157-161.

[104] Israel Koren "Digital computer arithmetic, ECE 666 Part 6a high-speed multiplication – I", University of Massachusetts, Dept of Electrical & computer Engineering. Spring 2004.

[105] Israel Koren "Digital computer arithmetic, ECE 666 Part 6b high-speed multiplication – I", University of Massachusetts, Dept of Electrical & computer Engineering. Spring 2004.

[106] Israel Koren "Digital computer arithmetic, ECE 666 Part 6c high-speed multiplication – I", University of Massachusetts, Dept of Electrical & computer Engineering. Spring 2004.

http://www.ecs.umass.edu/ece/koren/arith/slides/Part6c-mlt.ppt

[107] Paul F. Charles Martel, Vojin G. Oklobdzija and R. Tavi. "Optimal circuits for parallel multipliers", IEEE transactions on Computers, VOl. 47, N0 3, March 1998.

[108] L. Dadda "Some schemes for parallel multipliers. "Coloque sur lÁlgebre de Boole, Grenobie, 11.17 January 1965. Copyright1965 by Associazione Elettrotecnica ed. Electtronica italiana, pp 349 – 356.

[109] Marcelo Fonseca, Edardo da Costa, Sergio Bampi and Jose Monteiro. "Design of a radix – 2^m Hybrid Array Multiplier Using carry save adder"; UCPei, Pelotas, Brazil mrf,ecosta@ucpel.tche.br 2006.
http://ieeexplore.ieee.org/iel5/4286806/4286807/04286852.pdf

[110] Reto Zimmermann. " Lecture notes on Computer Arithmetic : Principles, Architectures, and VLSI design"; Integrated Systems Laboratory Swiss Federal Institute of Technology (ETH) CH-8092 Zurich, Switzerland. March 1999 zimmermann@iis.ee.ethz.ch.

CHAPTER 7

APPENDICES

Basic $SGF(7)$ arithmetic operations Appendix CH3.1

i) Addition $a_i + b_i$

	-3	-2	-1	0	1	2	3
-3	1	2	3	-3	-2	-1	0
-2	2	3	-3	-2	-1	0	1
-1	3	-3	-2	-1	0	1	2
0	-3	-2	-1	0	1	2	3
1	-2	-1	0	1	2	3	-3
2	-1	0	1	2	3	-3	-2
3	0	1	2	3	-3	-2	-1

ii) Subtraction $a_i - b_i$

	-3	-2	-1	0	1	2	3
-3	0	1	2	3	-3	-2	-1
-2	1	0	1	2	3	-3	-2
-1	-2	-1	0	1	2	3	-3
0	-3	-2	-1	0	1	2	3
1	3	-3	-2	-1	0	1	2
2	2	3	-3	-2	-1	0	-1
3	1	2	3	-3	-2	-1	0

iii) Multiplication $a_i * b_i$

	-3	-2	-1	0	1	2	3
-3	2	-1	3	0	-3	1	-2
-2	-1	-3	2	0	-2	3	1
-1	3	2	1	0	-1	-2	-3
0	0	0	0	0	0	0	0
1	-3	-2	-1	0	1	2	3
2	1	3	-2	0	2	-3	-1
3	-2	1	-3	0	3	-1	2

iv) Division $a_i / b_i = a_i * b_i^{-1}$

	-3	-2	-1	0	1	2	3
-3	1	-2	3	0	-3	2	-1
-2	3	1	2	0	-2	-1	-3
-1	-2	-3	1	0	-1	3	2
0	0	0	0	0	0	0	0
1	2	3	-1	0	1	-3	-2
2	-3	-1	-2	0	2	1	3
3	-1	2	-3	0	3	-2	1

v) Multiplicative inverse

a	1	2	3	-3	-2	-1
a^{-1}	1	-3	-2	2	3	-1

Appendix CH3.2 **Table of special primes**

Some 134 special primes H, of the sequence $H \in \{q^m + \psi_i\}$ where $\psi_i = j\alpha\beta; j\{01,2,3...\}$, $\alpha + \beta = q$, $\alpha.\beta = 2q$ and $q = 7$ for elliptic curves defined over the RR7SQSD elements finite fields				
7	13	19	31	37
43	61	67	73	79
97	103	109	127	139
151	157	163	181	193
199	211	223	229	241
271	277	283	307	313
331	337	349	367	373
379	397	409	439	463
487	499	523	547	571
607	619	631	643	691
2053	2089	2113	2137	2161
2221	2269	2281	2293	2341
2377	2389	2437	2473	2521
2557	2593	2617	2677	2689
2713	2749	16447	16519	16567
16603	16651	16699	16747	16759
16831	16843	16879	16903	16927
16963	16987	17011	17047	17107
17167	117361	117373	117517	117529
117541	117577	117673	117709	117721
117757	117841	117877	117889	117937
117973	823183	823219	823231	823243
823351	823399	823447	823483	823519
823591	823651	823663	823723	823747
823759	823819	823831	823843	832903
5764501	5764609	5764621	5764741	5764873
5764897	5764909	5765041	5765077	

Appendix CH3.3 RR7SQSD number system based big integer multiplication

```
2 CLS : REM USE OF THE RR7SQSD MULTIPLY-BY-7 AADITION/SUBTRACTION CHAIN
3 REM FOR BIG INTEGER MULTIPLICATION
4 LPRINT : LPRINT "  Appendix CH 3.3                                    QB 4.5 CODE"
6 LPRINT "_____________________________________________________________";
7 LPRINT "_____________________________": PRINT
8 LPRINT : LPRINT "      THIS PROGRAM CONVERTS A 160 DECIMAL DIGIT NUMBER ";
9 LPRINT "TO RR7SQSD"
10  LPRINT "          AND  AUTOMATICALLY COMPUTES ITS SQUARE"
14 DIM XX7F(80), FR(193), XS0(80), GS(80), LST(80), GF(80), MV(80), FSLT(80)
15 DIM B1(80), B2(80), CVH(80), D(80), D0(80), D1(80), D2(80), E(192)
16 DIM LMDA(192), Q(192, 80), R(192, 80), X0(80), X(192, 80), XX1(80)
17 DIM XXA7F(80), XS(80), XX(80), XXF(80), XXB7F(80), Z(193), ZP(193), ZF(193)
18  DATA 320,6,10,7,3: F = 0
19  READ N, TC, RN, RQ, AD
20  REM F = F + 1: READ GAG: IF GAG = 7777 THEN PRINT F: STOP
22 REM GOTO 20
26  DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000
28 DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000
30 DATA 0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000
32  DATA 1856,2115,9210,1757,4302,4531,6367,1207,7895,2313,7945,8679,4030
34 DATA 4589,6354,4756,8523, 9546,8671,7417,7532,1596,7894,4562,3216,4756
36 DATA 2346,7894,4562,7417,3216,1596,7895,4896,5643,1459,2587,3694,1002,7777
40 TP = TC - 2: M1 = INT(N / TP): IC = 0
42 LMDA(0) = 0: D(0) = 0: R(0, 0) = 0: Q(0, 0) = 0
44 LPRINT : LPRINT " _______________ BELOW IS THE VALUE OF THE OPERAND  ___________"
48 FOR TAU = 1 TO M1:  REM READING FROM THE MOST SIGNIFICANT PARTITION
52 READ X0(TAU): XS0(TAU) = X0(TAU): X(IC, TAU) = X0(TAU): Q(IC, TAU) = X0(TAU):
53 REM PRINT X(IC, M1 + TAU), Q(IC, M1 + TAU): NEXT TAU: PRINT : STOP: PRINT
54  LPRINT Q(IC, TAU); : NEXT TAU: LPRINT : : P = 1: REM STOP
56 REM NEXT TAU: P = 1
63 REM PRINT : PRINT "          THEY SHALL BE TAKEN IN THIS ORDER          ": REM PRINT
64 LPRINT : REM RESTRUCTURING THE INITIAL DATA FOR COMPUTATIONAL ACCURACY
68 FOR TY = M1 TO 1 STEP -1: XS(P) = XS0(TY):  P = P + 1: NEXT TY: REM   STOP
70  REM GENERATING THE RADIX 7 DIGITS/START OF CONVERSION PROCESS
72 LPRINT :  LPRINT " _____________ GENERATING THE RADIX-7 DIGITS _______________"
: PRINT : REM STOP
80 LPRINT :  R(IC, 0) = 0
90 IC = IC + 1
100 FOR JC = 1 TO M1
104 fg = (R(IC, JC - 1) * RN ^ TP + Q(IC - 1, JC)): REM PRINT fg
110 R(IC, JC) = fg MOD RQ:  REM PRINT R(IC, JC); : REM STOP
```

Appendix CH3.3 RR7SQSD number system based big integer multiplication

```
 158 Q(IC, JC) = INT(fg / RQ):  REM PRINT Q(IC, JC); : REM PRINT fg, q(ic, jc)
 170  NEXT JC:   FR(IC) = R(IC, JC - 1):  REM PRINT :
 172 REM PRINT "FINAL RADIX 7 DIGIT FOR IC="; IC; "IS FR(C)="; FR(IC):  REM STOP
 182 REM IF IC >= 2 * M1 + 4 OR Q(IC, JC - 1) <= 0 THEN 190
 183 IF Q(IC, JC - 1) <= 0 THEN 190
 185  FOR PR = 1 TO M1: X(IC, PR) = Q(IC, PR): NEXT PR:  GOTO 90:
 186 REM  PRINT X(IC, PR); : NEXT PR: PRINT : PRINT : GOTO 90
 190 REM PRINT : PRINT "    THE RADIX 7 DIGIT OBTAINED WHEN IC ="; IC; "IS"; FR(IC): REM STOP
 200 REM PRINT "        TOTAL NUMBER OF RADIX 7 DIGITS OBTAINED IS ="; IC: REM  STOP
 210 REM PRINT "     LISTING THE RADIX 7 DIGIT THUS OBTAINED FROM THE LEAST ";
 212 REM SIGNIFICANT DIGIT": REM STOP
 300 REM PRINT : FOR KC = 1 TO IC: Z(KC) = FR(KC): PRINT FR(KC); : NEXT KC:STOP: PRINT
 302 REM PRINT " LISTING THE RADIX 7 DIGIT THUS OBTAINED FROM THE MOST SIGNIFICANT DIGIT"
 308  FOR KC = IC TO 1 STEP -1: Z(KC) = FR(KC): LPRINT FR(KC); : NEXT KC: REM  STOP: PRINT :
 400 REM START OF THE CONVERSION TO RR7SQSD MODULE
 402  LPRINT " _____ CONVERTING TO THE RR7SQSD DOMAIN _____";
 403 LPRINT "_____":  REM STOP
 406 LMDA(0) = 0: LPRINT
 410 IF Z(VC) = AD AND LMDA(VC - 1) = 1 THEN LMDA(VC - 1) = -LMDA(VC - 1):
  LMDA(VC) = 1: ZP(VC) = -(Z(VC) - 1):  GOTO 430
 414 IF Z(VC) > AD THEN LMDA(VC) = 1: GOTO 420
 416 LMDA(VC) = 0
 420 ZP(VC) = Z(VC) - RQ * LMDA(VC): REM PRINT ZP(VC);
 430 ZF(VC) = ZP(VC) + LMDA(VC - 1)
 440 NEXT VC: LPRINT
 450 REM PRINT " _____RADIX-7 DIGITS OBTAINED _____ FROM THE MSD."
 460 REM FOR V1C = IC TO 1 STEP -1: PRINT Z(V1C); : NEXT V1C: REM PRINT : REM RADIX 7 DIGIT ARRAY
 476 REM PRINT "   RESTRICTED RADIX-7 SYMMETRICAL QUATERNARY SIGNED-DIGITS OBTAINED.."
 480 FOR LC = IC TO 1 STEP -1: LPRINT ZF(LC); : E(LC) = ZF(LC): NEXT LC:  REM RR7SQSD DIGITS ARRAY
 482 REM PRINT : REM PRINT "_____": PRINT :  STOP
 500 REM pointmulti2
 520  LPRINT " _____NOW PERORMING RR7SQSD MULTIPLICATION _____": REM STOP: PRINT
 530 REM PRINT : PRINT "_____ ONCE AGAIN YOUR INITIAL VALUES ARE _____"
 540 FOR R0P = 1 TO M1: GS(R0P) = XS(R0P): NEXT R0P: REM PRINT XS(R0P);
 542  REM NEXT R0P: PRINT : REM STOP: PRINT
 550 REM PRINT " _____INITIAL DATA RADIX 7 EQUIVALENT_____"
 556 REM FOR V1P = IC TO 1 STEP -1: PRINT Z(V1P); : NEXT V1P:  PRINT : REM RADIX 7 DIGIT ARRAY
 570 REM PRINT ".....AND THE RR7SQSD EQUIVALENT .....":PRINT
 574 REM FOR LP = IC TO 1 STEP -1: PRINT E(LP); : NEXT LP:  REM RR7SQSD DIGITS ARRAY

 578 REM PRINT : REM  PRINT

580  REM PRINT " .....BELOW COMPUTES THE VALUES OF Q.....":
REM COMPUTING THE START Q = F(K)*P
```

Appendix CH3.3 RR7SQSD number system based big integer multiplication

```
582 FOR J0P = 1 TO M1: GS(J0P) = XS(J0P): NEXT J0P: REM PRINT XS(J0P);
583 REM NEXT J0P: PRINT : PRINT : REM STOP
588 KP = IC: G = KP: D0(0) = 0
590 IF E(KP) = 1 THEN ST = 1: GOTO 620
600 IF E(KP) = 2 THEN ST = 2: GOTO 620
610 IF E(KP) = 3 THEN ST = 3: GOTO 620
612 IF E(KP) = 0 THEN ST = 0
620 D0(0) = 0:  REM PRINT "WWEEEEEEE"; ST: REM STOP:
630 FOR J1P = 1 TO M1
640 XX(J1P) = GS(J1P) * ST + D0(J1P - 1)
670 IF XX(J1P) > 9999 THEN D0(J1P) = INT(XX(J1P) / RN ^ TP): GOTO 690:
672 REM WRAPPING INTO WORD LENGTH
680 D0(J1P) = 0
690 XX1(J1P) = XX(J1P) - D0(J1P) * RN ^ TP
700 XXF(J1P) = XX1(J1P): REM PRINT XXF(J1P);
710 NEXT J1P: REM STOP:
720  REM PRINT "....HERE BELOW IS MULTPLICATION BY 7 USING 'NAF'....."
730 FOR IP = IC - 1 TO 1 STEP -1: REM THIS IS WHERE THE RRSQSDs ARE PICKED
740 G = IP
770 FOR TT = 1 TO 3
776 D1(0) = 0:
810 FOR J2P = 1 TO M1
820 XX7F(J2P) = XXF(J2P) + XXF(J2P) + D1(J2P - 1)
830 IF XX7F(J2P) > 9999 THEN D1(J2P) = 1: XX7F(J2P) = XX7F(J2P) MOD RN ^ TP: GOTO 850
840 D1(J2P) = 0
850 REM PRINT XX7F(J2P); : XXF(J2P) = XX7F(J2P): XXB7F(J2P) = XX7F(J2P): NEXT J2P:  PRINT
852 XXF(J2P) = XX7F(J2P): XXB7F(J2P) = XX7F(J2P): NEXT J2P:
900 NEXT TT: REM PRINT :  REM STOP
910 REM COMPUTING THE 'NAF' FINAL STEP OF THE MULTIPLICATION PROCEEDURE USING 'NAF'
920 B1(0) = 0: REM PRINT ".....HERE BELOW IS OPERATION 8P-P .....
921 REM FOR IC="; IP: REM
922 REM PRINT : FOR TKP = 1 TO M1: PRINT XXB7F(TKP); : NEXT TKP:
: PRINT : PRINT "......CURRENT Q OR 8P.....": PRINT : REM STOP
924 REM FOR G3P = 1 TO M1: PRINT XX1(G3P); : NEXT G3P: PRINT :
925 REM PRINT ".....PREVIOUS Q.....": REM  STOP
928   REM PRINT
930 FOR J3P = 1 TO M1
940  FTR = XXB7F(J3P) + B1(J3P - 1)
942 IF FTR < XX1(J3P) THEN B1(J3P) = -1: CVH(J3P) = (RN ^ TP + FTR) - XX1(J3P): GOTO 960:
943 REM PRINT CVH(J3P); : GOTO 960
950 B1(J3P) = 0
952 CVH(J3P) = FTR - XX1(J3P): REM PRINT CVH(J3P);
960 NEXT J3P: REM STOP
```

Appendix CH3.3 RR7SQSD number system based big integer multiplication

```
990  REM PRINT "----HERE BELOW IS REMOVING RR7SQSDs OF------FOR I="; "  "; E(IP): REM STOP
1000 IF E(IP) = 0 THEN LPH = 0: GOTO 1070
1010 IF E(IP) = 1 THEN LPH = 1: GOTO 1070
1020 IF E(IP) = -1 THEN LPH = -1:  GOTO 1070
1030 IF E(IP) = 2 THEN LPH = 2: GOTO 1070
1040 IF E(IP) = -2 THEN LPH = -2: GOTO 1070
1050 IF E(IP) = 3 THEN LPH = 3: GOTO 1070
1060 IF E(IP) = -3 THEN LPH = -3
1070 B2(0) = 0: MV(0) = 0
1071 IF LPH < 0 GOTO 1080
1072 FOR ZP = 1 TO M1: GF(ZP) = GS(ZP) * LPH + MV(ZP - 1)
1073 IF GF(ZP) > 9999 THEN MV(ZP) = INT(GF(ZP) / RN ^ TP): GOTO 1075
: REM WRAPPING INTO WORD LENGTH
1074 MV(ZP) = 0
1075 GF(ZP) = GF(ZP) - MV(ZP) * RN ^ TP
1076 NEXT ZP: GOTO 1122: REM PRINT GF(ZP); : NEXT ZP: GOTO 1122
1080 FOR ZN = 1 TO M1: GF(ZN) = GS(ZN) * LPH + MV(ZN - 1)
1090 IF ABS(GF(ZN)) > 9999 THEN MV(ZN) = -INT(ABS(GF(ZN)) / RN ^ TP): GOTO 1093
1092 MV(ZN) = 0
1093 GF(ZN) = -(ABS(GF(ZN)) MOD RN ^ TP)
1094 NEXT ZN: GOTO 1122: REM PRINT GF(ZN); : NEXT ZN:  GOTO 1122
1100 REM IF GF(Z) > 0 THEN GF(Z) = GF(Z) MOD RN ^ TP
: MV(Z) = INT(GF(Z) / RN ^ TP): GOTO 1120
1110 REM MV(Z) = -INT(GF(Z)): GF(Z) = -(ABS(GF(Z)) MOD RN ^ TP)
1120 REM PRINT GF(Z); : NEXT Z:
1122 B2(0) = 0:  REM PRINT "FOR WHEN LPH= "; LPH: B2(0) = 0: REM PRINT
1130  FOR J4 = 1 TO M1
1140 IF CVH(J4) < ABS(GF(J4) + B2(J4 - 1)) AND (GF(J4) + B2(J4 - 1)) < 0 THEN B2(J4) = -1:
 FSLT(J4) = CVH(J4) + RN ^ TP + GF(J4) + B2(J4 - 1): GOTO 1160
1142 IF GF(J4) + CVH(J4) + B2(J4 - 1) > 9999 THEN FSLT(J4) = (CVH(J4) + GF(J4) + B2(J4 - 1)) MOD RN ^ TP
: B2(J4) = 1: GOTO 1160
1150 FSLT(J4) = CVH(J4) + GF(J4) + B2(J4 - 1): B2(J4) = 0
1160 XXF(J4) = FSLT(J4): XX1(J4) = FSLT(J4): NEXT J4: REM PRINT FSLT(J4); : NEXT J4: PRINT : PRINT
1164 REM XS(J4) = FSLT(J4): XXF(J4) = FSLT(J4): XX1(J4) = FSLT(J4): PRINT FSLT(J4);
: NEXT J4: PRINT : PRINT
1170 IF IP <> 1 THEN 3090
1172 FOR J5 = M1 TO 1 STEP -1: LPRINT FSLT(J5); : NEXT J5: LPRINT
3080  REM PRINT : PRINT "  HENCE END OF THE ..."; G; " TH ITERATION OF THE POINT MULTIPLICATION
... ": PRINT : REM STOP: PRINT
3082  LPRINT : LPRINT " _____ END OF THE LAST ITERATION _____  "
: PRINT : REM STOP: PRINT
3090  NEXT IP
3092 LPRINT "_____": PRINT : REM STOP

3094 LPRINT "    QB 4.5 CODE                                    "
3500 END
```

Appendix CH 3.4 PLAIN TEXT MESSAGE EMBEDMENT

```
10 CLS : PRINT : PRINT "    Appendix  CH 3.4                                    QB 4.5 CODE    ": PRINT
12 PRINT " ____________________________________________________________________": REM STOP
14 PRINT : PRINT : REM THIS PROGRAM GENERATOR THE CURVE POINTS AND PERFORM MESSAGE
EMBEDDMENT
20 DIM CD(23), XCBP(22), YCBP(22)
50 ZX$ = "    'THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG'"
60  PRINT "     THE PLAN TEXT MESSAGE TO BE SENT IS "; ZX$: PRINT
100 PRINT "        THE CORRESPONDING NUMERIC CODE IS": PRINT
110 DATA 1507,2704,2625,1201,527,2006,1827,2205,2715,1619,2113,2710,1524,2706,2314
120 DATA 1815,2702,311,2109,2717,605,20,99: LN = 1
130 READ CD(LN): IF CD(LN) <> 99 THEN PRINT CD(LN), : LN = LN + 1: GOTO 130
140 LN = LN - 1: REM STOP: REM 99 IS END OF MESSAGE
150 DATA -53,218,17011
160 READ AE, BE, P: PRINT "AND THE ELLIPTIC CURVE CONSTANTS ARE"; AE; BE; P; : PRINT
170 PTR = 1: PRINT : PRINT "WHILE THE RATIONAL CURVE POINTS ARE": PRINT : PTR = 1: GOSUB 360
180 PRINT : PRINT "            THE ORDER OF FIELD =    "; INT(DORR):    REM STOP
182 PRINT " ____________________________________________________________________"
: REM STOP
184 PRINT : PRINT "   THE PLAN NUMERIC TEXT EMBEDDED INTO THE ELLPTIC CURVE
GROUP OF POINTS IS  "
186 PRINT : PRINT " S/N0  RAW MESSAGE     CURVE POINTS EMBEDDMENT"
190  PRINT : PTR = 2: GOSUB 600
192 PRINT " ____________________________________________________________________"
: REM STOP
194  PRINT : PRINT "    QB 4.5 CODE                                    "
200 END
360 REM
362 XD = -1: DORR = P + 1: REM QUADRATIC RESIDUE AND CURVE POINTS COMPUTATION MODULE
:REM BQ=1
364 XD = XD + 1: IF XD > P - 1 THEN 456
366 YS = (((XD ^ 2) MOD P) * XD - 53 * XD + 218) MOD P:
367 IF YS < 0 THEN YS = YS + P
368 SLK = 0: GOSUB 400:
369 IF SLK <> 1 THEN PRINT USING "    ##### ##### "; XD; QE1;
370 GOTO 364
372 RETURN
400 REM
404 DSQ = (YS ^ 2) MOD P:
405 DCB = (DSQ * YS) MOD P:
406 DQD = (DSQ ^ 2) MOD P
408 DCT = (DQD ^ 2) MOD P
409 D12 = (DCT * DQD) MOD P
410 D60 = (((((D12 ^ 2) MOD P) ^ 2) MOD P) * D12) MOD P
412 D68 = (D60 * DCT) MOD P:
414 D5 = (((((D68 ^ 2) MOD P) ^ 2) MOD P) * D68) MOD P
```

Appendix CH 3.4 Plain text message embedment

```
416 DD5 = (((((D5 ^ 2) MOD P) ^ 2) MOD P) * D5) MOD P
418 DDD5 = (((((DD5 ^ 2) MOD P) ^ 2) MOD P) * DD5) MOD P:
420 DEX = (DSQ * DCB) MOD P:

422 PHJ = (DEX * DDD5) MOD P: REM   PRINT PHJ;
424 IF PHJ <> 1 THEN SLK = 1: GOTO 456
432 DORR = DORR + YS / P
433 VT2 = (YS ^ 2) MOD P
434 VT4 = (((YS ^ 2) MOD P) ^ 2) MOD P:
436 VT5 = (((((YS ^ 2) MOD P) ^ 2) MOD P) * YS) MOD P
440 VT30 = (((((((VT5 ^ 2) MOD P) ^ 2) MOD P) * VT5) MOD P) * VT5) MOD P
441 VT34 = (VT30 * VT4) MOD P
444 VTT1 = (((((VT34 ^ 2) MOD P) ^ 2) MOD P) * VT34) MOD P:
446 VTT2 = (((((VTT1 ^ 2) MOD P) ^ 2) MOD P) * VTT1) MOD P:
448 VTT3 = (((((VTT2 ^ 2) MOD P) ^ 2) MOD P) * VTT2) MOD P
450 QE1 = (VTT3 * VT2) MOD P
456 RETURN

600 REM  EMBEDDING THEN PLAN TEXT MESSAGE INTO THE ELLIPTIC CURVE GROUP
602 KP = -1: KQ = 1: GASO = CD(KQ): CV = 0
604 KP = KP + 1:
606 D1 = ((KP * ((KP ^ 2) MOD P)) MOD P - 53 * KP + 218) MOD P:
607 IF D1 < 0 THEN D1 = D1 + P
608 YS = D1: SLK = 0: PTR = 2: GOSUB 400: BR = 0:
610 IF SLK = 1 THEN 604
612 IF KP = CD(KQ) THEN CV = CV + 1: XCBP(CV) = KP: YCBP(CV) = QE1:
PRINT USING "  ##    #####          ##### #####"; KQ; CD(KQ); XCBP(CV); YCBP(CV): GOTO 630
614 IF KP < P - 1 THEN 604:
616 BR = CV:
618 GASO = GASO + 1: GASO = GASO MOD P:
620 IF GASO = XCBP(BR) THEN 618:
622 BR = BR - 1: IF BR > 0 THEN 620
624 KP = GASO: D1 = ((KP * ((KP ^ 2) MOD P)) MOD P - 53 * KP + 218) MOD P: IF D1 < 0
THEN D1 = D1 + P
626 YS = D1: SLK = 0:  GOSUB 400:
627 IF SLK <> 1 THEN CV = CV + 1: XCBP(CV) = KP: YCBP(CV) = QE1:
PRINT USING "  ##    #####          ##### #####"; KQ; CD(KQ); XCBP(CV); YCBP(CV): GOTO 630
628  GOTO 616
630 KQ = KQ + 1
632 IF KQ > LN THEN 636
634  GASO = CD(KQ): KP = -1: BR = 0: GOTO 604: REM STOP
636  RETURN
1000 REM TO FIND THE MULTIPLICATIVE INVERSES
```

Appendix CH3.4 **Plain text message embedment**

```
1010 DIM T(20)
1020 IF BINV = 1 THEN LS = 1: GOTO 1150
1030 LA = AST: LB = BINV
1040 AA = LA: LK = 0: LC = 1: LD = 0: LE = 0: LF = 1
1050 LK = LK + 1
1060 T(LK) = INT(LA / LB): U = LA - LB * T(LK)
1070 IF ABS(U) = 1 OR U = 0 THEN 1090
1080 LA = LB: LB = U: GOTO 1050
1090 FOR LJ = 1 TO LK
1100 LG = LE * T(LJ): LH = LF * T(LJ)
1110 LL = LC - LG: LS = LD – LH
1120 LC = LE: LD = LF: LE = LL: LF = LS
1130 NEXT LJ
1140 IF LS < 0 THEN LS = P + LS
1150 INV = LS: XC = LS
1160 RETURN
```

Appendix CH3.5 RR7SQSD FINITE FIELD ELEMENT INVERSE COMPUTATION

```
1 CLS : REM THIS PROGRAM COMPUTE MULTIPLICATIVE INVERSE USING EXTENDED
2 REM EUCLIDEAN ALGORITHM. 241,691,17011,117721,5764873 ARE GIVEN AS
3 PRINT : PRINT "    Appendix  CH 3.5                                    QB 4.5
CODE": REM STOP
5 PRITN: PRINT " _____________________________________________________________": PRINT :
7 DIM T(30)
10 DATA 37,79,241,691,17011,17107,117721: REM ,5764873
12 PRINT "LHS IS THE NUMBER AND THE RHS IS THE MULTIPLICATIVE INVERSE"
14 FOR YT2 = 1 TO 7
15 READ AST: REM AST = 17011: REM INPUT "MODULO "; AST
16 XX = AST: LV = 0: FG = 0: GOSUB 300
18 IF CH = 0 THEN PRINT "YOU CAN NOT FIND THE MULTIPLICATIVE INVERSE OF A NON-PRIME"
: GOTO 192
19 Z = 0
20 FOR B = 1 TO AST – 1
22 A = AST:
24 IF B = 1 THEN INV = 1: GOTO 172
26 W = B: AA = A: K = 0
40  C = 1: D = 0: E = 0: F = 1
50 K = K + 1
60 T(K) = INT(A / W): U = A - W * T(K)
70 IF ABS(U) = 1 OR U = 0 THEN 100
80 A = W: W = U: GOTO 50
100 FOR J = 1 TO K
110 G = E * T(J): H = F * T(J)
120 L = C - G: M = D – H
130 C = E: D = F :E=L:F=M
150 NEXT J
160 IF M < 0 THEN M = AA + M
162 INV = M
172 ON YT2 GOTO 174, 174, 174, 174, 176, 176, 178, 178
174 PRINT B; INV, : GOTO 190
176 PRINT USING "   ##### #####  "; B; INV; : GOTO 190
178 PRINT USING "   ####### #######  "; B; INV;
190 NEXT B: PRINT : PRINT "  _____________________ END OF MODULO "; AST; "_____________________"
: REM STOP
192 NEXT YT2
194 PRINT : PRINT  “_____________________________________________________________"
196 PRINT : PRINT "      QB 4.5 CODE                                    "
200 END
300 LV = LV + 1: IF LV = 1 THEN LV = LV + 1
310 IF CDBL(FIX(XX / LV)) = CDBL(XX / LV) THEN 350
320 IF LV < CDBL(XX - 1) THEN 300
350 RETURN
```

Appendix CH 3.6 ONE-STOP EC POINT MULTIPLICATION

```
2 CLS : PRINT : PRINT "    Appendix CH 3.6                                QB 4.5 CODE"
4 PRINT : PRINT " _____________________________________________________": REM STOP
6 REM RR7SQSD ARITHMETIC OPERATIONS. ONE-STOP MULTIPLY-BY-7 CERIFIED CORRECT8 REM
ELLIPTIC POINT MULTIPLICATION BASED ON THE VLSI CIRCUIT DESIGN ABSOLUTELY CORRECT
10 DIM ED(21, 10), KS(21), NT(21), XP(25), YP(25), MR(21), X(16000), Y(16000)
12 DATA -53,218,17011
14 READ AE, BE, P: PRINT "     CURVE CONSTANTS ARE  "; AE; BE; P: AST = P: PRINT
16 DATA 822,4,761,4,900,4,1191,4,1289,5,873,4,29,3,156,3,375,4,1379,5,2340,5,2983
19 DATA 5,3399,5,3408,5,3414,5,6085,5,7985,5,5690,5,8387,5,2713,5,619,4
20 PRINT "    SET OF SPECIMEN MULTIPLIERS": PRINT :
22 FOR QL = 1 TO 21: READ MR(QL), NT(QL): PRINT MR(QL); NT(QL), : NEXT QL: PRINT : PRINT "
SPECIMEN GENERATOR POINTS": PRINT
24 DATA 8360,14564,10,9691,4,16541,7,3799,12,10051,1,14209,5,4843,11,2893,23,1029
25 DATA 30,8895,38,5491,60,3457,68,9024,78,3457,88,2689,91,1785,92,9878,96,3457,18,2408
26 FOR DL = 1 TO 19: READ XP(DL), YP(DL): PRINT XP(DL); YP(DL), : NEXT DL: REM STOP
27 PRINT : PRINT " _____________________________________________________":  REM STOP
28 FOR ZL = 1 TO 19: X0P = XP(ZL): Y0P = YP(ZL): AXGP = XP(ZL): AYGP = YP(ZL): PRINT XP(ZL);
YP(ZL): PRINT
30 CC0 = 0: BB0 = X0P: EE0 = Y0P: XIP = X0P: YIP = EE0: Y0PR = EE0
32 FOR QQ = 1 TO 21: N = NT(QQ): KS = MR(QQ)
34 REM PRINT "FOR THE GENERATOR POINT ="; BB0; EE0: PRINT
40 REM PRINT " THE RESULTS OF THEMULTIPLY-BY-7 POINT MULTIPLICATION OPERATION ARE
AS FOLLOWS:-": PRINT
42 DATA 2,3,-2,3,2,2,-3,-2,3,-3,3,-3,3,3,2,1,1,-3,-2,2,1,3,-3,-1,-2,1,-3,1,3
43 DATA 1,2,1,1,-2,-3,1,-3,0,1,0,1,0,-1,-2,2,1,2,-2,-1,1,1,3,-1,3,-3,1,3,0,-3,-1,1,3,0,-2,-2
44 DATA 3,-3,-2,1,2,3,2,2,0,-2,2,3,-3,1,-1,3,3,3,1,1,1,1,-1,3,-3,2,-1,-3,3
46 REM X0P7 = 0: Y0P7 = 0:
48  FOR JK = N - 1 TO 0 STEP -1: READ ED(QQ, JK): PRINT ED(QQ, JK); : NEXT JK:  PRINT :  REM
STOP: REM NEXT QQ: STOP
50  FOR PK = N - 1 TO 0 STEP -1: REM 1 TO N
56   REM X0P7 = 0: Y0P7 = 0
58   REM N = NT(QQ)
60   REM FOR JK = N - 1 TO 0 STEP -1
62    EE0 = Y0PR:  REM PRINT ED(QQ, PK); : STOP:
64 IF ED(QQ, PK) < 0 THEN EE0 = -EE0
66  REM PRINT BB0; EE0; "INPUT TO UPDATE MODULES": REM STOP
68 IF ABS(ED(QQ, PK)) = 0 THEN XI7P = BBD7: YI7P = EED7: GOSUB 6000: CUD = C77: BUD = B77:
EUD = E77: GOTO 310
70 ON ABS(ED(QQ, PK)) GOTO 72, 80, 90
72 GOSUB 1000
76 WUD = C0P0: XUD = X0P0: YUD = Y0P0
78 GOTO 100
80  GOSUB 2000
84 WUD = C0P1: XUD = X0P1: YUD = Y0P1
88 GOTO 100
90 GOSUB 3000
94 WUD = C0P2: XUD = X0P2: YUD = Y0P2
100 REM PRINT WUD; XUD; YUD; " 1P,2P,3P SELECTING MODULES OUTPUT ": REM
```

Appendix 3.6 ONE-STOP EC POINT MULTIPLICATION

```
102  IF PK = N - 1 THEN XI7P = XUD: YI7P = YUD: BUD = XUD: EUD = YUD: CUD = WUD: GOTO 310:
REM INV = 1: GOTO 330
116   REM XI7P = BBD7: YI7P = EED7: REM ; "INPUT TO 7P":
120 GOSUB 6000: REM MULTIPLYING -BY - 7"
122 GOSUB 4000: REM UPDATE MODULE
310  IF CUD < 0 THEN CUD = CUD + P
320   BINV = CUD: AST = P: GOSUB 1500
330  BBD7 = (BUD * ((INV ^ 2) MOD P)) MOD P:
332 IF BBD7 < 0 THEN BBD7 = BBD7 + P
334  EED7 = (EUD * ((INV * ((INV ^ 2) MOD P)) MOD P)) MOD P:
336 IF EED7 < 0 THEN EED7 = EED7 + P
338 XI7P = BBD7: YI7P = EED7: PRINT BBD7; EED7; : REM PRINT : REM STOP
340 NEXT PK: PRINT : PRINT " END OF MULTIPLIER MR="; MR(QQ); "AND BASE POINT ="; BB0; ",";
EE0; : PRINT :
342 GOSUB 8000:  REM STOP
344      NEXT QQ:
345 PRINT : PRINT "   _______ END OF GENERATOR POINT"; XP(ZL); YP(ZL); ___": REM STOP:
 346 RESTORE 42: PRINT
348  NEXT ZL
349 PRINT : PRINT "_____________________________________________________": REM STOP
350 PRINT : PRINT "   QB 4.5 CODE                                        "
360  END

1000 REM 1P MODULE
1010 X0P0 = BB0
1020 Y0P0 = EE0
1030 C0P0 = 1
1040 RETURN

1500 REM TO FIND THE MULTIPLICATIVE INVERSES
1510 DIM T(30)
1520 IF BINV = 1 THEN LM = 1: GOTO 1650
1530 LA = AST: LB = BINV
1540 AA = LA: LK = 0: LC = 1: LD = 0: LE = 0: LF = 1
1550 LK = LK + 1:
1560 T(LK) = INT(LA / LB): U = LA - LB * T(LK)
1570 IF ABS(U) = 1 OR U = 0 THEN 1590
1580 LA = LB: LB = U: GOTO 1550
1590 FOR LJ = 1 TO LK
1600 LG = LE * T(LJ): LH = LF * T(LJ)
1610 LL = LC - LG: LM = LD - LH
1620 LC = LE: LD = LF: LE = LL: LF = LM
1630 NEXT LJ
1640 IF LM < 0 THEN LM = AA + LM
1650 INV = LM: REM LMDA = LM
1700 RETURN
```

Appendix 3.6 ONE-STOP EC POINT MULTIPLICATION

```
2000 REM 2P MODULE
2010 A1 = ((3 * (BB0 ^ 2) MOD P) MOD P + AE) MOD P
2020 B1 = ((A1 ^ 2) MOD P - (8 * ((BB0 * ((EE0 ^ 2) MOD P)) MOD P)) MOD P) MOD P
2022 REM IF B1 < 0 THEN B1 = B1 + P
2030 C1 = (2 * EE0) MOD P
2040 D1 = ((((C1 ^ 2) MOD P) * BB0) MOD P - B1) MOD P
2050 E1 = ((A1 * D1) MOD P - (8 * ((((EE0 ^ 2) MOD P) ^ 2) MOD P)) MOD P) MOD P
2052 F1 = ((((C1 ^ 2) MOD P) * BB0) MOD P + B1) MOD P
2054 REM PRINT A1; B1; C1; D1; E1: REM STOP
2056 C0P1 = C1: X0P1 = B1: Y0P1 = E1
2100 RETURN
3000 REM 3P MODULE
3010 GOSUB 2000
3020 A2 = ((EE0 * ((C1 * ((C1 ^ 2) MOD P)) MOD P)) MOD P - E1) MOD P
3052 B2 = ((A2 ^ 2) MOD P - (F1 * ((D1 ^ 2) MOD P)) MOD P) MOD P
3060 C2 = (C1 * D1) MOD P
3070 D2 = (((B1 * ((D1 ^ 2) MOD P)) MOD P) MOD P - B2) MOD P
3080 E2 = ((A2 * D2) MOD P - (E1 * ((D1 * ((D1 ^ 2) MOD P)) MOD P)) MOD P) MOD P
3082 C0P2 = C2: X0P2 = B2: Y0P2 = E2
3130 RETURN
4000 REM UPDATED MODULE
4010 AUD = (((E77 * ((WUD * ((WUD ^ 2) MOD P)) MOD P)) MOD P - (YUD * ((C77 * ((C77 ^ 2)
MOD P)) MOD P)) MOD P)) MOD P
4020 DUD = ((B77 * ((WUD ^ 2) MOD P)) MOD P - (XUD * ((C77 ^ 2) MOD P)) MOD P) MOD P
4030 FUD = ((B77 * ((WUD ^ 2) MOD P)) MOD P + (XUD * ((C77 ^ 2) MOD P)) MOD P) MOD P
4040 BUD = ((AUD ^ 2) MOD P - (FUD * ((DUD ^ 2) MOD P)) MOD P) MOD P
4050 CUD = (C77 * ((WUD * DUD) MOD P)) MOD P
4060 LUD = (C77 * DUD) MOD P
4070 HUD = ((XUD * ((LUD ^ 2) MOD P)) MOD P - BUD) MOD P
4080 EUD = ((AUD * HUD) MOD P - (YUD * ((LUD * ((LUD ^ 2) MOD P)) MOD P)) MOD P) MOD P
4084 REM PRINT CUD; BUD; EUD; "CENTRAL UPDATE MODULE OUPUT UN-INVERTED": REM
4086   XI7P = BUD: YI7P = EUD
4090 RETURN
6000 A17 = ((3 * ((XI7P ^ 2) MOD P)) MOD P + AE) MOD P
6010 B17 = ((A17 ^ 2) MOD P - (8 * ((XI7P * ((YI7P ^ 2) MOD P)) MOD P)) MOD P) MOD P
6020 C17 = (2 * YI7P) MOD P
6030 D17 = ((XI7P * ((C17 ^ 2) MOD P)) MOD P - B17) MOD P
6040 F17 = ((XI7P * ((C17 ^ 2) MOD P)) MOD P + B17) MOD P
6050  E17 = ((A17 * D17) MOD P - (8 * ((((YI7P ^ 2) MOD P) ^ 2) MOD P)) MOD P) MOD P
6052 REM E17 = ((A17 * D17) MOD P - (8 * ((YI7P ^ 4) MOD P)) MOD P) MOD P
6060 REM PRINT A17; B17; C17; D17; E17; F17; H17; "FROM MODULE 2P": REM STOP
6100 A27 = ((YI7P * ((C17 * ((C17 ^ 2) MOD P)) MOD P)) MOD P - E17) MOD P
6110 B27 = ((A27 ^ 2) MOD P - (F17 * ((D17 ^ 2) MOD P)) MOD P) MOD P
6120 C27 = (C17 * D17) MOD P
6130 H27 = ((B17 * ((D17 ^ 2) MOD P)) MOD P - B27) MOD P
```

Appendix 3.6 ONE-STOP EC POINT MULTIPLICATION

```
6140 D27 = ((XI7P * ((C27 ^ 2) MOD P)) MOD P - B27) MOD P
6150 F27 = ((XI7P * ((C27 ^ 2) MOD P)) MOD P + B27) MOD P
6160 E27 = ((A27 * H27) MOD P - (E17 * ((D17 * ((D17 ^ 2) MOD P)) MOD P)) MOD P) MOD P
6170 REM PRINT A27; B27; C27; D27; E27; F27; H27; "FROM MODULE 3P": REM STOP


6200 A37 = ((YI7P * ((C27 * ((C27 ^ 2) MOD P)) MOD P)) MOD P - E27) MOD P
6210 B37 = ((A37 ^ 2) MOD P - (F27 * ((D27 ^ 2) MOD P)) MOD P) MOD P
6220 C37 = (C17 * ((D17 * D27) MOD P)) MOD P
6230 H37 = ((B27 * ((D27 ^ 2) MOD P)) MOD P - B37) MOD P
6240 D37 = ((B27 * ((C37 ^ 2) MOD P)) MOD P - (B37 * ((C27 ^ 2) MOD P)) MOD P) MOD P
6250 F37 = ((B27 * ((C37 ^ 2) MOD P)) MOD P + (B37 * ((C27 ^ 2) MOD P)) MOD P) MOD P 6260
E37 = ((A37 * H37) MOD P - (E27 * ((D27 * ((D27 ^ 2) MOD P)) MOD P)) MOD P) MOD P 6270 REM
PRINT A37; B37; C37; D37; E37; F37; H37; "FROM MODULE 4P": REM STOP


6300 A77 = (((E27 * ((C37 * ((C37 ^ 2) MOD P)) MOD P)) - (E37 * ((C27 * ((C27 ^ 2) MOD P)) MOD
P)) MOD P)) MOD P
6310 B77 = ((A77 ^ 2) MOD P - (F37 * ((D37 ^ 2) MOD P)) MOD P) MOD P
6320 C77 = (C27 * ((C37 * D37) MOD P)) MOD P
6340 H77 = ((B37 * ((((C27 * D37) MOD P) ^ 2) MOD P)) MOD P - B77) MOD P
6344 HJ = (C27 * D37) MOD P
6350 E77 = ((A77 * H77) MOD P - (E37 * ((HJ * ((HJ ^ 2) MOD P)) MOD P)) MOD P) MOD P
6360 REM PRINT A77; B77; C77; D77; E77; F77; H77; "FROM MODULE 7P": REM PRINT : REM
STOP6400 RETURN


8000  AST = P
8002 J = 1: X(J) = AXGP: Y(J) = AYGP: REM PRINT X(J); Y(J)
8004 FOR J = 2 TO KS
8008 IF J = 2 THEN 8022
8010 BINV = (AXGP - X(J - 1)):
8012 IF BINV < 0 THEN BINV = P + BINV
8014  GOSUB 1500
8016 LMDA = ((AYGP - Y(J - 1)) * INV) MOD P: IF LMDA < 0 THEN LMDA = P + LMDA
8018 X(J) = (LMDA ^ 2 - X(J - 1) - AXGP) MOD P: IF X(J) < 0 THEN X(J) = X(J) + P
8020 GOTO 8032
8022 BINV = 2 * Y(J - 1): IF BINV <= 0 THEN BINV = P + BINV
8026 GOSUB 1500
8028 LMDA = ((3 * (X(J - 1) ^ 2) MOD P + AE) * INV) MOD P: IF LMDA < 0 THEN LMDA = P + LMDA
8030 X(J) = ((LMDA ^ 2) MOD P - 2 * X(J - 1)) MOD P: IF LMDA < 0 THEN LMDA = P + LMDA
8032 Y(J) = (LMDA * (X(J - 1) - X(J)) - Y(J - 1)) MOD P:  REM PRINT J;
8034 IF Y(J) < 0 THEN Y(J) = P + Y(J):
8036 REM PRINT X(J); Y(J)
8052 NEXT J: PRINT "AND FROM REFERENCE PROGRAM OPERATION [nP] FOR"; J - 1; "IS "; X(J - 1);
Y(J - 1): PRINT : REM STOP
8200 RETURN
```

Appendix CH 3.7 MESSAGE ENCRYPTION AND DECRYPTION

```
2 CLS : PRINT : PRINT "    Appendix  CH 3.7                          QB 4.5 CODE"
4 PRINT :  PRINT " _____________________________________________________":  REM STOP
5 CLS : REM PLAIN TEXT ENCRYPTION PROGRAM SOURCE CODE (FINAL  ALL UNIFYING  PROGRAM)
DIM JEP(8505), ED(21, 10), NT(17), XP(9), YP(9), MR(17), CD(23), YC1(22), YC2(22), CDC1(22),
CDC2(22), PSK(2, 22), XCBP(22), YCBP(22), WW(22), CCYA1(22), CCYA2(22), CY1(22), CY2(22), CYA1(22),
 CYA2(22), DC1(22), DC2(22), X(16000), Y(16000)
7 PRINT : PRINT "   THE PLAN TEXT MESSAGE FOR THE CRYPTOGRAPHIC EXERCISE IS": PRINT
8 PRINT "        'THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG'": REM STOP
16 LN = 1: PRINT " THE CORRESPONDING NUMERIC CODE OF THE MESSAGE TO BE TRANSMITTED"
: PRINT : REM STOP
18 DATA 1507,2704,2625,1201,527,2006,1827,2205,2715,1619,2113,2710,1524,2706,2314
20 DATA 1815,2702,311,2109,2717,605,20,99
22 READ CD(LN): IF CD(LN) <> 99 THEN PRINT CD(LN), : LN = LN + 1: GOTO 22
23 LN = LN - 1: REM 99 IS END OF MESSAGE
25 DATA -53,218,17011
26 READ AE, BE, P: PRINT : PRINT "     CURVE CONSTANTS ARE"; AE; BE; P; : PRINT
27 PRINT " ____________________________________________________________":   REM STOP
30  FOR BQ = 1 TO 11
32  ON BQ GOTO 34, 40, 50, 60, 70, 80, 90, 100, 130, 150, 200
34 REM PRINT : PRINT " THE ELLIPTIC CURVE POINTS FOR ENCRYPTION/DECRYPTION OPERATION ARE
:-"
35 GOSUB 6500: GOTO 270: REM BQ=1:REM CURVE POINTS GENERATOR
40 REM BQ=2:REM DATA CURVE POINTS SELECTOR /PROCESSOR
42 LN = 22: GOSUB 9800: PRINT "CODED INPUT MESSAGE"; "    THE CURVE EMBEDDMENT"
44 FOR FTO = 1 TO 22: PRINT USING "  ##   #####      ##### ##### "; FTO; CD(FTO); XCBP(FTO);
 YCBP(FTO)
46 NEXT FTO:
47 PRINT " _____________________________________________":     REM STOP
48 GOTO 270
50  REM LN = 1: KS = MSK: XGP = X0P: YGP = Y0P:
51 LN = 1: N = 1: GOSUB 1300: PM1 = BBD7: PM2 = EED7
55 PRINT "   THEREFORE MICHAEL'S PUBLIC KEY IS "; PM1; PM2; : REM STOP: REM BQ=3
56 YC1(LN) = PM1: YC2(LN) = PM2: W = 0: GOSUB 8600:  REM  FOR CHECKING
57  IF W <> 0 THEN PRINT "    MICHAEL'S CONFIRMED PUBLIC KEY IS "; CPM1; CPM2; PM2:  REM STOP
: GOTO 270
58 PRINT : PRINT " _____________________________________________": GOTO 270
59  PRINT "   FAILED MISSION": STOP: GOTO 262
60  KS = LSK: XGP = FXGP: YGP = FYGP: GOSUB 1300: REM BQ=4
61 PL1 = BBD7: PL2 = EED7: PRINT "  THEREFORE LILIAN'S PUBLIC KEY IS"; PL1; PL2: REM STOP
62  LN = 1: YC1(LN) = PL1: YC2(LN) = PL2: W = 0: GOSUB 8600: REM  CROSS CHECKING
64  IF W <> 0 THEN PRINT "    LILIAN'S CONFIRMED PUBLIC KEY IS "; CPL1; CPL2; PL2: REM STOP
: GOTO 270
65 PRINT : PRINT " _____________________________________________________":  GOTO 270
68  PRINT "   FAILED MISSION": STOP: GOTO 262
```

Appendix CH 3.7 MESSAGE ENCRYPTION AND DECRYPTION

```
70  REM  BQ=5
72 GOSUB 1300: CPLT1 = BBD7: CPLT2 = EED7: PRINT "    LILIAN'S TRANSFOPRMED PUBLIC KEY IS"
; CPLT1; CPLT2: REM STOP
74  LN = 1: YC1(LN) = CPLT1: YC2(LN) = CPLT2: W = 0: GOSUB 8600: REM FOR CHECKING
80   KS = LSK: XGP = CPM1: YGP = CPM2: BB0 = CPM1: EE0 = CPM2: REM BQ=6
81 GOSUB 1300: CPMT1 = BBD7: CPMT2 = EED7: PRINT "   MICHAEL'S TRANSFORMED PUBLIC KEY IS "
; CPMT1; CPMT2: REM STOP
82   LN = 1: YC1(LN) = CPMT1: YC2(LN) = CPMT2: W = 0: GOSUB 8600: REM 9000: REM FOR CHECKING
84  IF W <> 0 THEN PRINT "    MICHAEL'S TRANSFORMED AND CONFIRMED PUBLIC KEY IS "; CCPMT1
; CCPMT2; CPMT2:  REM GOTO 270
86 PRINT : PRINT " _____": GOTO 270
88  PRINT "   FAILED MISSION": STOP:  GOTO 262
90 LN = 22: GOSUB 7200: PRINT "      BELOW IS YOUR SCRAMBLED MESSAGE": PRINT : REM BQ=7
91 PRINT "  BLK/N0    ENCODED MESSAGE      EMBEDDED MESSAGE      ENCRYPTED  MESSAGE ": PRINT
92 FOR VV = 1 TO 22: PRINT USING "  ##        #####             ##### #####          ##### ##### "
; VV; CD(VV); XCBP(VV); YCBP(VV); CYA1(VV); CYA2(VV)
94 NEXT VV: PRINT "_____END OF ENCRYPTION_____":  REM STOP
96 GOTO 270
100 LN = 22: REM BQ = 8: REM CROSS CHECKING CIPHER
102 FOR FTJ = 1 TO LN: YC1(FTJ) = CYA1(FTJ): YC2(FTJ) = CYA2(FTJ): NEXT FTJ
104 GOSUB 8600: FOR FTK = 1 TO 22: PRINT USING "  ##   ##### ##### "; FTK; CCYA1(FTK); CCYA2(FTK);
106  IF WW(FTK) <> 0 THEN PRINT "A HIT": GOTO 110
108  PRINT " A MISS"
110 NEXT FTK:   PRINT
112 PRINT "   STATE OF CIPHER TEXT IN RELATION TO THE POINTS ON THE ELLIPTIC CURVE"
: REM STOP
116 GOTO 270
 REM ALIGNMENT
130 LN = 22: GOSUB 7200: REM  DECRYPTING CIHPER:REM BQ=9
134 FOR TV = 1 TO 22: PRINT USING "  ##   #####     ##### #####     ##### #####     ##### ##### "
; TV; CD(TV); XCBP(TV); YCBP(TV); CYA1(TV); CYA2(TV); DC1(TV); DC2(TV)
136 NEXT TV: PRINT : PRINT : PRINT "       -----THE DECIPHERED MESSAGE-------   ":   REM STOP
138 GOTO 270
150 LN = 22: FOR TJI = 1 TO LN: YC1(TJI) = DC1(TJI): REM BQ=10
152 YC2(TJI) = DC2(TJI): NEXT TJI
156 GOSUB 8600: REM CROSS CHECKING DECIPHERED MESSAGE:REM
158 FOR TJO = 1 TO 22: PRINT USING "  ##   ##### #####      ##### #####"; TJO;
DC1(TJO); DC2(TJO); CDC1(TJO); CDC2(TJO);
160 IF WW(TJO) <> 0 THEN PRINT , "A HIT": GOTO 164
162 PRINT " A MISS"
164 NEXT TJO: PRINT "   CONFIRMING THE DECODED TEXT TO BE THE ORIGNAL MESSAGE   ": REM STOP
166 PRINT : PRINT :  GOTO 270
```

Appendix CH 3.7 MESSAGE ENCRYPTION AND DECRYPTION

```
200 PRINT : REM
202 PRINT " BLK/N0 ENCODED MESSAGE  EMBEDDED MESSAGE  ENCRYPTED MESSAGE
 DECRYPTED MESSAGE ": PRINT
220 FOR FCK = 1 TO 22: REM BQ = 11
224  PRINT USING "  ##     #####          ##### #####      ##### #####     ##### #####"
; FCK; CD(FCK); XCBP(FCK); YCBP(FCK); CYA1(FCK); CYA2(FCK); DC1(FCK); DC2(FCK)
226 NEXT FCK:  PRINT "          FINAL  COMPARATIVE ANALYSIS IN SUMMARY": REM STOP
230   GOTO 270
262 PRINT : PRINT " FAILED MISSION DO NOT MOVE FURTHER": STOP: GOTO 300
270   NEXT BQ: PRINT : REM STOP
272  PRINT "                MISSION ACCOMPLISHED THANK YOU LORD          ":
274 PRINT : PRINT "_____________________________________________________":  REM STOP
276 PRINT : PRINT "      QB 4.5 CODE                                      "
300 END
1000 REM TO FIND THE MULTIPLICATIVE INVERSES
1010 DIM T(20)
1020 IF BINV = 1 THEN LS = 1: GOTO 1150
1030 LA = AST: LB = BINV
1040 AA = LA: LK = 0: LC = 1: LD = 0: LE = 0: LF = 1
1050 LK = LK + 1
1060 T(LK) = INT(LA / LB): U = LA - LB * T(LK)
1070 IF ABS(U) = 1 OR U = 0 THEN 1090
1080 LA = LB: LB = U: GOTO 1050
1090 FOR LJ = 1 TO LK
1100 LG = LE * T(LJ): LH = LF * T(LJ)
1110 LL = LC - LG: LS = LD – LH
1120 LC = LE: LD = LF: LE = LL: LF = LS
1130 NEXT LJ
1140 IF LS < 0 THEN LS = P + LS
1150 INV = LS: XC = LS
1160 RETURN
1300 REM POINT MULTIPLICATION CONTROLLER MODULE
1310 ON BQ GOTO 1320, 1320, 1387, 1389, 1332, 1346, 1320, 1320, 1320, 1320, 1320, 1320
1320 GOTO 1528: REM UN-USED PIN CODE-READING AND CHECKING
1332 XP(1) = PL1: YP(1) = PL2: QQ = 1: N = 1: MR(1) = 1289: NT(1) = 5: VEGA = 1: RESTORE 1464
: GOTO 1450
1346 XP(1) = PM1: YP(1) = PM2: N = 1: QQ = 2: MR(2) = 873: NT(2) = 4: VEGA = 2: RESTORE 1464
: GOTO 1450
1387 VEGA = 1: GOTO 1400
1389 VEGA = 2
1400 DATA 1289,5,873,4: REM 2713,5,619,4:
1410 PRINT : PRINT "SET OF SPECIMEN MULTIPLIERS":
```

Appendix CH 3.7 MESSAGE ENCRYPTION AND DECRYPTION

```
1420 FOR QL = 1 TO 2: READ MR(QL), NT(QL): PRINT MR(QL); NT(QL), : NEXT QL: PRINT : PRINT
1430 DATA 8360,14564:
1440 FOR DL = 1 TO 1: READ XP(DL), YP(DL): NEXT DL:
1450 FOR ZL = 1 TO 1: X0P = XP(ZL): Y0P = YP(ZL): : REM AXGP = XP(ZL): AYGP = YP(ZL)
1454 CC0 = 0: BB0 = X0P: EE0 = Y0P: XIP = X0P: YIP = EE0: Y0PR = EE0
1458 FOR QQ = 1 TO VEGA: N = NT(QQ): KS = MR(QQ)
1460  PRINT "FOR THE GENERATOR POINT ="; BB0; EE0: PRINT
1462  PRINT " THE RESULTS OF THE MULTIPLY-BY-7 POINT MULTIPLICATION OPERATION ARE AS
FOLLOWS:-": PRINT
1464 DATA 1,-3,-2,2,1,3,-3,-1,-2: REM 1,1,-1,3,-3,2,-1,-3,3
1466 REM X0P7 = 0: Y0P7 = 0:
1468  FOR JK = N - 1 TO 0 STEP -1: READ ED(QQ, JK): PRINT ED(QQ, JK); : NEXT JK: PRINT
: PRINT : REM  STOP:
1470  FOR PK = N - 1 TO 0 STEP -1: REM 1 TO N
1472    EE0 = Y0PR:  REM PRINT ED(QQ, PK); : STOP:
1474 IF ED(QQ, PK) < 0 THEN EE0 = -EE0
1476 REM  PRINT BB0; EE0; "INPUT TO UPDATE MODULES":  STOP
1478 ON ABS(ED(QQ, PK)) GOTO 1480, 1486, 1494
1480 GOSUB 1700
1482 WUD = C0P0: XUD = X0P0: YUD = Y0P0
1484 GOTO 1498
1486  GOSUB 2000
1490 WUD = C0P1: XUD = X0P1: YUD = Y0P1
1492 GOTO 1498
1494 GOSUB 3000
1496 WUD = C0P2: XUD = X0P2: YUD = Y0P2
1498  REM PRINT WUD; XUD; YUD; " 1P,2P,3P SELECTING MODULES OUTPUT ": REM
1500  IF PK = N - 1 THEN XI7P = XUD: YI7P = YUD: BUD = XUD: EUD = YUD: CUD = WUD: GOTO 1508:
 INV = 1: GOTO 1512
1502   REM XI7P = BBD7: YI7P = EED7: REM ; "INPUT TO 7P":
1504 GOSUB 5000: REM MULTIPLYING -BY - 7"
1506 GOSUB 4000: REM UPDATE MODULE
1510   BINV = CUD: AST = P: GOSUB 1000
1512  BBD7 = (BUD * ((INV ^ 2) MOD P)) MOD P:
1514 IF BBD7 < 0 THEN BBD7 = BBD7 + P
1516  EED7 = (EUD * ((INV * ((INV ^ 2) MOD P)) MOD P)) MOD P:
1518 IF EED7 < 0 THEN EED7 = EED7 + P
1520 XI7P = BBD7: YI7P = EED7: PRINT BBD7; EED7; : REM PRINT : REM STOP
1522      NEXT PK: PRINT : PRINT "     END OF MULTIPLIER MR="; MR(QQ); "AND BASE POINT ="
; BB0; "  "; EE0;
: PRINT : PRINT : REM RESTORE 44
1523 GOSUB 10000: REM STOP: REM TO INSERT ADDER
1524     NEXT QQ: RESTORE 42: PRINT
1526  NEXT ZL
```

```
1527 REM PRINT " ____________________________________________________________"
:     REM STOP
1528  RETURN
1700 REM 1P MODULE
1710 X0P0 = BB0
1720 Y0P0 = EE0
1730 C0P0 = 1
1740 RETURN

2000 REM 2P MODULE
2010 A1 = ((3 * (BB0 ^ 2) MOD P) MOD P + AE) MOD P
2020 B1 = ((A1 ^ 2) MOD P - (8 * ((BB0 * ((EE0 ^ 2) MOD P)) MOD P)) MOD P) MOD P
2022 REM IF B1 < 0 THEN B1 = B1 + P
2030 C1 = (2 * EE0) MOD P
2040 D1 = ((((C1 ^ 2) MOD P) * BB0) MOD P - B1) MOD P
2050 E1 = ((A1 * D1) MOD P - (8 * ((((EE0 ^ 2) MOD P) ^ 2) MOD P)) MOD P) MOD P
2052 F1 = ((((C1 ^ 2) MOD P) * BB0) MOD P + B1) MOD P
2054 REM PRINT A1; B1; C1; D1; E1: REM STOP
2056 C0P1 = C1: X0P1 = B1: Y0P1 = E1
2100 RETURN

3000 REM 3P MODULE
3010 GOSUB 2000
3020 A2 = ((EE0 * ((C1 * ((C1 ^ 2) MOD P)) MOD P)) MOD P - E1) MOD P
3052 B2 = ((A2 ^ 2) MOD P - (F1 * ((D1 ^ 2) MOD P)) MOD P) MOD P
3060 C2 = (C1 * D1) MOD P
3070 D2 = (((B1 * ((D1 ^ 2) MOD P)) MOD P) MOD P - B2) MOD P
3080 E2 = ((A2 * D2) MOD P - (E1 * ((D1 * ((D1 ^ 2) MOD P)) MOD P)) MOD P) MOD P
3082 C0P2 = C2: X0P2 = B2: Y0P2 = E2
3130 RETURN

4000 REM UPDATED MODULE
4010 AUD = (((E77 * ((WUD * ((WUD ^ 2) MOD P)) MOD P)) MOD P - (YUD * ((C77 * ((C77 ^ 2) MOD P)
MOD P)) MOD P
4020 DUD = ((B77 * ((WUD ^ 2) MOD P)) MOD P - (XUD * ((C77 ^ 2) MOD P)) MOD P) MOD P
4030 FUD = ((B77 * ((WUD ^ 2) MOD P)) MOD P + (XUD * ((C77 ^ 2) MOD P)) MOD P) MOD P
4040 BUD = ((AUD ^ 2) MOD P - (FUD * ((DUD ^ 2) MOD P)) MOD P) MOD P
4050 CUD = (C77 * ((WUD * DUD) MOD P)) MOD P
4060 LUD = (C77 * DUD) MOD P
4070 HUD = ((XUD * ((LUD ^ 2) MOD P)) MOD P - BUD) MOD P
4080 EUD = ((AUD * HUD) MOD P - (YUD * ((LUD * ((LUD ^ 2) MOD P)) MOD P)) MOD P) MOD P
4084 REM PRINT CUD; BUD; EUD; "CENTRAL UPDATE MODULE OUPUT UN-INVERTED": REM
4086   XI7P = BUD: YI7P = EUD
4090 RETURN
```

Appendix CH 3.7 MESSAGE ENCRYPTION AND DECRYPTION

```
5000 A17 = ((3 * ((XI7P ^ 2) MOD P)) MOD P + AE) MOD P
5010 B17 = ((A17 ^ 2) MOD P - (8 * ((XI7P * ((YI7P ^ 2) MOD P)) MOD P)) MOD P) MOD P
5020 C17 = (2 * YI7P) MOD P
5030 D17 = ((XI7P * ((C17 ^ 2) MOD P)) MOD P - B17) MOD P
5040 F17 = ((XI7P * ((C17 ^ 2) MOD P)) MOD P + B17) MOD P
5050  E17 = ((A17 * D17) MOD P - (8 * ((((YI7P ^ 2) MOD P) ^ 2) MOD P)) MOD P) MOD P
5052 REM E17 = ((A17 * D17) MOD P - (8 * ((YI7P ^ 4) MOD P)) MOD P) MOD P
5060  REM PRINT A17; B17; C17; D17; E17; F17; H17; "FROM MODULE 2P":  STOP
5100 A27 = ((YI7P * ((C17 * ((C17 ^ 2) MOD P)) MOD P)) MOD P - E17) MOD P
5110 B27 = ((A27 ^ 2) MOD P - (F17 * ((D17 ^ 2) MOD P)) MOD P) MOD P
5120 C27 = (C17 * D17) MOD P
5130 H27 = ((B17 * ((D17 ^ 2) MOD P)) MOD P - B27) MOD P
5140 D27 = ((XI7P * ((C27 ^ 2) MOD P)) MOD P - B27) MOD P
5150 F27 = ((XI7P * ((C27 ^ 2) MOD P)) MOD P + B27) MOD P
5160 E27 = ((A27 * H27) MOD P - (E17 * ((D17 * ((D17 ^ 2) MOD P)) MOD P)) MOD P) MOD P
5170  REM PRINT A27; B27; C27; D27; E27; F27; H27; "FROM MODULE 3P":  STOP
5200 A37 = ((YI7P * ((C27 * ((C27 ^ 2) MOD P)) MOD P)) MOD P - E27) MOD P
5210 B37 = ((A37 ^ 2) MOD P - (F27 * ((D27 ^ 2) MOD P)) MOD P) MOD P
5220 C37 = (C17 * ((D17 * D27) MOD P)) MOD P
5230 H37 = ((B27 * ((D27 ^ 2) MOD P)) MOD P - B37) MOD P
5240 D37 = ((B27 * ((C37 ^ 2) MOD P)) MOD P - (B37 * ((C27 ^ 2) MOD P)) MOD P) MOD P
5250 F37 = ((B27 * ((C37 ^ 2) MOD P)) MOD P + (B37 * ((C27 ^ 2) MOD P)) MOD P) MOD P
5260 E37 = ((A37 * H37) MOD P - (E27 * ((D27 * ((D27 ^ 2) MOD P)) MOD P)) MOD P) MOD P
5270  REM PRINT A37; B37; C37; D37; E37; F37; H37; "FROM MODULE 4P":  STOP
5300 A77 = (((E27 * ((C37 * ((C37 ^ 2) MOD P)) MOD P)) - (E37 * ((C27 * ((C27 ^ 2) MOD P))
 MOD P)) MOD P))  MOD P
5310 B77 = ((A77 ^ 2) MOD P - (F37 * ((D37 ^ 2) MOD P)) MOD P) MOD P
5320 C77 = (C27 * ((C37 * D37) MOD P)) MOD P
5340 H77 = ((B37 * ((((C27 * D37) MOD P) ^ 2) MOD P)) MOD P - B77) MOD P
5344 HJ = (C27 * D37) MOD P
5350 E77 = ((A77 * H77) MOD P - (E37 * ((HJ * ((HJ ^ 2) MOD P)) MOD P)) MOD P) MOD P
5360  REM PRINT A77; B77; C77; D77; E77; F77; H77; "FROM MODULE 7P":  PRINT : REM STOP
5400 RETURN
6500   REM QUADRATIC RESIDUE AND CURVE POINTS COMPUTATION MODULE:REM BQ=1
6502 DORR = P + 1: REM QUADRATIC RESIDUE AND CURVE POINTS COMPUTATION MODULE
:REM BQ=1
6503 PRINT "   THE QUADRATIC RESIDUES OF THE ELLPTIC CURVE EQUATION ARE:-":  PRINT
6504 FOR XD = 1 TO (P - 1) / 2
6506 IF (XD ^ 2) MOD P <> ((P - XD) ^ 2) MOD P THEN 6510
6508 JEP(XD) = (XD ^ 2) MOD P: PRINT USING "    ##### "; JEP(XD);
6510 NEXT XD:  PRINT : REM STOP
6512 FOR XD = 0 TO P – 1
6514 YS = (((XD ^ 2) MOD P) * XD - 53 * XD + 218) MOD P:  IF YS < 0 THEN YS = YS + P
```

```
6516 SLK = 0: GOSUB 9400: IF SLK = 0 THEN PRINT USING "    ##### ##### "; XD; QE1;
: REM     STOP
6518 NEXT XD:
6536  PRINT : PRINT : PRINT "               ORDER OF FIELD IS"; INT(DORR): PRINT
6537  PRINT "  _____________________________________________________":   REM STOP
6540 RETURN
7200 REM SCRAMBLING CODED MESSEGE WITH EC ADDITION FORMULAR
7220 ON BQ GOTO 7230, 7230, 7230, 7230, 7230, 7230, 7240, 7230, 7250, 7230
7230 GOTO 7270: PRINT : REM NOT ASSIGNED HERE
7240 FOR AT7 = 1 TO 22: XC1 = XCBP(AT7): XC2 = PLT1
7242 YC1 = YCBP(AT7): YC2 = PLT2: GOSUB 7500
7244 CYA1(AT7) = XWC: CYA2(AT7) = YWC:
7245 PRINT USING "  ##   #####       ##### ##### "; AT7; CYA1(AT7); CYA2(AT7)
7246 NEXT AT7:  GOTO 7270
7250 FOR AT10 = 1 TO 22: XC1 = CYA1(AT10): XC2 = PMT1:
7252 YC1 = CYA2(AT10): YC2 = P - PMT2: GOSUB 7500
7254 DC1(AT10) = XWC: DC2(AT10) = YWC:
7256 PRINT AT10; CYA1(AT10); CYA2(AT10); DC1(AT10); DC2(AT10):
7258 NEXT AT10:  GOTO 7270
7270  REM STOP:
7280 RETURN

7500 REM ELLIPTIC CURVE POINT ADDITION MODULE
7502 DYC = YC2 - YC1: IDXC = XC1 + XC2: DCX = XC2 - XC1: IF DCX < 0 THEN DCX = DCX + P
7506 BINV = DCX: AST = P
7520 GOSUB 1000:  REM CALLING INVERSE MODULE
7530 LCM = (DYC * XC) MOD P: LCM2 = (LCM ^ 2) MOD P
7540 XWC = (LCM2 - IDXC) MOD P: IF XWC < 0 THEN XWC = P + XWC
7550 YWC = ((LCM * ((XC1 + XC1 + XC2 - LCM2) MOD P)) - YC1) MOD P: IF YWC < 0 THEN
YWC = P + YWC
7560  REM PRINT XWC; YWC
7570 RETURN
8600 REM CHECKER MODULE:REM LIST TO BE CHECKED IS SUPPLIED AT THE CALLING
MODULE POINT
8602 FOR RT = 1 TO LN: PSK(2, RT) = 0: NEXT RT: CT = 0
8603 FOR H2 = 0 TO P – 1
8604 Y0 = ((H2 * ((H2 ^ 2) MOD P)) MOD P - 53 * H2 + 218) MOD P: IF Y0 < 0 THEN Y0 = Y0 + P
8605 DV = 0: YS = Y0: PTR = 2: GOSUB 9400:
8606 ON BQ GOTO 8607, 8608, 8608, 8608, 8608, 8608, 8608, 8607, 8608, 8607, 8608
8607 PRINT USING "    ##### ##### "; H2; QE1; : GOTO 8610
8608 GOTO 8648: REM UN-USED PIN CODE:
8610 DV = DV + 1:
```

```
8612 IF H2 = YC1(DV) AND (QE1 = YC2(DV) OR P - YC2(DV)) THEN CT = CT + 1: VAP1 = H2: VAP2 = QE1
: VAP3 = P - YC2(DV): PSK(2, DV) = 1: GOTO 8617: REM PRINT H2; QE1; YC2(DV); "A HIT ": GOTO 8618
8613 REM IF H2 = YC1(DV) AND QE1 = YC2(DV) THEN CT = CT + 1: VAP1 = H2: VAP2 = QE1
: VAP3 = P - YC2(DV): PSK(2, DV) = 1:
 PRINT H2; QE1; H2;P-QE1;:"A HIT ": GOTO 8618
8614 IF DV < LN THEN 8610: REM LENGTH OF ARRAY TO BE CHECKED
8616 PSK(2, DV) = 0: GOTO 8642: REM NEXT XF:
8617 IF BQ <> 3 OR BQ <> 4 OR BQ <> 5 OR BQ <> 6 THEN PRINT H2; QE1; YC2(DV); "A HIT "
 REM GOTO 8618
8618 ON BQ GOTO 8619, 8620, 8622, 8624, 8626, 8628, 8630, 8632, 8634, 8636, 8637
8619 GOTO 8648: REM  QQ=1:REM UN-USED PIN CODES-CURVE PARAMETERS
8620 GOTO 8648: REM QQ=2: REM UN-USED PIN CODE- DATA CURVE POINTS GENERATOR
8622 W = 1: CPM1 = VAP1: CPM2 = VAP2: GOTO 8642: REM BQ=3
8624 W = 1: CPL1 = VAP1: CPL2 = VAP2: GOTO 8642: REM BQ=4
8626 W = 1: CCPLT1 = VAP1: CCPLT2 = VAP2: GOTO 8642: REM BQ=5
8628 W = 1: CCPMT1 = VAP1: CCPMT2 = VAP2: GOTO 8642: REM BQ=6
8630 GOTO 8648: REM BQ=7 NOT IN USE PN CODE -ENCRYPTOR-CYA
8632 WW(DV) = 1: CCYA1(DV) = VAP1: CCYA2(DV) = VAP2: GOTO 8642: REM BQ=8
8634 GOTO 8648: REM BQ=9 NOT IN USE PIN CODE ENCRYPTION CURIATOR
8636 WW(DV) = 1: CDC1(DV) = VAP1: CDC2(DV) = VAP2: GOTO 8642: REM BQ=10
8637 GOTO 8648: REM QQ=12 NOT IN USE PINCODE DECRYPTOR CURIATOR
8642 NEXT H2: PRINT : PRINT
8644 PRINT "   TOTAL NUMBER OF HITS="; CT; "AND THE ORDER OF THE GROUP IS ="; RDR
:  REM STOP
8645 REM PRINT :
PRINT " _____":     STOP
8646 GOTO 8648: REM UN-USED PIN CODES FOR ENCRYPTION,DECRYPTION AND THEIR CURES
8648 RETURN
9400 REM
9402 DSQ = (YS ^ 2) MOD P:
9405 DCB = (DSQ * YS) MOD P:
9406 DQD = (DSQ ^ 2) MOD P
9408 DCT = (DQD ^ 2) MOD P
9409 D12 = (DCT * DQD) MOD P
9410 D60 = (((((D12 ^ 2) MOD P) ^ 2) MOD P) * D12) MOD P
9412 D68 = (D60 * DCT) MOD P:
9414 D5 = (((((D68 ^ 2) MOD P) ^ 2) MOD P) * D68) MOD P
9416 DD5 = (((((D5 ^ 2) MOD P) ^ 2) MOD P) * D5) MOD P
9418 DDD5 = (((((DD5 ^ 2) MOD P) ^ 2) MOD P) * DD5) MOD P:
9420 DEX = (DSQ * DCB) MOD P:
9422 PHJ = (DEX * DDD5) MOD P
9424 IF PHJ <> 1 THEN SLK = 1: GOTO 9460
9426 DORR = DORR + YS / P
```

Appendix CH 3.7 MESSAGE ENCRYPTION AND DECRYPTION

```
9428 VT2 = (YS ^ 2) MOD P
9429 VT3 = (VT2 * YS) MOD P
9440 VT30 = (((((((VT5 ^ 2) MOD P) ^ 2) MOD P) * VT5) MOD P) * VT5) MOD P
9441 VT34 = (VT30 * VT4) MOD P:
9444 VTT1 = (((((VT34 ^ 2) MOD P) ^ 2) MOD P) * VT34) MOD P:
9446 VTT2 = (((((VTT1 ^ 2) MOD P) ^ 2) MOD P) * VTT1) MOD P:
9448 VTT3 = (((((VTT2 ^ 2) MOD P) ^ 2) MOD P) * VTT2) MOD P
9450 QE1 = (VTT3 * VT2) MOD P
9452  REM PRINT XD; QE1, : REM STOP
9460 RETURN
9800 REM DATA CURVE POINTS BY SELECTION/PROCESSING
9802 KQ = 0: KP = -1: KQ = 1: GASO = CD(KQ): CV = 0
9804 KP = KP + 1:
9806 D1 = ((KP * ((KP ^ 2) MOD P)) MOD P - 53 * KP + 218) MOD P: IF D1 < 0 THEN D1 = D1 + P
9808 YS = D1: SLK = 0: PTR = 2: GOSUB 9400: BR = 0:
9810 IF SLK = 1 THEN 9804
9812 IF KP = CD(KQ) THEN CV = CV + 1: XCBP(CV) = KP: YCBP(CV) = QE1:
 PRINT USING "  ##   #####       ##### #####"; CV; CD(KQ); XCBP(CV); YCBP(CV): GOTO 9830
9814 IF KP < P - 1 THEN 9804:
9816 BR = CV:
9818 GASO = GASO + 1: GASO = GASO MOD P:
9820 IF GASO = XCBP(BR) THEN 9818:
9822 BR = BR - 1: IF BR > 0 THEN 9820
9824 KP = GASO: D1 = ((KP * ((KP ^ 2) MOD P)) MOD P - 53 * KP + 218) MOD P
: IF D1 < 0 THEN D1 = D1 + P
9826 YS = D1: SLK = 0: GOSUB 9400: IF SLK <> 1 THEN CV = CV + 1: XCBP(CV) = KP
: YCBP(CV) = QE1:
PRINT USING "  ##   #####       ##### #####"; CV; CD(KQ); XCBP(CV); YCBP(CV): GOTO 9830
9828  GOTO 9816
9830 KQ = KQ + 1
9832 IF KQ > LN THEN 9836
9834  GASO = CD(KQ): KP = -1: BR = 0: GOTO 9804: REM STOP
9836  RETURN
10000  AST = P
10002 J = 1: X(J) = XP(ZL): Y(J) = YP(ZL):  REM PRINT X(J); Y(J)
10004 FOR J = 2 TO KS
10008 IF J = 2 THEN 10022
10010 BINV = (XP(ZL) - X(J - 1)):
10012 IF BINV < 0 THEN BINV = P + BINV
10014  GOSUB 1000
```

Appendix CH 3.7 MESSAGE ENCRYPTION AND DECRYPTION

```
10016 LMDA = ((YP(ZL) - Y(J - 1)) * INV) MOD P: IF LMDA < 0 THEN LMDA = P + LMDA
10018 X(J) = (LMDA ^ 2 - X(J - 1) - XP(ZL)) MOD P: IF X(J) < 0 THEN X(J) = X(J) + P: REM PRINT X(J);
10020 GOTO 10032
10022 BINV = 2 * Y(J - 1): IF BINV <= 0 THEN BINV = P + BINV
10026 GOSUB 1000
10028 LMDA = ((3 * (X(J - 1) ^ 2) MOD P + AE) * INV) MOD P: IF LMDA < 0 THEN LMDA = P + LMDA
10030 X(J) = ((LMDA ^ 2) MOD P - 2 * X(J - 1)) MOD P: IF X(J) < 0 THEN X(J) = P + X(J)
: REM PRINT X(J);
10032 Y(J) = (LMDA * (X(J - 1) - X(J)) - Y(J - 1)) MOD P: REM  PRINT J;
10034 IF Y(J) < 0 THEN Y(J) = P + Y(J):
10036  REM PRINT X(J); Y(J)
10052 NEXT J: PRINT "AND FROM REFERENCE PROGRAM OPERATION [nP] FOR"; J - 1; "IS "; X(J - 1)
; Y(J - 1)
:  REM STOP
10200 RETURN
```

Appendix CH 3.8 RR7SQSD T-gate Multiplexing property Test QB 4.5 CODE

```
2 CLS : SCREEN 1: COLOR 6 : RR7SQSD multiplexing property Test
4 DIM X6(18), D0(18), D1(18), D2(18), D3(18), D4(18), D5(18), D6(18), X(18)
6 DATA -3,-2,-1,0,1,2,3
10 READ LM3, LM2, LM1, L0, L1, L2, L3
12 DATA 2,-3,1,0,2,-2,1,3,-3,0,-1,2,3,1,-2,-1,0,2
14 DATA 1,1,-2,-3,0,1,-2,1,1,2,-1,2,2,3,1,0,-1,2
16 DATA -2,1,-3,0,1,0,-2,3,3,1,2,2,3,-2,-2,1,3,-3
20 DATA 0,3,-3,2,-1,2,-3,3,3,0,0,1,2,3,1,1,-2,3
22 DATA 1,1,-1,2,-2,2,3,3,3,-3,-3,-3,2,-2,2,3,1,-3
24 DATA -3,2,1,0,-1,2,1,-1,1,2,3,0,1,3,2,0,1,0
26 DATA 3,2,1,1,2,3,-1,1,2,2,1,-3,3,2,-1,3,1,1
28 DATA 2,-3,-1,0,1,-2,3,0,2,1,2,-3,1,2,0,3,-2,-1
30 L = 0: GOSUB 200: REM STOP: REM DRAW "BL160": DRAW "BD60": DRAW "M+10,00"
32 PRINT : PRINT :  PRINT "  Appendix CH 3.8    QB 4.5   Code"
34 PRINT " _____": PRINT
36 FOR P = 1 TO 18
40 READ D0(P), D1(P), D2(P), D3(P), D4(P), D5(P), D6(P), X(P)
48 GOSUB 2000: REM CIRCUIT SIMULATION
50 GOSUB 8000: REM  OUTPUT PROCESSOR
55 NEXT P: REM VIEW PRINT 1 TO 10: COLOR 8, 4
58 PRINT " D3"; "  "; "D2"; "  "; "D1"; "  "; "D0"; "  "; "D-1"; "  "; "D-2"; "  "; "D-3"; " "; " X"; "
"; "O/P"
60 FOR K = 1 TO 6
64 PRINT D0(K); " "; D1(K); " "; D2(K); " "; D3(K); "  "; D4(K); "  "; D5(K); "  "; D6(K); " "; X(K);
" "; X6(K)
66 NEXT K: REM COLOR 6
68 FOR TY = 1 TO 9: PRINT : NEXT TY: PRINT
70 PRINT : PRINT "    CP-gate derived RR7SQSD T-gate "
76 PRINT "   Multiplexing ability simulation  ": REM STOP
 120 END
200 COLOR 8: DRAW "BD45"
210  DRAW "L150": DRAW "NR280": DRAW "BR140": DRAW "BD50": DRAW "BU100"
220  DRAW "D100": DRAW "U50": DRAW "BL100"
230  RETURN
2000 REM CIRCUIT  SIMULATION
2010 L = L + 1
3400 IF X(P) <= LM3 THEN X0 = D0(P): GOTO 4060
4050 X0 = D1(P)
```

Appendix CH 3.8 RR7SQSD T-gate Multiplexing property Test QB 4.5 CODE

```
4060 IF X(P) <= LM1 THEN X1 = D2(P): GOTO 6000
5065 X1 = D3(P)
6000 IF X(P) < L1 THEN X2 = D3(P): GOTO 6280
6075 X2 = D4(P)
6280 IF X(P) <= L2 THEN X3 = D5(P): GOTO 7090
6485 X3 = D6(P)
7090 IF X(P) <= LM2 THEN X4 = X0: GOTO 7100
7095 X4 = X1
7100 IF X(P) <= L1 THEN X5 = X2: GOTO 7310
7205 X5 = X3
7310 IF X(P) <= L0 THEN X6(L) = X4: GOTO 7220
7315 X6(L) = X5
7220 RETURN

8000 REM OUTPUT PROCESSOR
8202 IF X6(L) = LM3 AND X6(L - 1) = LM3 THEN DRAW "R10": GOTO 8900
8204 IF X6(L) = LM3 AND X6(L - 1) = LM2 THEN DRAW "D10": GOTO 8900
8206 IF X6(L) = LM3 AND X6(L - 1) = LM1 THEN DRAW "D20": DRAW "R10": GOTO 8900
8208 IF X6(L) = LM3 AND X6(L - 1) = L0 THEN DRAW "D30": DRAW "R10": GOTO 8900
8210 IF X6(L) = LM3 AND X6(L - 1) = L1 THEN DRAW "D40": DRAW "R10": GOTO 8900
8212 IF X6(L) = LM3 AND X6(L - 1) = L2 THEN DRAW "D50": DRAW "R10": GOTO 8900
8214 IF X6(L) = LM3 AND X6(L - 1) = L3 THEN DRAW "D60": DRAW "R10": GOTO 8900

8300 REM PHASE 0
8302 IF X6(L) = LM2 AND X6(L - 1) = LM3 THEN DRAW "U10": DRAW "R10": GOTO 8900
8304 IF X6(L) = LM2 AND X6(L - 1) = LM2 THEN DRAW "R10": GOTO 8900
8306 IF X6(L) = LM2 AND X6(L - 1) = LM1 THEN DRAW "D10": DRAW "R10": GOTO 8900
8308 IF X6(L) = LM2 AND X6(L - 1) = L0 THEN DRAW "D20": DRAW "R10": GOTO 8900
8310 IF X6(L) = LM2 AND X6(L - 1) = L1 THEN DRAW "D30": DRAW "R10": GOTO 8900
8312 IF X6(L) = LM2 AND X6(L - 1) = L2 THEN DRAW "D40": DRAW "R10": GOTO 8900
8314 IF X6(L) = LM2 AND X6(L - 1) = L3 THEN DRAW "D50": DRAW "R10": GOTO 8900

8400 REM PHASE 1
8402 IF X6(L) = LM1 AND X6(L - 1) = LM3 THEN DRAW "U20": DRAW "R10": GOTO 8900
8404 IF X6(L) = LM1 AND X6(L - 1) = LM2 THEN DRAW "U10": DRAW "R10": GOTO 8900
8406 IF X6(L) = LM1 AND X6(L - 1) = LM1 THEN DRAW "R10": GOTO 8900
8408 IF X6(L) = LM1 AND X6(L - 1) = L0 THEN DRAW "F10": GOTO 8900
8410 IF X6(L) = LM1 AND X6(L - 1) = L1 THEN DRAW "D20": DRAW "R10": GOTO 8900
8412 IF X6(L) = LM1 AND X6(L - 1) = L2 THEN DRAW "D30": DRAW "R10": GOTO 8900
8414 IF X6(L) = LM1 AND X6(L - 1) = L3 THEN DRAW "D40": DRAW "R10": GOTO 8900
```

Appendix CH 3.8 RR7SQSD T-gate Multiplexing property Test QB 4.5 CODE

```
8500 REM PHASE 2
8502 IF X6(L) = L0 AND X6(L - 1) = LM3 THEN DRAW "U30": DRAW "R10": GOTO 8900
8504 IF X6(L) = L0 AND X6(L - 1) = LM2 THEN DRAW "U20": DRAW "R10": GOTO 8900
8506 IF X6(L) = L0 AND X6(L - 1) = LM1 THEN DRAW "U10": DRAW "R10": GOTO 8900
8508 IF X6(L) = L0 AND X6(L - 1) = L0 THEN DRAW "R10": GOTO 8900
8510 IF X6(L) = L0 AND X6(L - 1) = L1 THEN DRAW "D10": DRAW "R10": GOTO 8900
8512 IF X6(L) = L0 AND X6(L - 1) = L2 THEN DRAW "D20": DRAW "R10": GOTO 8900
8514 IF X6(L) = L0 AND X6(L - 1) = L3 THEN DRAW "D30": DRAW "R10": GOTO 8900

8600 REM PHASE 3
8602 IF X6(L) = L1 AND X6(L - 1) = LM3 THEN DRAW "U40": DRAW "R10": GOTO 8900
8604 IF X6(L) = L1 AND X6(L - 1) = LM2 THEN DRAW "U30": DRAW "E10": GOTO 8900
8606 IF X6(L) = L1 AND X6(L - 1) = LM1 THEN DRAW "U20": DRAW "R10": GOTO 8900
8608 IF X6(L) = L1 AND X6(L - 1) = L0 THEN DRAW "U10": DRAW "R10": GOTO 8900
8610 IF X6(L) = L1 AND X6(L - 1) = L1 THEN DRAW "R10": GOTO 8900
8612 IF X6(L) = L1 AND X6(L - 1) = L2 THEN DRAW "D10": DRAW "R10": GOTO 8900
8614 IF X6(L) = L1 AND X6(L - 1) = L3 THEN DRAW "D20": DRAW "R10": GOTO 8900

8700 REM PHASE 4
8702 IF X6(L) = L2 AND X6(L - 1) = LM3 THEN DRAW "U50": DRAW "R10": GOTO 8900
8704 IF X6(L) = L2 AND X6(L - 1) = LM2 THEN DRAW "U40": DRAW "R10": GOTO 8900
8706 IF X6(L) = L2 AND X6(L - 1) = LM1 THEN DRAW "U30": DRAW "R10": GOTO 8900
8708 IF X6(L) = L2 AND X6(L - 1) = L0 THEN DRAW "U20": DRAW "R10": GOTO 8900
8710 IF X6(L) = L2 AND X6(L - 1) = L1 THEN DRAW "U10": DRAW "R10": GOTO 8900
8712 IF X6(L) = L2 AND X6(L - 1) = L2 THEN DRAW "R10": GOTO 8900
8714 IF X6(L) = L2 AND X6(L - 1) = L3 THEN DRAW "D10": DRAW "R10": GOTO 8900

8800 REM PHASE 5
8802 IF X6(L) = L3 AND X6(L - 1) = LM3 THEN DRAW "U60": DRAW "R10": GOTO 8900
8804 IF X6(L) = L3 AND X6(L - 1) = LM2 THEN DRAW "U50": DRAW "R10": GOTO 8900
8806 IF X6(L) = L3 AND X6(L - 1) = LM1 THEN DRAW "U40": DRAW "R10": GOTO 8900
8808 IF X6(L) = L3 AND X6(L - 1) = L0 THEN DRAW "U30": DRAW "R10": GOTO 8900
8810 IF X6(L) = L3 AND X6(L - 1) = L1 THEN DRAW "U20": DRAW "R10": GOTO 8900
8812 IF X6(L) = L3 AND X6(L - 1) = L2 THEN DRAW "U10": DRAW "R10": GOTO 8900
8814 IF X6(L) = L3 AND X6(L - 1) = L3 THEN DRAW "R10"
8900 SLEEP 1: RETURN
```

Appendix CH 3.9 RR7SQSD full adder circuit simulation QB 4.5 CODE

```
5   CLS : REM FULL ADDER :SCREEN 1: REM COLOR 4
10 DIM ALPHA(22), BETA(22), DL(3, 7, 7), CL(3, 7, 7), GAMA(23), DELT(23), XL0(3, 7), XC0(3, 7)
, XL1(3), XC1(3), X6(23)
20 DATA 0,1,2,3,-3,-2,-1, -1,-1,-1,-1,0,0,0
30 DATA 1,2,3,-3,-2,-1,0,-1,-1,-1,0,0,0,0
40 DATA  2,3,-3,-2,-1,0,1,-1,-1,0,0,0,0,0
50 DATA 3,-3,-2,-1,0,1,2,-1,0,0,0,0,0,0
60 DATA -3,-2,-1,0,1,2,3,0,0,0,0,0,0,0
70 DATA -2,-1,0,1,2,3,-3,0,0,0,0,0,0,1
80 DATA -1,0,1,2,3,-3,-2,0,0,0,0,0,1,1
82 REM
90 DATA 1,2,3,-3,-2,-1,0,-1,-1,-1,0,0,0,0
92 DATA 2,3,-3,-2,-1,0,1,-1,-1,0,0,0,0,0
94 DATA 3,-3,-2,-1,0,1,2,-1,0,0,0,0,0,0
96 DATA -3,-2,-1,0,1,2,3,0,0,0,0,0,0,0
98 DATA -2,-1,0,1,2,3,-3,0,0,0,0,0,0,1
100 DATA -1,0,1,2,3,-3,-2,0,0,0,0,0,1,1
102 DATA 0,1,2,3,-3,-2,-1,0,0,0,0,1,1,1
104 REM
106 DATA 2,3,-3,-2,-1,0,-1,-1,-1,0,0,0,0,0
108 DATA 3,-3,-2,-1,0,1,2,-1,0,0,0,0,0,0
110 DATA -3,-2,-1,0,1,2,3,0,0,0,0,0,0,0
114 DATA -1,0,1,2,3,-3,-2,0,0,0,0,0,1,1
116 DATA 0,1,2,3,-3,-2,-1,0,0,0,0,1,1,1
118 DATA 1,2,3,-3,-2,-1,0,0,0,0,1,1,1,1
120 REM DATA -3,-2,2,-1,1,3,3,-2,2,-2,2,1,1,2
122 DATA 3,2,2,0,-3,-3,2,2,1,3,0,2,-1,-3,2,0,-3,2,2,1,-1,2,-3,-3,2,3,2,-1,0,2,0,-2,0,0,2,2,1,2,1,0,2,3,0,3
124 DATA -3,-2,-1,0,1,2,3
130 FOR L = 1 TO 3
132 FOR G = 0 TO 6
140 FOR I = 0 TO 6
150 READ DL(L, G, I): PRINT DL(L, G, I); : NEXT I: PRINT ,
160 FOR I = 0 TO 6
170 READ CL(L, G, I): PRINT CL(L, G, I); : NEXT I
172 PRINT : PRINT
178  NEXT G: PRINT "L="; L - 1: REM STOP
180 NEXT L
182 REM FOR Y = 6 TO 0 STEP -1
183 FOR Y = 21 TO 0 STEP -1
184 READ ALPHA(Y), BETA(Y)
186 PRINT ALPHA(Y); BETA(Y); : NEXT Y: PRINT : REM PRINT : STOP
187 READ LM3, LM2, LM1, L0, L1, L2, L3
188 DELT(7)=0
190  REM  FOR Y = 6 TO 0 STEP -1
191 FOR Y = 21 TO 0 STEP -1
192 C = 0: P0 = -3
```

Appendix CH 3.9 RR7SQSD full adder circuit simulation QB 4.5 CODE

```
194 C = C + 1
196 IF C = 1 THEN X = ALPHA(Y): GOTO 340
220 IF C = 2 THEN X = BETA(Y): GOTO 300
230 X = DELT(Y + 1)
270 T = -1: TT = 0: P0 = -1
280 T = T + 1: TT = TT + 1
282 IF T > 2 THEN PRINT "ERROR3":  GOTO 440
284 P = P0 + T
290 IF X = P THEN XL2 = XL1(TT): XC2 = XC1(TT): GAMA(Y) = XL2: DELT(Y) = XC2: PRINT , GAMA(Y)
; DELT(Y); X; P; T; TT: GOTO 440
292 GOTO 280
300 FOR L = 1 TO 3
302 T = -1: P0 = -3: T=T+1
310  IF T > 6 THEN PRINT "ERROR2": GOTO 336
320  P = P0 + T
330  IF X = P THEN XL1(L) = XL0(L, T): XC1(L) = XC0(L, T): PRINT XL1(L); : GOTO 336
332 GOTO 304
336 NEXT L: PRINT : GOTO 194
340 P0 = -3: FOR L = 1 TO 3
350 FOR G = 0 TO 6
354 T = T + 1
356 IF T > 6 THEN PRINT "ERROR1": GOTO 380
358 P = P0 + T
360 IF X = P THEN XL0(L, G) = DL(L, G, T): XC0(L, G) = CL(L, G, T): PRINT XL0(L, G); : GOTO 380
370 GOTO 354
380 NEXT G: PRINT
390 NEXT L: PRINT "Y="; Y
400 GOTO 194
440 REM DELT(7) = DELT(6)
442 DELT(12) = DELT(11)
450  NEXT Y: GAMA(23) = DELT(0): X6(23) = DELT(0): PRINT GAMA(23);
: FOR Y = 0 TO 22: PRINT GAMA(Y); : NEXT Y
452 X6(1) = GAMA(23): FOR L = 2 TO 23: X6(L) = GAMA(L - 2): NEXT L: PRINT : PRINT
: FOR T = 1 TO 23: PRINT X6(T); : NEXT T
455  GOSUB 10000:PRINT "  Appendix CH 3.9           QB 4.5 Code"
456  PRINT " _____": PRINT
457 FOR K = 0 TO 21: REM 21 TO 0 STEP -1
459 IF K < 9 THEN 462
460 PRINT ALPHA(K);
464 IF K < 9 THEN 468
466 PRINT BETA(K);
468 NEXT K:  PRINT : K = 24: L = 0
500  GOSUB 8000: PRINT : PRINT : SCREEN 1: COLOR 1: REM FOR Y = 23 TO 1 STEP -1
: PRINT X6(Y); : NEXT Y
502 FOR TY = 1 TO 10: PRINT : NEXT TY: PRINT
504 PRINT : PRINT " Dual RR7SQSD full adder/multiplier"
505 PRINT "   simulation (addition mode)  ": REM STOP
600 END
```

Appendix CH 3.9 RR7SQSD full adder circuit simulation QB 4.5 CODE

```
8000 SCREEN 1: COLOR C2: REM OUTPUT PROCESSOR
8001 PRINT
8002 L = L + 1:  IF L = 1 THEN X6(L - 1) = 0: GOTO 8202
8004 IF L > 23 THEN RETURN
8006 X6(L) = GAMA(L - 2): REM PRINT X6(L);
8202 PRINT X6(L); : IF X6(L) = LM3 AND X6(L - 1) = LM3 THEN DRAW "R10": GOTO 8900
8204 IF X6(L) = LM3 AND X6(L - 1) = LM2 THEN DRAW "D10": GOTO 8900
8206 IF X6(L) = LM3 AND X6(L - 1) = LM1 THEN DRAW "D20": DRAW "R10": GOTO 8900
8208 IF X6(L) = LM3 AND X6(L - 1) = L0 THEN DRAW "D30": DRAW "R10": GOTO 8900
8210 IF X6(L) = LM3 AND X6(L - 1) = L1 THEN DRAW "D40": DRAW "R10": GOTO 8900
8212 IF X6(L) = LM3 AND X6(L - 1) = L2 THEN DRAW "D50": DRAW "R10": GOTO 8900
8214 IF X6(L) = LM3 AND X6(L - 1) = L3 THEN DRAW "D60": DRAW "R10": GOTO 8900

8300 REM PHASE 0
8302 IF X6(L) = LM2 AND X6(L - 1) = LM3 THEN DRAW "U10": DRAW "R10": GOTO 8900
8304 IF X6(L) = LM2 AND X6(L - 1) = LM2 THEN DRAW "R10": GOTO 8900
8306 IF X6(L) = LM2 AND X6(L - 1) = LM1 THEN DRAW "D10": DRAW "R10": GOTO 8900
8308 IF X6(L) = LM2 AND X6(L - 1) = L0 THEN DRAW "D20": DRAW "R10": GOTO 8900
8310 IF X6(L) = LM2 AND X6(L - 1) = L1 THEN DRAW "D30": DRAW "R10": GOTO 8900
8312 IF X6(L) = LM2 AND X6(L - 1) = L2 THEN DRAW "D40": DRAW "R10": GOTO 8900
8314 IF X6(L) = LM2 AND X6(L - 1) = L3 THEN DRAW "D50": DRAW "R10": GOTO 8900

8400 REM PHASE 1
8402 IF X6(L) = LM1 AND X6(L - 1) = LM3 THEN DRAW "U20": DRAW "R10": GOTO 8900
8404 IF X6(L) = LM1 AND X6(L - 1) = LM2 THEN DRAW "U10": DRAW "R10": GOTO 8900
8406 IF X6(L) = LM1 AND X6(L - 1) = LM1 THEN DRAW "R10": GOTO 8900
8408 IF X6(L) = LM1 AND X6(L - 1) = L0 THEN DRAW "F10": GOTO 8900
8410 IF X6(L) = LM1 AND X6(L - 1) = L1 THEN DRAW "D20": DRAW "R10": GOTO 8900
8412 IF X6(L) = LM1 AND X6(L - 1) = L2 THEN DRAW "D30": DRAW "R10": GOTO 8900
8414 IF X6(L) = LM1 AND X6(L - 1) = L3 THEN DRAW "D40": DRAW "R10": GOTO 8900

8500 REM PHASE 2
8502 IF X6(L) = L0 AND X6(L - 1) = LM3 THEN DRAW "U30": DRAW "R10": GOTO 8900
8504 IF X6(L) = L0 AND X6(L - 1) = LM2 THEN DRAW "U20": DRAW "R10": GOTO 8900
8506 IF X6(L) = L0 AND X6(L - 1) = LM1 THEN DRAW "U10": DRAW "R10": GOTO 8900
8508 IF X6(L) = L0 AND X6(L - 1) = L0 THEN DRAW "R10": GOTO 8900
8510 IF X6(L) = L0 AND X6(L - 1) = L1 THEN DRAW "D10": DRAW "R10": GOTO 8900
8512 IF X6(L) = L0 AND X6(L - 1) = L2 THEN DRAW "D20": DRAW "R10": GOTO 8900
8514 IF X6(L) = L0 AND X6(L - 1) = L3 THEN DRAW "D30": DRAW "R10": GOTO 8900

 8600 REM PHASE 3
 8602 IF X6(L) = L1 AND X6(L - 1) = LM3 THEN DRAW "U40": DRAW "R10": GOTO 8900
 8604 IF X6(L) = L1 AND X6(L - 1) = LM2 THEN DRAW "U30": DRAW "E10": GOTO 8900
 8606 IF X6(L) = L1 AND X6(L - 1) = LM1 THEN DRAW "U20": DRAW "R10": GOTO 8900
 8608 IF X6(L) = L1 AND X6(L - 1) = L0 THEN DRAW "U10": DRAW "R10": GOTO 8900
 8610 IF X6(L) = L1 AND X6(L - 1) = L1 THEN DRAW "R10": GOTO 8900
 8612 IF X6(L) = L1 AND X6(L - 1) = L2 THEN DRAW "D10": DRAW "R10": GOTO 8900
 8614 IF X6(L) = L1 AND X6(L - 1) = L3 THEN DRAW "D20": DRAW "R10": GOTO 8900
```

Appendix CH 3.9 RR7SQSD full adder circuit simulation QB 4.5 CODE

```
8600 REM PHASE 3
8602 IF X6(L) = L1 AND X6(L - 1) = LM3 THEN DRAW "U40": DRAW "R10": GOTO 8900
8604 IF X6(L) = L1 AND X6(L - 1) = LM2 THEN DRAW "U30": DRAW "E10": GOTO 8900
8606 IF X6(L) = L1 AND X6(L - 1) = LM1 THEN DRAW "U20": DRAW "R10": GOTO 8900
8608 IF X6(L) = L1 AND X6(L - 1) = L0 THEN DRAW "U10": DRAW "R10": GOTO 8900
8610 IF X6(L) = L1 AND X6(L - 1) = L1 THEN DRAW "R10": GOTO 8900
8612 IF X6(L) = L1 AND X6(L - 1) = L2 THEN DRAW "D10": DRAW "R10": GOTO 8900
8614 IF X6(L) = L1 AND X6(L - 1) = L3 THEN DRAW "D20": DRAW "R10": GOTO 8900

8700 REM PHASE 4
8702 IF X6(L) = L2 AND X6(L - 1) = LM3 THEN DRAW "U50": DRAW "R10": GOTO 8900
8704 IF X6(L) = L2 AND X6(L - 1) = LM2 THEN DRAW "U40": DRAW "R10": GOTO 8900
8706 IF X6(L) = L2 AND X6(L - 1) = LM1 THEN DRAW "U30": DRAW "R10": GOTO 8900
8708 IF X6(L) = L2 AND X6(L - 1) = L0 THEN DRAW "U20": DRAW "R10": GOTO 8900
8710 IF X6(L) = L2 AND X6(L - 1) = L1 THEN DRAW "U10": DRAW "R10": GOTO 8900
8712 IF X6(L) = L2 AND X6(L - 1) = L2 THEN DRAW "R10": GOTO 8900
8714 IF X6(L) = L2 AND X6(L - 1) = L3 THEN DRAW "D10": DRAW "R10": GOTO 8900

8800 REM PHASE 5
8802 IF X6(L) = L3 AND X6(L - 1) = LM3 THEN DRAW "U60": DRAW "R10": GOTO 8900
8804 IF X6(L) = L3 AND X6(L - 1) = LM2 THEN DRAW "U50": DRAW "R10": GOTO 8900
8806 IF X6(L) = L3 AND X6(L - 1) = LM1 THEN DRAW "U40": DRAW "R10": GOTO 8900
8810 IF X6(L) = L3 AND X6(L - 1) = L1 THEN DRAW "U20": DRAW "R10": GOTO 8900
8812 IF X6(L) = L3 AND X6(L - 1) = L2 THEN DRAW "U10": DRAW "R10": GOTO 8900
8814 IF X6(L) = L3 AND X6(L - 1) = L3 THEN DRAW "R10"
8900 SLEEP 1: GOTO 8002
10000    SCREEN 1:  COLOR 9

10002  DRAW "BD15"
10010   DRAW "L150": DRAW "NR280": DRAW "BR140": DRAW "BD50": DRAW "BU100"
10020    DRAW "D100": DRAW "U50": DRAW "BL100"
10030   RETURN
```

Appendix CH3.10 Quasi-RR7SQSD T-gate based multiplier circuit simulation QB 4.5 CODE

```
2 REM PRINT “SIMULATION OF  RR7SQSD FULL MULTIPLIER CORRECTED VERSION"
5 CLS : REM SCREEN 1: COLOR 4
10 DIM ALPHA(22), BETA(22), DL(3, 7, 7), CL(3, 7, 7), GAMA(23), DELT(23), ZGA(23), X6(23)
20 DATA 1,-2,2,-1,3,0,-3,1,1,0,0,-1,-1,-1
30 DATA -2,3,1,-1,-3,2,0,1,0,0,0,0,-1,-1
40 DATA 2,1,0,-1,-2,-3,3,0,0,0,0,0,0,-1
50 DATA -1,-1,-1,-1,-1,-1,-1,-1,0,0,0,0,0,0,0
60 DATA 3,-3,-2,-1,0,1,2,-1,0,0,0,0,0,0
70 DATA 0,2,-3,-1,1,3,2,-1,-1,0,0,0,0,1
80 DATA -3,0,3,-1,2,-2,1,-1,-1,-1,0,0,1,1

82 REM
90 DATA 2,-1,3,0,-3,1,-2,1,1,0,0,0,-1,-1
92 DATA -1,-3,2,0,-2,3,1,1,1,0,0,0,-1,-1
94 DATA 3,2,1,0,-1,-2,-3,0,0,0,0,0,0,0
96 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
98 DATA -3,-2,-1,0,1,2,3,0,0,0,0,0,0,0
100 DATA 1,3,-2,0,2,-3,-1,-1,-1,0,0,0,1,1
102 DATA -2,1,-3,0,3,-1,2,-1,-1,0,0,0,1,1

104 REM
106 DATA 3,0,-3,1,-2,2,-1,1,1,1,0,0,-1,-1
108 DATA 0,-2,3,1,-1,-3,2,1,1,0,0,0,0,-1
110 DATA -3,3,2,1,0,-1,-2,1,0,0,0,0,0,0
112 DATA 1,1,1,1,1,1,1,0,0,0,0,0,0,0
114 DATA -2,-1,0,1,2,3,-3,0,0,0,0,0,0,1
116 DATA 2,-3,-1,1,3,-2,0,-1,0,0,0,0,1,1
118 DATA -1,2,-2,1,-3,0,3,-1,-1,0,0,1,1,1

120 REM
130 FOR L = 1 TO 3
132 FOR G = 0 TO 6
140 FOR I = 0 TO 6
150 READ DL(L, G, I): PRINT DL(L, G, I); : NEXT I: PRINT ,
160 FOR I = 0 TO 6
170 READ CL(L, G, I): PRINT CL(L, G, I); : NEXT I
172 PRINT : PRINT
178  NEXT G: PRINT "L="; L - 1: REM STOP
179 NEXT L
180 REM DATA 3,2,2,0,-3,-3,2,2,1,3,0,2,-1,-3,2,0,-3,2,2,1
REM ,-1,2,-3,-3,2,3,2,-1,0,2,0,-2,0,0,2,2,1,2,1,0,2,3,0,3
181 DATA 0,3,2,3,1,0,1,2,2,2,0,0,0,-2,0,2,2,-1,2,3,-3,-3,-1
182 DATA 2,2,1,-3,2,2,0,-1,-3,0,2,1,3,2,2,-3,-3,2,0,3,2
183 FOR Y = 1 TO 22
```

Appendix CH3.10 Quasi-RR7SQSD T-gate based multiplier circuit simulation QB 4.5 CODE

```
184 READ ALPHA(Y), BETA(Y)
192 Z = ALPHA(Y) * BETA(Y)
194 IF Z < -3 OR Z = -3 AND DELT(Y - 1) = -1 THEN DELT(Y) = -1: GOTO 220
196 IF Z > 3 OR Z = 3 AND DELT(Y - 1) = 1 THEN DELT(Y) = 1: GOTO 220
198 DELT(Y) = 0
220 ZGA(Y) = Z - 7 * DELT(Y)
221 GAMA(Y) = ZGA(Y) + DELT(Y - 1)
223 X6(Y) = GAMA(Y)
225   NEXT Y: GAMA(Y) = DELT(Y - 1): X6(Y) = GAMA(Y):
228 READ LM3, LM2, LM1, L0, L1, L2, L3
230 DATA -3,-2,-1,0,1,2,3
453  GOSUB 10000:
454  PRINT "  Appendix CH4.8B          QB 4.5 Code"
456  PRINT " _____": PRINT
458 FOR K = 1 TO 22: REM 22 TO 1 STEP -1
459  REM IF K < 9 THEN 462
460 PRINT ALPHA(K);
462 NEXT K: PRINT : FOR K = 1 TO 22
464  REM IF K < 9 THEN 468
466 PRINT BETA(K);
468 NEXT K:  PRINT : L = 0
500 GOSUB 8000: PRINT : PRINT : SCREEN 1: COLOR 1:
502 FOR TY = 1 TO 8: PRINT : NEXT TY: PRINT
504 PRINT : PRINT " Dual RR7SQSD full adder/multiplier"
505 PRINT "   simulation (multiplication mode)  ": REM STOP
600 END

8000 SCREEN 1: COLOR C2: L = 0: REM OUTPUT PROCESSOR
8001 PRINT
8002 L = L + 1: IF L > 23 THEN RETURN
8202 PRINT X6(L);
8203 IF X6(L) = LM3 AND X6(L - 1) = LM3 THEN DRAW "R10": GOTO 8900
8204 IF X6(L) = LM3 AND X6(L - 1) = LM2 THEN DRAW "D10": GOTO 8900
8206 IF X6(L) = LM3 AND X6(L - 1) = LM1 THEN DRAW "D20": DRAW "R10": GOTO 8900
8208 IF X6(L) = LM3 AND X6(L - 1) = L0 THEN DRAW "D30": DRAW "R10": GOTO 8900
8210 IF X6(L) = LM3 AND X6(L - 1) = L1 THEN DRAW "D40": DRAW "R10": GOTO 8900
8212 IF X6(L) = LM3 AND X6(L - 1) = L2 THEN DRAW "D50": DRAW "R10": GOTO 8900
8214 IF X6(L) = LM3 AND X6(L - 1) = L3 THEN DRAW "D60": DRAW "R10": GOTO 8900

8300 REM PHASE 0
8302 IF X6(L) = LM2 AND X6(L - 1) = LM3 THEN DRAW "U10": DRAW "R10": GOTO 8900
8304 IF X6(L) = LM2 AND X6(L - 1) = LM2 THEN DRAW "R10": GOTO 8900
8306 IF X6(L) = LM2 AND X6(L - 1) = LM1 THEN DRAW "D10": DRAW "R10": GOTO 8900
8308 IF X6(L) = LM2 AND X6(L - 1) = L0 THEN DRAW "D20": DRAW "R10": GOTO 8900
8310 IF X6(L) = LM2 AND X6(L - 1) = L1 THEN DRAW "D30": DRAW "R10": GOTO 8900
8312 IF X6(L) = LM2 AND X6(L - 1) = L2 THEN DRAW "D40": DRAW "R10": GOTO 8900
8314 IF X6(L) = LM2 AND X6(L - 1) = L3 THEN DRAW "D50": DRAW "R10": GOTO 8900
```

Appendix CH3.10 Quasi-RR7SQSD T-gate based multiplier circuit simulation QB 4.5 CODE

```
8400 REM PHASE 1
8402 IF X6(L) = LM1 AND X6(L - 1) = LM3 THEN DRAW "U20": DRAW "R10": GOTO 8900
8404 IF X6(L) = LM1 AND X6(L - 1) = LM2 THEN DRAW "U10": DRAW "R10": GOTO 8900
8406 IF X6(L) = LM1 AND X6(L - 1) = LM1 THEN DRAW "R10": GOTO 8900
8408 IF X6(L) = LM1 AND X6(L - 1) = L0 THEN DRAW "D10": DRAW "R10": GOTO 8900
8410 IF X6(L) = LM1 AND X6(L - 1) = L1 THEN DRAW "D20": DRAW "R10": GOTO 8900
8412 IF X6(L) = LM1 AND X6(L - 1) = L2 THEN DRAW "D30": DRAW "R10": GOTO 8900
8414 IF X6(L) = LM1 AND X6(L - 1) = L3 THEN DRAW "D40": DRAW "R10": GOTO 8900
8500 REM PHASE 2
8502 IF X6(L) = L0 AND X6(L - 1) = LM3 THEN DRAW "U30": DRAW "R10": GOTO 8900
8504 IF X6(L) = L0 AND X6(L - 1) = LM2 THEN DRAW "U20": DRAW "R10": GOTO 8900
8506 IF X6(L) = L0 AND X6(L - 1) = LM1 THEN DRAW "U10": DRAW "R10": GOTO 8900
8508 IF X6(L) = L0 AND X6(L - 1) = L0 THEN DRAW "R10": GOTO 8900
8510 IF X6(L) = L0 AND X6(L - 1) = L1 THEN DRAW "D10": DRAW "R10": GOTO 8900
8512 IF X6(L) = L0 AND X6(L - 1) = L2 THEN DRAW "D20": DRAW "R10": GOTO 8900
8514 IF X6(L) = L0 AND X6(L - 1) = L3 THEN DRAW "D30": DRAW "R10": GOTO 8900
8600 REM PHASE 3
8602 IF X6(L) = L1 AND X6(L - 1) = LM3 THEN DRAW "U40": DRAW "R10": GOTO 8900
8604 IF X6(L) = L1 AND X6(L - 1) = LM2 THEN DRAW "U30": DRAW "R10": GOTO 8900
8606 IF X6(L) = L1 AND X6(L - 1) = LM1 THEN DRAW "U20": DRAW "R10": GOTO 8900
8608 IF X6(L) = L1 AND X6(L - 1) = L0 THEN DRAW "U10": DRAW "R10": GOTO 8900
8610 IF X6(L) = L1 AND X6(L - 1) = L1 THEN DRAW "R10": GOTO 8900
8612 IF X6(L) = L1 AND X6(L - 1) = L2 THEN DRAW "D10": DRAW "R10": GOTO 8900
8614 IF X6(L) = L1 AND X6(L - 1) = L3 THEN DRAW "D20": DRAW "R10": GOTO 8900
8700 REM PHASE 4
8702 IF X6(L) = L2 AND X6(L - 1) = LM3 THEN DRAW "U50": DRAW "R10": GOTO 8900
8704 IF X6(L) = L2 AND X6(L - 1) = LM2 THEN DRAW "U40": DRAW "R10": GOTO 8900
8706 IF X6(L) = L2 AND X6(L - 1) = LM1 THEN DRAW "U30": DRAW "R10": GOTO 8900
8708 IF X6(L) = L2 AND X6(L - 1) = L0 THEN DRAW "U20": DRAW "R10": GOTO 8900
8710 IF X6(L) = L2 AND X6(L - 1) = L1 THEN DRAW "U10": DRAW "R10": GOTO 8900
8712 IF X6(L) = L2 AND X6(L - 1) = L2 THEN DRAW "R10": GOTO 8900
8714 IF X6(L) = L2 AND X6(L - 1) = L3 THEN DRAW "D10": DRAW "R10": GOTO 8900
8800 REM PHASE 5
8802 IF X6(L) = L3 AND X6(L - 1) = LM3 THEN DRAW "U60": DRAW "R10": GOTO 8900
8804 IF X6(L) = L3 AND X6(L - 1) = LM2 THEN DRAW "U50": DRAW "R10": GOTO 8900
8806 IF X6(L) = L3 AND X6(L - 1) = LM1 THEN DRAW "U40": DRAW "R10": GOTO 8900
8808 IF X6(L) = L3 AND X6(L - 1) = L0 THEN DRAW "U30": DRAW "R10": GOTO 8900
8810 IF X6(L) = L3 AND X6(L - 1) = L1 THEN DRAW "U20": DRAW "R10": GOTO 8900
8812 IF X6(L) = L3 AND X6(L - 1) = L2 THEN DRAW "U10": DRAW "R10": GOTO 8900
8814 IF X6(L) = L3 AND X6(L - 1) = L3 THEN DRAW "R10"
8900 SLEEP 1: GOTO 8002
10000  SCREEN 1:  COLOR 9
10002  DRAW "BD15"
10010  DRAW "L150": DRAW "NR280": DRAW "BR140": DRAW "BD50": DRAW "BU100"
10020  DRAW "D100": DRAW "U50": DRAW "BL100"
10030  RETURN
```

Appendix CH4.1 Output (in part) of the big integer multiplication program shown in Appendix CH3.3

THIS PROGRAM CONVERTS A 160 DECIMAL DIGIT NUMBER TO RR7SQSD AND AUTOMATICALLY COMPUTES ITS SQUARE

_____ BELOW IS THE VALUE OF THE OPERAND _____

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1856 2115 9210 1757 4302 4531 6367 1207 7895 2313 7945 8679 4030 4589 6354 4756 8523 9546 8671 7417 7532 1596 7894 4562 3216 4756 2346 7894 4562 7417 3216 1596 7895 4896 5643 1459 2587 3694 1002 7777

_____ GENERATING THE RADIX-7 DIGITS _____

2 3 1 2 2 2 6 2 4 6 1 0 2 2 3 6 1 6 0 6 5 5 2 3 6 2 1 5 1 2 3 6 1 6 5 1 0 5 0 6 4 5 3 4 3 4 4 6 4 6 4 6 5 1 2 5 1 5 3 0 0 1 4 4 0 1 2 5 1 0 6 5 3 3 0 6 1 0 6 1 4 0 0 0 2 4 4 4 4 2 4 1 5 6 5 5 1 2 5 2 4 1 1 6 5 0 4 0 0 0 0 6 0 5 4 1 0 1 0 5 3 1 0 0 6 6 3 0 2 1 6 3 5 4 5 1 1 5 6 6 0 2 0 6 6 2 3 1 4 3 6 1 3 6 6 0 6 2 1 3 6 4 5 6 5 2 1 6 3 0 2 3 0 4 0 6 5 2 3 3 0 6 3 1 1 1 5 1 5

_____ CONVERTING TO THE RR7SQSD DOMAIN _____

2 3 1 2 2 3 -1 3 -2 -1 1 0 2 3 -3 -1 2 -1 1 0 -1 -2 3 -3 -1 2 2 -2 1 3 -3 -1 2 0 -2 1 1 -2 1 0 -2 -1 -3 -2 -3 -2 -2 0 -2 0 -2 0 -2 1 3 -2 2 -2 3 0 0 2 -2 -3 0 1 3 -2 1 1 0 -2 3 3 1 -1 1 1 -1 2 -3 0 0 0 3 -2 -2 -2 -3 3 -3 2 -1 0 -1 -2 1 3 -2 3 -3 1 2 0 -2 1 -3 0 0 0 1 -1 1 -1 -3 1 0 1 1 -2 3 1 0 1 0 -1 3 0 2 2 0 -3 -1 -2 -2 1 2 -1 0 -1 0 2 1 0 -1 2 3 2 -2 -3 -1 2 -3 0 -1 1 -1 2 2 -3 0 -2 -1 0 -2 2 2 -1 3 0 2 3 1 -3 1 0 -2 2 3 3 1 -1 3 1 1 2 -2 2 -2

_____NOW PERORMING RR7SQSD MULTIPLICATION _____

344 5521 4746 5294 1107 1973 2986 3323 9655 3687 9435 5185 7628 4286 6565 2934 3207 3506 8585 9607 473 306 3112 4319 206 6212 5169 2140 9578 9340 2381 4333 3222 2442 1109 5551 1824 9035 1320 7474 1145 8850 2040 8256 203 6672 1000 6553 9676 9795 6445 5618 8679 6448 7754 4747 9881 7224 4197 4580 4089 6943 3712 1435 234 1419 5569 2433 4453 4172 7302 1161 258 3598 1966 8034 6820 2039 1156 1729

_____ END OF THE LAST ITERATION _____

Appendix CH 4.2 Result of the multiplicative inverse computation (mod 37, 79,241)

LHS IS THE NUMBER AND THE RHS IS THE MULTIPLICATIVE INVERSE

1	1	2	19	3	25	4	28	5	15	6	31
7	16	8	14	9	33	10	26	11	27	12	34
13	20	14	8	15	5	16	7	17	24	18	35
19	2	20	13	21	30	22	32	23	29	24	17
25	3	26	10	27	11	28	4	29	23	30	21
31	6	32	22	33	9	34	12	35	18	36	36

END OF MODULO 37

1	1	2	40	3	53	4	20	5	16	6	66
7	34	8	10	9	44	10	8	11	36	12	33
13	73	14	17	15	58	16	5	17	14	18	22
19	25	20	4	21	64	22	18	23	55	24	56
25	19	26	76	27	41	28	48	29	30	30	29
31	51	32	42	33	12	34	7	35	70	36	11
37	47	38	52	39	77	40	2	41	27	42	32
43	68	44	9	45	72	46	67	47	37	48	28
49	50	50	49	51	31	52	38	53	3	54	60
55	23	56	24	57	61	58	15	59	75	60	54
61	57	62	65	63	74	64	21	65	62	66	6
67	46	68	43	69	71	70	35	71	69	72	45
73	13	74	63	75	59	76	26	77	39	78	78

END OF MODULO 79

1	1	2	121	3	161	4	181	5	193	6	201
7	69	8	211	9	134	10	217	11	22	12	221
13	204	14	155	15	225	16	226	17	156	18	67
19	203	20	229	21	23	22	11	23	21	24	231
25	135	26	102	27	125	28	198	29	133	30	233
31	70	32	113	33	168	34	78	35	62	36	154
37	228	38	222	39	68	40	235	41	194	42	132
43	213	44	126	45	75	46	131	47	200	48	236
49	182	50	188	51	52	52	51	53	191	54	183
55	149	56	99	57	148	58	187	59	192	60	237
61	162	62	35	63	88	64	177	65	89	66	84
67	18	68	39	69	7	70	31	71	129	72	77
73	208	74	114	75	45	76	111	77	72	78	34
79	180	80	238	81	122	82	97	83	151	84	66
85	224	86	227	87	205	88	63	89	65	90	158
91	98	92	186	93	184	94	100	95	137	96	118
97	82	98	91	99	56	100	94	101	105	102	26
103	117	104	146	105	101	106	216	107	232	108	212
109	199	110	195	111	76	112	170	113	32	114	74
115	197	116	214	117	103	118	96	119	160	120	239
121	2	122	81	123	145	124	138	125	27	126	44
127	167	128	209	129	71	130	165	131	46	132	42
133	29	134	9	135	25	136	140	137	95	138	124
139	215	140	136	141	147	142	185	143	150	144	159
145	123	146	104	147	141	148	57	149	55	150	143
151	83	152	176	153	178	154	36	155	14	156	17
157	175	158	90	159	144	160	119	161	3	162	61
163	207	164	169	165	130	166	196	167	127	168	33
169	164	170	112	171	210	172	234	173	202	174	223
175	157	176	152	177	64	178	153	179	206	180	79
181	4	182	49	183	54	184	93	185	142	186	92

Appendix CH 4.2 Result of the multiplicative inverse computation (mod 241,691)

187	58	188	50	189	190	190	189	191	53	192	59				
193	5	194	41	195	110	196	166	197	115	198	28				
199	109	200	47	201	6	202	173	203	19	204	13				
205	87	206	179	207	163	208	73	209	128	210	171				
211	8	212	108	213	43	214	116	215	139	216	106				
217	10	218	220	219	230	220	218	221	12	222	38				
223	174	224	85	225	15	226	16	227	86	228	37				
229	20	230	219	231	24	232	107	233	30	234	172				
235	40	236	48	237	60	238	80	239	120	240	240				

END OF MODULO 241

1	1	2	346	3	461	4	173	5	553	6	576	7	395		
8	432	9	384	10	622	11	377	12	288	13	319	14	543		
15	645	16	216	17	122	18	192	19	291	20	311	21	362		
22	534	23	661	24	144	25	387	26	505	27	128	28	617		
29	143	30	668	31	535	32	108	33	356	34	61	35	79		
36	96	37	635	38	491	39	567	40	501	41	118	42	181		
43	225	44	267	45	215	46	676	47	544	48	72	49	550		
50	539	51	271	52	598	53	339	54	64	55	490	56	654		
57	97	58	417	59	82	60	334	61	34	62	613	63	351		
64	54	65	202	66	178	67	526	68	376	69	681	70	385		
71	146	72	48	73	142	74	663	75	129	76	591	77	350		
78	629	79	35	80	596	81	273	82	59	83	358	84	436		
85	439	86	458	87	278	88	479	89	132	90	453	91	243		
92	338	93	639	94	272	95	611	96	36	97	57	98	275		
99	349	100	615	101	130	102	481	103	530	104	299	105	487		
106	515	107	155	108	32	109	336	110	245	111	442	112	327		
113	159	114	394	115	685	116	554	117	189	118	41	119	511		
120	167	121	474	122	17	123	500	124	652	125	492	126	521		
127	185	128	27	129	75	130	101	131	211	132	89	133	239		
134	263	135	302	136	188	137	575	138	686	139	174	140	538		
141	642	142	73	143	29	144	24	145	305	146	71	147	644		
148	677	149	320	150	410	151	270	152	641	153	551	154	175		
155	107	156	660	157	669	158	363	159	113	160	298	161	588		
162	482	163	496	164	375	165	624	166	179	167	120	168	218		
169	184	170	565	171	493	172	229	173	4	174	139	175	154		
176	585	177	488	178	66	179	166	180	572	181	42	182	467		
183	472	184	169	185	127	186	665	187	388	188	136	189	117		
190	651	191	568	192	18	193	401	194	374	195	528	196	483		
197	228	198	520	199	566	200	653	201	636	202	65	203	514		
204	586	205	300	206	265	207	227	208	495	209	529	210	589		
211	131	212	603	213	279	214	423	215	45	216	16	217	570		
218	168	219	508	220	468	221	222	222	221	223	471	224	509		
225	43	226	425	227	207	228	197	229	172	230	688	231	347		
232	277	233	605	234	440	235	247	236	366	237	242	238	601		
239	133	240	429	241	324	242	237	243	91	244	354	245	110		
246	250	247	235	248	326	249	580	250	246	251	457	252	606		
253	437	254	438	255	607	256	359	257	406	258	383	259	683		
260	396	261	323	262	451	263	134	264	390	265	206	266	465		
267	44	268	477	269	280	270	151	271	51	272	94	273	81		

Appendix CH 4.2 Result of the multiplicative inverse computation mod 691

274	633	275	98	276	343	277	232	278	87	279	213
280	269	281	541	282	321	283	398	284	382	285	434
286	360	287	313	288	12	289	373	290	498	291	19
292	381	293	408	294	322	295	431	296	684	297	577
298	160	299	104	300	205	301	427	302	135	303	504
304	666	305	145	306	621	307	682	308	433	309	407
310	399	311	20	312	330	313	287	314	680	315	623
316	527	317	497	318	402	319	13	320	149	321	282
322	294	323	261	324	241	325	455	326	248	327	112
328	533	329	670	330	312	331	405	332	435	333	608
334	60	335	658	336	109	337	447	338	92	339	53
340	628	341	614	342	592	343	276	344	460	345	689
346	2	347	231	348	415	349	99	350	77	351	63
352	638	353	599	354	244	355	582	356	33	357	631
358	83	359	256	360	286	361	379	362	21	363	158
364	579	365	443	366	236	367	450	368	430	369	397
370	409	371	542	372	678	373	289	374	194	375	164
376	68	377	11	378	404	379	361	380	671	381	292
382	284	383	258	384	9	385	70	386	546	387	25
388	187	389	556	390	264	391	486	392	587	393	531
394	114	395	7	396	260	397	369	398	283	399	310
400	672	401	193	402	318	403	679	404	378	405	331
406	257	407	309	408	293	409	370	410	150	411	422
412	478	413	604	414	459	415	348	416	593	417	58
418	610	419	597	420	640	421	540	422	411	423	214
424	647	425	226	426	485	427	301	428	557	429	240
430	368	431	295	432	8	433	308	434	285	435	332
436	84	437	253	438	254	439	85	440	234	441	445
442	111	443	365	444	456	445	441	446	581	447	337
448	600	449	454	450	367	451	262	452	558	453	90
454	449	455	325	456	444	457	251	458	86	459	414
460	344	461	3	462	519	463	494	464	484	465	266
466	648	467	182	468	220	469	470	470	469	471	223
472	183	473	523	474	121	475	675	476	646	477	268
478	412	479	88	480	560	481	102	482	162	483	196
484	464	485	426	486	391	487	105	488	177	489	626
490	55	491	38	492	125	493	171	494	463	495	208
496	163	497	317	498	290	499	673	500	123	501	40
502	574	503	555	504	303	505	26	506	564	507	522
508	219	509	224	510	649	511	119	512	525	513	625
514	203	515	106	516	537	517	552	518	687	519	462
520	198	521	126	522	507	523	473	524	571	525	512
526	67	527	316	528	195	529	209	530	103	531	393
532	578	533	328	534	22	535	31	536	584	537	516
538	140	539	50	540	421	541	281	542	371	543	14
544	47	545	620	546	386	547	667	548	662	549	618
550	49	551	153	552	517	553	5	554	116	555	503
556	389	557	428	558	452	559	602	560	480	561	590
562	616	563	664	564	506	565	170	566	199	567	39
568	191	569	674	570	217	571	524	572	180	573	650
574	502	575	137	576	6	577	297	578	532	579	364
580	249	581	446	582	355	583	659	584	536	585	176
586	204	587	392	588	161	589	210	590	561	591	76

Appendix CH 4.2 Result of the multiplicative inverse computation (mod 691)

592	342	593	416	594	634	595	655	596	80	597	419
598	52	599	353	600	448	601	238	602	559	603	212
604	413	605	233	606	252	607	255	608	333	609	632
610	418	611	95	612	656	613	62	614	341	615	100
616	562	617	28	618	549	619	643	620	545	621	306
622	10	623	315	624	165	625	513	626	489	627	637
628	340	629	78	630	657	631	357	632	609	633	274
634	594	635	37	636	201	637	627	638	352	639	93
640	420	641	152	642	141	643	619	644	147	645	15
646	476	647	424	648	466	649	510	650	573	651	190
652	124	653	200	654	56	655	595	656	612	657	630
658	335	659	583	660	156	661	23	662	548	663	74
664	563	665	186	666	304	667	547	668	30	669	157
670	329	671	380	672	400	673	499	674	569	675	475
676	46	677	148	678	372	679	403	680	314	681	69
682	307	683	259	684	296	685	115	686	138	687	518
688	230	689	345	690	690						

END OF MODULO 691

Appendix CH 4.2 Result of the multiplicative inverse computation (mod 17011)

```
1     1         2  8506         3 11341         4  4253         5 13609
6 14176         7 14581         8 10632         9 15121        10 15310
11  3093        12  7088        13 14394        14 15796        15 15877

-------------------------------------------------------------------------
1041 13465      1042 10889      1043 14483      1044   277      1045  1286
1046  3464      1047 13924      1048  1120      1049 12065      1050  3386
1051 11864      1052 11594      1053  5218      1054 11572      1055  9739
1056  1627      1057  5842      1058 16191      1059  9140      1060 12341
1061  7776      1062  8057      1063  5665      1064  9161      1065  6453
-------------------------------------------------------------------------

9491 10702      9492 15278      9493  2093      9494 10050      9495 16203
9496  6114      9497 11924      9498 15537      9499  2380      9500  2523
9501  6707      9502  8083      9503 14308      9504  3961      9505  6978
9506 13602      9507 15839      9508  7831      9509  4560      9510 15453
9511 10413      9512  8121      9513 15770      9514  6435      9515 14694
9516  4621      9517  1337      9518 13592      9519  3231      9520 15133
9521 10813      9522  1799      9523   861      9524  7875      9525  5865
9526 12752      9527 13772      9528  4808      9529  4313      9530  5139
9531  8576      9532 11302      9533  5571      9534  8766      9535  3156
9536  8229      9537  1946      9538  4885      9539  9168      9540 14602
9541 13939      9542 12071      9543 15314      9544  1335      9545  4623
9546 10872      9547    98      9548 10998      9549   864      9550  8477
-------------------------------------------------------------------------

14596 13320     14597  1656     14598 11766     14599   134     14600   127
14601  7842     14602  9540     14603  3073     14604 16276     14605  3825
14606   290     14607 14966     14608 15489     14609  4922     14610 15927
14611   645     14612 16933     14613  2873     14614   220     14615 14164
14616  3665     14617  5379     14618  6739     14619 10831     14620  5898
14621  5630     14622 13408     14623  9111     14624  7041     14625 11671
14626  9636     14627  1106     14628  5825     14629 14790     14630  2522
14631  7512     14632 15538     14633  1538     14634  2698     14635  1167
14636  6919     14637  9566     14638  6932     14639  6562     14640   452
14641  9051     14642 10771     14643 11458     14644  9968     14645  5342
14646  2201     14647 10067     14648  9157     14649  2960     14650 15570
14651 15937     14652 16362     14653 12733     14654  7542     14655 14520

-------------------------------------------------------------------------

16956  9588     16957   315     16958  8345     16959  4907     16960 10340
16961 13949     16962 14928     16963 15239     16964 11220     16965  1849
16966   378     16967 11985     16968  1978     16969   405     16970  4149
16971  4678     16972 12213     16973 15668     16974  1839     16975  8978
16976   486     16977 15510     16978 15980     16979 14353     16980 14816
16981   567     16982  7039     16983  9113     16984   630     16985  9814
16986 10887     16987 13467     16988  3698     16989  6959     16990   810
16991  9356     16992 14325     16993   945     16994 14009     16995 11695
16996  1134     16997  1215     16998  2617     16999  9923     17000 13918
17001  1701     17002  1890     17003  6379     17004  2430     17005  2835
17006  3402     17007 12758     17008  5670     17009  8505     17010 17010
                              END OF MODULO  17011
```

Appendix CH4.3 Result of the EC group $E_{17011}(-53,218)$ and message embedment

THE PLAN TEXT MESSAGE TO BE SENT IS

'THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG'

THE CORRESPONDING NUMERIC CODE IS

1507	2704	2625	1201	527	2006
1827	2205	2715	1619	2113	2710
1524	2706	2314	1815	2702	311
2109	2717	605	20		

AND THE ELLIPTIC CURVE CONSTANTS ARE-53 218 17011

WHILE THE RATIONAL CURVE POINTS ARE

2 4835	4 16541	7 3799	8 2632	10 9691
12 10031	13 16051	15 12516	16 12088	18 13186
19 9968	21 8173	22 7965	24 12986	25 12332
27 12618	29 4603	30 11849	33 5600	37 3030
42 1188	43 14253	44 10474	45 15296	46 4691
47 9130	55 7007	56 143	58 13948	59 5048
60 4832	62 10629	63 841	66 8655	67 15587
72 13994	73 14948	74 16998	76 3339	77 10711
78 1959	80 10575	81 13609	83 9230	85 12119
88 13594	90 14153	92 10123	94 11914	95 9634
97 8804	99 9924	101 16674	107 5996	108 14893
109 682	111 5328	112 4525	115 3883	120 14043
121 5789	124 3938	125 11383	126 1629	127 11618
129 5936	132 11640	133 8916	134 10066	140 9048
141 15295	142 14068	143 14046	145 8922	146 11350
149 3948	150 4928	151 13385	152 13877	157 14503
159 6276	162 4225	163 11877	165 6676	166 7629
167 4881	169 3970	171 6117	173 3588	175 9700
176 2179	177 15572	180 10650	183 4108	184 16561
186 14819	188 342	189 1779	193 11651	194 11168
196 6187	199 10962	201 6837	204 6711	208 1398
209 3201	212 12396	214 5649	215 897	218 14209
226 606	234 2647	235 8469	237 10601	243 6211
245 434	246 8513	247 13022	248 8612	250 7835
252 3856	258 11130	260 15862	261 11060	263 4855
264 11142	266 4691	267 16992	270 14767	271 5308
273 9443	274 15473	275 1542	276 14734	278 8813
279 9984	281 6	282 11984	283 12680	285 8079
286 14814	288 2343	291 3938	293 4167	294 8892
295 14108	296 14681	299 16631	300 11615	301 10259
302 14050	305 16967	307 5906	308 15282	311 4188
313 4403	316 16190	318 3951	320 14439	321 2769
322 13689	323 5656	324 7756	325 8314	327 14469
328 8230	331 11943	333 5874	337 6479	338 15026
339 3355	343 11244	344 7401	345 10474	348 1470
349 7473	351 2793	354 9924	356 12616	357 2465
358 16956	359 16067	366 13609	367 13056	368 1322

Appendix CH4.3 Result of the EC group E_{17011}(-53,218) and message embedment

370 12109	372 11155	373 4510	374 13589	375 9100
378 3588	379 9224	380 11350	383 9891	384 8281
385 10375	386 719	387 16922	388 2573	389 3442
391 575	392 10134	393 15627	394 237	395 6881
397 9709	398 2893	399 3001	401 1898	403 5857
406 7040	414 620	418 1723	420 6379	421 13300
422 7671	423 11296	424 14388	425 13156	426 3205
428 9579	429 7341	432 9967	433 11085	434 15579
439 4239	441 16304	442 1460	444 8253	445 3613
447 8034	453 12809	455 15961	457 10251	458 15219
461 15209	465 14601	466 834	467 14555	471 13506
472 14278	474 6485	475 4279	477 15472	480 3653
481 9884	483 6332	485 11127	487 12407	488 14676
490 7425	491 5947	492 8359	493 8159	496 14509
498 13000	499 12521	500 2292	501 4134	505 6623
508 5890	514 12227	515 1755	518 12315	525 14566
526 13960	528 12647	530 3788	532 8494	533 16604
534 4544	535 2413	539 16019	540 1299	543 8086
545 5988	546 15543	548 7955	555 8058	556 2289
558 12411	559 6531	560 5731	563 6420	566 6691
567 10913	569 15423	571 12946	572 8058	574 13876
576 1779	579 8631	582 16382	584 16130	585 2083
590 6699	591 14924	592 5157	593 11489	596 8684
597 4489	602 16569	603 1045	610 3309	615 12835
618 13048	619 11530	620 14954	621 8630	622 16154
623 10387	629 4719	631 12227	632 15291	633 1431
635 13789	639 5983	640 3028	641 4501	643 10231
644 1796	647 10579	652 12027	654 10458	656 14330
658 6102	659 13807	661 3733	662 11433	663 8118
667 8405	670 2946	671 8385	672 11413	674 2477
676 7337	679 8574	680 7145	681 8483	682 9398
683 16547	687 6759	689 729	690 5004	691 10786
692 1641	695 8495	696 14468	697 5883	699 2734
700 11328	701 6882	704 2770	707 4136	708 6171
709 7269	710 16056	714 10028	715 255	720 2258
721 13298	722 790	723 2935	726 6742	727 15657
729 14097	730 14184	732 10893	733 6450	734 3608
736 11932	737 4296	740 9165	744 591	745 11476
747 4680	752 15850	754 12382	757 16619	758 2708
759 3180	760 6242	761 16672	762 16933	763 7056
764 2037	767 8636	769 8655	770 4619	771 10077
776 8477	777 7420	779 4776	781 15698	785 3647
788 13823	789 1603	790 1334	791 3617	792 5486
796 14376	797 16392	798 9529	799 13951	801 11537
803 3246	804 4660	806 7942	808 9237	810 8574
814 6424	819 3298	821 6964	825 13800	828 13771
829 14536	831 5453	832 1570	834 12039	835 14621
836 11694	837 4324	838 14167	840 3389	842 5846
843 7756	847 6199	851 1201	854 6994	855 10410
858 16272	860 14979	864 3943	866 9445	868 2251
869 6208	872 6212	874 6652	875 12125	878 13906
879 15743	880 6549	882 1526	884 1634	886 15470
889 5188	890 16038	892 1286	895 520	898 5230
899 6776	902 5563	903 7416	905 3327	906 14369
907 5409	908 1548	912 16599	913 825	917 5930
918 6832	921 2408	928 11955	929 7577	931 4441
932 7293	934 14158	936 14868	937 8230	938 3896
939 3879	941 8164	944 12938	948 14103	949 14535

Appendix CH4.3 Result of the EC group $E_{17011}(-53,218)$ and message embedment

952 6186	953 7169	954 11503	955 2928	956 5363
959 16476	960 7790	962 546	963 9544	966 4236
967 12957	968 11490	970 9992	971 10193	974 6401
980 12028	981 4965	983 10916	985 16537	986 15688
987 11939	991 6604	993 16522	996 3779	997 8823
998 5812	999 12776	1000 8320	1005 14897	1008 5061
1009 15553	1011 10058	1012 9368	1014 1003	1018 16529
1020 3764	1023 8548	1025 3047	1026 7048	1029 14326
1030 9550	1037 9988	1040 12230	1041 15427	1042 14776
1043 10139	1044 13548	1045 7254	1046 4039	1047 5802
1048 10118	1052 10881	1054 12330	1056 11554	1057 12519
1058 9696	1059 16883	1060 9078	1062 15218	1064 9397
1068 10592	1069 12988	1074 3920	1077 8118	1080 6792
1083 8227	1088 1306	1090 10537	1091 9414	1093 13125
1094 12445	1097 13740	1098 12162	1104 3299	1105 14757
1106 6660	1107 12251	1109 4587	1114 2704	1115 9956
1119 10271	1120 11378	1121 6691	1124 11804	1125 10697
1127 1230	1131 3793	1136 8630	1138 16645	1139 5025
1144 8573	1146 724	1148 3740	1151 7125	1153 10878
1154 13902	1155 110	1158 1290	1159 5364	1162 6467
1168 3792	1169 9397	1172 9579	1174 2490	1176 16785
1179 2223	1181 598	1182 319	1186 4300	1187 2182
1188 9208	1189 7874	1190 5001	1191 15651	1193 11556
1194 10609	1195 15269	1199 6634	1200 336	1204 15324
1206 9317	1207 4961	1209 10139	1211 16616	1214 9163
1215 5802	1220 6093	1224 16633	1227 544	1228 8912
1229 13233	1230 5534	1232 12521	1237 6058	1238 9833
1240 2260	1242 13516	1245 4527	1246 13822	1249 3939
1250 6028	1252 9047	1257 4783	1258 16051	1259 6948
1261 10324	1264 2037	1266 2107	1275 9893	1276 6045
1277 7710	1278 7561	1282 13230	1284 5357	1285 10830
1286 14634	1288 8445	1303 12238	1305 16771	1308 8022
1309 12221	1310 9642	1315 332	1321 4239	1322 724
1324 4545	1327 11751	1329 3802	1330 10691	1332 4888
1334 2440	1336 14718	1338 15264	1339 708	1340 1408
1341 10775	1342 6381	1343 3581	1346 13914	1347 1660
1349 15762	1353 16054	1354 9192	1356 8658	1361 1009
1363 6282	1364 5842	1366 6067	1370 15959	1372 9629
1373 7137	1375 16579	1376 7592	1381 16139	1382 7636
1383 890	1384 4795	1385 14204	1387 7031	1389 14367
1390 11008	1391 12737	1394 3530	1395 1873	1397 14184
1401 6039	1403 9136	1404 9696	1409 4181	1410 7280
1411 8073	1412 1274	1414 7667	1418 9258	1423 2866
1424 12546	1427 12262	1432 5651	1434 14191	1435 10099
1437 4052	1438 6610	1441 12846	1443 2551	1444 5167
1445 5057	1451 9743	1452 5083	1453 12433	1457 2224
1459 16658	1461 693	1463 8539	1468 10049	1469 8361

Appendix CH4.3 Result of the EC group E_{17011}(-53,218) and message embedment

1473 15335	1475 7350	1476 16791	1479 4053	1480 1611
1481 11534	1483 11416	1484 6144	1485 9575	1489 6355
1490 14972	1491 387	1494 13071	1495 5549	1498 1641
1507 16417	1509 4862	1510 13380	1514 13193	1515 10031
1517 2462	1519 8714	1525 4099	1529 8000	1530 11752
1532 15911	1537 8718	1540 513	1543 2573	1544 15455
1547 16042	1548 3447	1550 12304	1553 3529	1554 11978
1555 2671	1559 1110	1561 1854	1564 1522	1566 1840
1568 7273	1570 4279	1571 4354	1572 12544	1575 676
1580 9625	1582 4420	1585 6530	1591 13658	1592 14474
1594 12170	1597 15286	1598 3962	1599 1042	1602 8919
1603 4294	1608 4734	1611 4055	1612 5651	1613 14778
1614 6742	1615 3123	1616 2229	1617 6470	1618 15451
1619 915	1620 3964	1621 8796	1622 4747	1623 11259
1624 11948	1626 15888	1631 4947	1636 5932	1639 5158
1643 6268	1644 5046	1645 5737	1647 15380	1649 12274
1650 1335	1652 7903	1654 3593	1656 525	1657 2874
1658 13973	1661 11076	1663 15140	1664 6761	1665 9136
1667 1949	1668 4863	1669 5610	1670 15863	1674 8445
16192 6143	16193 7706	16194 16383	16195 9998	16197 6644
16198 11008	16199 6232	16200 3112	16201 6355	16203 8048
16206 14527	16207 3203	16209 12724	16211 9032	16212 5882
16213 16743	16215 4549	16225 5482	16227 2653	16230 4195
16233 14880	16236 3648	16240 16499	16241 1927	16242 14621
16243 740	16246 1779	16249 6482	16253 13347	16254 11267
16256 14593	16257 16343	16262 8299	16263 11102	16265 5850
16268 14771	16269 624	16270 7573	16272 3790	16274 3547
16275 7101	16276 9912	16278 15709	16279 9514	16281 2549
16282 4705	16283 14398	16284 6185	16288 7970	16289 13296
16290 1881	16294 37	16295 8294	16296 6737	16297 4753
16298 5963	16302 3579	16304 4020	16305 13937	16307 7746
16308 16214	16312 2226	16313 16921	16316 383	16317 306
16319 12022	16320 1763	16322 9420	16323 14026	16330 6247
16332 6355	16333 1077	16334 9694	16336 2091	16337 16119
16338 9138	16341 11942	16343 3683	16347 1296	16352 2938
16354 15621	16356 5579	16358 15699	16360 12609	16361 11800
16363 11022	16366 4385	16368 1229	16369 9179	16370 3393
16372 3536	16374 5004	16376 12131	16377 4628	16378 16581
16379 4096	16382 12950	16384 8907	16387 10112	16388 12864
16391 13650	16392 16497	16393 14192	16398 10313	16399 11149
16400 1698	16401 11009	16403 121	16407 7939	16408 11834
16414 5702	16417 3629	16419 13616	16422 5224	16425 13683
16427 11002	16428 13657	16430 2954	16433 14520	16434 11008
16437 13916	16439 1230	16440 11278	16447 9773	16450 1007
16452 3669	16455 9811	16456 1230	16459 14311	16460 3588
16461 16062	16462 12012	16463 6591	16464 3929	16466 13797
16476 11722	16481 5681	16485 11350	16487 1021	16488 13788
16492 8030	16493 2616	16496 3657	16499 138	16500 11548
16503 16301	16504 15532	16505 7230	16506 15860	16508 2754
16509 3986	16510 6378	16511 12741	16513 8743	16515 1510
16517 8943	16521 8601	16523 842	16524 13509	16525 7622
16526 10968	16527 4384	16531 13369	16532 11616	16534 8701
16535 16397	16536 15633	16541 3270	16542 12373	16547 8677
16549 14632	16550 3076	16552 8002	16553 2809	16558 9924
16559 16501	16561 6919	16562 7232	16563 14275	16564 13609
16566 11059	16567 6724	16569 8008	16570 1911	16571 11775

Appendix CH4.3 Result of the EC group E_{17011}(-53,218) and message embedment

16576 2530	16578 9630	16580 2033	16581 13714	16584 14417
16585 16418	16587 3279	16588 10552	16590 16308	16592 3248
16594 8636	16596 3938	16598 2971	16600 620	16601 3082
16602 6555	16603 13409	16604 3128	16609 9366	16610 11665
16612 15497	16613 8067	16614 7346	16616 16536	16617 9536
16620 10694	16622 10474	16624 10131	16625 4177	16626 6196
16627 12794	16628 2939	16629 6602	16631 13960	16637 13531
16642 12874	16643 1679	16644 4088	16645 8034	16646 2102
16648 6228	16651 14495	16652 471	16653 4345	16657 12809
16658 7957	16659 9111	16660 6243	16661 8636	16662 3681
16664 12005	16665 14252	16666 3442	16667 11306	16668 14240
16669 2366	16670 2963	16674 2724	16675 2732	16676 5659
16678 7430	16680 13020	16682 8868	16686 12642	16688 1823
16690 14926	16691 8586	16693 1165	16695 15074	16698 6827
16699 4691	16700 5810	16701 6610	16702 6112	16705 3003
16708 3120	16711 5344	16712 8807	16713 14949	16714 13506
16715 15305	16716 14655	16718 574	16719 7685	16723 16225
16726 16733	16727 9877	16728 9243	16730 14567	16731 5189
16733 10277	16735 13998	16738 2948	16740 2983	16744 1023
16746 14813	16747 8771	16748 16143	16750 1133	16751 9181
16752 10237	16753 13336	16754 16564	16758 5669	16760 13905
16763 5975	16764 7017	16767 6030	16768 2933	16769 13174
16770 4227	16772 1410	16773 2216	16775 3192	16776 14916
16778 8386	16779 15431	16784 7131	16785 3269	16787 6268
16791 15911	16792 3380	16794 6233	16795 573	16798 12868
16799 11644	16800 2866	16803 1816	16804 14888	16808 6447
16809 14658	16810 7429	16811 9945	16812 8728	16813 14786
16815 10146	16819 1061	16821 4859	16823 13649	16824 11134
16825 10982	16828 10430	16830 13760	16832 7348	16833 12454
16835 6516	16837 13506	16841 6093	16842 2737	16843 11678
16846 12822	16853 15696	16854 5970	16856 3349	16859 15443
16860 4651	16861 8763	16863 23	16865 13960	16868 1887
16869 15239	16871 4356	16873 14510	16874 1727	16875 5806
16876 3373	16879 4417	16880 1089	16881 2434	16884 14295
16885 2673	16893 1459	16897 8298	16898 9848	16899 13154
16901 16112	16902 3645	16904 8816	16905 12573	16908 371
16911 9937	16912 12809	16913 5161	16918 10323	16919 13887
16920 43	16922 9693	16930 8034	16933 8561	16935 3329
16936 7524	16938 5608	16939 10327	16943 14764	16945 14621
16946 126	16948 6523	16953 4931	16954 3916	16955 816
16958 5004	16960 10592	16961 1553	16967 3442	16976 2660
16979 9335	16980 11794	16982 772	16983 11642	16985 2714
16987 16091	16989 12166	16990 792	16996 3300	17001 7136
17003 16350	17004 14102	17005 15111	17008 620	17010 14564

THE ORDER OF FIELD = 21223

Appendix CH4.3 Result of the EC group E_{17011}(-53,218) and message embedment

THE PLAN NUMERIC TEXT EMBEDDED INTO THE ELLPTIC CURVE GROUP OF POINTS IS

S/N0	RAW MESSAGE	CURVE POINTS EMBEDDMENT
1	1507	1507 16417
2	2704	2704 393
3	2625	2628 2950
4	1201	1204 15324
5	527	528 12647
6	2006	2006 9629
7	1827	1828 1936
8	2205	2205 14215
9	2715	2715 13753
10	1619	1619 915
11	2113	2113 10358
12	2710	2710 401
13	1524	1525 4099
14	2706	2708 11651
15	2314	2314 12327
16	1815	1816 6343
17	2702	2702 11279
18	311	311 4188
19	2109	2109 15967
20	2717	2718 3716
21	605	610 3309
22	20	21 8173

QB 4.5 CODE

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication

---

CURVE CONSTANTS ARE -53 218 17011

SET OF SPECIMEN MULTIPLIERS

822 4 761 4 900 4 1191 4 1289 5 873 4 29 3 156 3
375 4 1379 5 2340 5 2983 5 3399 5 3408 5 3414 5 6085 5
7985 5 5690 5 8387 5 2713 5 619 4

SPECIMEN GENERATOR POINTS

8360 14564 10 9691 4 16541 7 3799 12 10051 1 14209 5 4843
11 2893 23 1029 30 8895 38 5491 60 3457 68 9024 78 3457
88 2689 91 1785 92 9878 96 3457 18 2408

8360 14564

2 3 -2 3
12321 4459 4719 4691 14865 8849 2454 10939
END OF MULTIPLIER MR= 822 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 822 IS 2454 10939

2 2 -3 -2
12321 4459 5751 13213 8843 5660 16672 10659
END OF MULTIPLIER MR= 761 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 761 IS 16672 10659

3 -3 3 -3
5187 16942 3955 14672 7970 2321 5012 15195
END OF MULTIPLIER MR= 900 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 900 IS 5012 15195

3 3 2 1
5187 16942 11383 11726 10002 4151 11724 15967
END OF MULTIPLIER MR= 1191 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1191 IS 11724 15967

1 -3 -2 2 1
8360 14564 5865 16280 4916 3405 13532 13881 978 301
END OF MULTIPLIER MR= 1289 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS 978 301

3 -3 -1 -2
5187 16942 3955 14672 16809 13832 16530 5539
END OF MULTIPLIER MR= 873 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS 16530 5539

1 -3 1
8360 14564 5865 16280 2336 8654
END OF MULTIPLIER MR= 29 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 29 IS 2336 8654

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 3  1  2
5187  16942  14851  7309  800  5192
 END OF MULTIPLIER MR= 156 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 156 IS  800  5192

 1  1 -2 -3
8360  14564  112  16439  2597  4241  1426  6857
 END OF MULTIPLIER MR= 375 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 375 IS  1426  6857


 1 -3  0  1  0
8360  14564  5865  16280  3659  4256  8311  16533  6033  7999
 END OF MULTIPLIER MR= 1379 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1379 IS  6033  7999


 1  0 -1 -2  2
8360  14564  13509  216  12536  6665  3138  12270  5631  10019
  END OF MULTIPLIER MR= 2340 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2340 IS  5631  10019

 1  2 -2 -1  1
8360  14564  4864  10562  2001  1170  994  10094  5533  14546
 END OF MULTIPLIER MR= 2983 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2983 IS  5533  14546

 1  3 -1  3 -3
8360  14564  11908  5548  15826  1651  15718  9868  9214  14664
 END OF MULTIPLIER MR= 3399 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3399 IS  9214  14664

 1  3  0 -3 -1
8360  14564  11908  5548  2346  9029  11179  3793  461  2991
 END OF MULTIPLIER MR= 3408 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3408 IS  461  2991


 1  3  0 -2 -2
8360  14564  11908  5548  2346  9029  8162  9149  8789  8363
 END OF MULTIPLIER MR= 3414 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3414 IS  8789  8363

 3 -3 -2  1  2  5187  16942  3955  14672  11914  4055  15027  6752  7466  2155
 END OF MULTIPLIER MR= 6085 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 6085 IS  7466  2155

 3  2  2  0 -2
5187  16942  2455  11621  5918  7726  506  3481  6476  4466
 END OF MULTIPLIER MR= 7985 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 7985 IS  6476  4466

 2  3 -3  1 -1
12321  4459  4719  4691  5077  4704  8691  5269  6241  6985
 END OF MULTIPLIER MR= 5690 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 5690 IS  6241  6985
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 3  3  3  1  1
5187  16942  11383  11726  8140  11764  2197  4767  1157  5554
 END OF MULTIPLIER MR= 8387 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 8387 IS  1157  5554

 1  1 -1  3 -3
 8360  14564  112  16439  6365  5932  8533  12024  9208  6260
 END OF MULTIPLIER MR= 2713 AND BASE POINT = 8360 ,-14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2713 IS  9208  6260

 2 -1 -3  3
12321  4459  15964  15641  3267  13379  5792  5742
 END OF MULTIPLIER MR= 619 AND BASE POINT = 8360 , 14564
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 619 IS  5792  5742

  _____ END OF GENERATOR POINT 8360  14564 _____

 4  16541

 2  3 -2  3
8935  11733  3354  3608  7516  10630  15981  13868
 END OF MULTIPLIER MR= 822 AND BASE POINT = 4 , 16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 822 IS  15981  13868

 2  2 -3 -2
8935  11733  7596  12659  13351  5612  15377  10730
 END OF MULTIPLIER MR= 761 AND BASE POINT = 4 ,-16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 761 IS  15377  10730

 3 -3  3 -3
5331  10050  3312  932  483  6143  8420  11983
 END OF MULTIPLIER MR= 900 AND BASE POINT = 4 ,-16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 900 IS  8420  11983

 3  3  2  1
5331  10050  10735  6949  15763  10586  3474  11240
 END OF MULTIPLIER MR= 1191 AND BASE POINT = 4 , 16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1191 IS  3474  11240

 1 -3 -2  2  1
4  16541  411  9968  11641  9975  1848  10144  1624  8933
 END OF MULTIPLIER MR= 1289 AND BASE POINT = 4 , 16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS  1624  8933

 3 -3 -1 -2
5331  10050  3312  932  3515  14746  4976  10422
 END OF MULTIPLIER MR= 873 AND BASE POINT = 4 ,-16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS  4976  10422

 1 -3  1
 4  16541  411  9968  4711  13909
 END OF MULTIPLIER MR= 29 AND BASE POINT = 4 , 16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 29 IS  4711  13909
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 3  1  2
5331  10050  2036  758  15576  5789
 END OF MULTIPLIER MR= 156 AND BASE POINT = 4 , 16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 156 IS  15576  5789

 1  1 -2 -3
4  16541  16145  10057  2240  11398  15084  3944
 END OF MULTIPLIER MR= 375 AND BASE POINT = 4 ,-16541
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 375 IS  15084  3944

 1 -3  0  1  0
END OF MULTIPLIER MR= 873 AND BASE POINT = 30 ,-8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS  1395  10257

 1 -3  1
 30  8895  16994  9778  1521  5115
 END OF MULTIPLIER MR= 29 AND BASE POINT = 30 , 8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 29 IS  1521  5115

3  1  2
1738  11383  13635  12385  8913  491
 END OF MULTIPLIER MR= 156 AND BASE POINT = 30 , 8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 156 IS  8913  491

 1  1 -2 -3
 30  8895  11937  1645  14052  5390  5887  11985
 END OF MULTIPLIER MR= 375 AND BASE POINT = 30 ,-8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 375 IS  5887  11985

 1 -3  0  1  0
30  8895  16994  9778  13053  1393  16997  6849  13396  10294
 END OF MULTIPLIER MR= 1379 AND BASE POINT = 30 , 8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1379 IS  13396  10294

 1  0 -1 -2  2
30  8895  2070  13321  10239  14030  450  5033  265  1290
 END OF MULTIPLIER MR= 2340 AND BASE POINT = 30 , 8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2340 IS  265  1290

 1  2 -2 -1  1
30  8895  3739  1881  653  14097  5485  12523  7269  11037
 END OF MULTIPLIER MR= 2983 AND BASE POINT = 30 , 8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2983 IS  7269  11037

 1  3 -1  3 -3
30  8895  143  15246  14579  10382  6179  7198  4369  3510
 END OF MULTIPLIER MR= 3399 AND BASE POINT = 30 ,-8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3399 IS  4369  3510

 1  3  0 -3 -1
 30  8895  143  15246  10252  3373  5392  13787  5261  1960
 END OF MULTIPLIER MR= 3408 AND BASE POINT = 30 ,-8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3408 IS  5261  1960
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 1  3  0 -2 -2
30  8895  143  15246  10252  3373  1183  11734  14429  13906
 END OF MULTIPLIER MR= 3414 AND BASE POINT = 30 ,-8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3414 IS  14429  13906


 3 -3 -2  1  2
 1738  11383  10172  12507  1375  6178  13264  418  13612  12252
 END OF MULTIPLIER MR= 6085 AND BASE POINT = 30 , 8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 6085 IS  13612  12252

 3  2  2  0 -2
1738  11383  3550  14240  14322  8748  14901  11573  5961  13507
 END OF MULTIPLIER MR= 7985 AND BASE POINT = 30 ,-8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 7985 IS  5961  13507

 2  3 -3  1 -1
 4004  16253  2937  2358  3546  5137  1916  12499  3515  2009
 END OF MULTIPLIER MR= 5690 AND BASE POINT = 30 ,-8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 5690 IS  3515  2009

 3  3  3  1  1
1738  11383  2457  13057  12403  7402  7390  9771  14431  14474
 END OF MULTIPLIER MR= 8387 AND BASE POINT = 30 , 8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 8387 IS  14431  14474

 1  1 -1  3 -3
30  8895  11937  1645  10744  9530  9518  11740  5410  5348
 END OF MULTIPLIER MR= 2713 AND BASE POINT = 30 ,-8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2713 IS  5410  5348

 2 -1 -3  3
4004  16253  4760  6484  6444  10939  3412  745
 END OF MULTIPLIER MR= 619 AND BASE POINT = 30 , 8895
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 619 IS  3412  745

  _____ END OF GENERATOR POINT 30  8895 _____

 88  2689

 2  3 -2  3
5404  15652  3267  16648  8097  6169  11495  2864
 END OF MULTIPLIER MR= 822 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 822 IS  11495  2864

 2  2 -3 -2
 5404  15652  1172  2340  13295  7807  2812  1352
 END OF MULTIPLIER MR= 761 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 761 IS  2812  1352

 3 -3  3 -3
9754  5977  907  11903  16022  10523  8157  666
 END OF MULTIPLIER MR= 900 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 900 IS  8157  666
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

3 3 2 1
9754 5977 8941 8768 7511 15485 5226 392
END OF MULTIPLIER MR= 1191 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1191 IS 5226 392

1 -3 -2 2 1
88 2689 9730 1127 16393 4567 1249 8649 1473 2555
END OF MULTIPLIER MR= 1289 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS 1473 2555

3 -3 -1 -2
9754 5977 907 11903 3774 2011 697 16142
END OF MULTIPLIER MR= 873 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS 697 16142

1 -3 1
88 2689 9730 1127 8225 578
END OF MULTIPLIER MR= 29 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 29 IS 8225 578

3 1 2
9754 5977 209 14395 3559 15805
END OF MULTIPLIER MR= 156 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 156 IS 3559 15805

1 1 -2 -3
88 2689 12756 382 14667 5069 1139 16598
END OF MULTIPLIER MR= 375 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 375 IS 1139 16598

1 -3 0 1 0
88 2689 9730 1127 1929 13425 3039 4088 9468 10752
END OF MULTIPLIER MR= 1379 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1379 IS 9468 10752

1 0 -1 -2 2
88 2689 16274 8968 5831 16684 6134 12008 3821 3101
END OF MULTIPLIER MR= 2340 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2340 IS 3821 3101

1 2 -2 -1 1
88 2689 5586 16701 3790 16505 6793 10950 4207 6834
END OF MULTIPLIER MR= 2983 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2983 IS 4207 6834

1 3 -1 3 -3
88 2689 4955 14647 16552 665 12788 4363 7853 12238
END OF MULTIPLIER MR= 3399 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3399 IS 7853 12238

1 3 0 -3 -1
88 2689 4955 14647 14671 2501 9379 4115 12107 3603
END OF MULTIPLIER MR= 3408 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3408 IS 12107 3603

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 1  3  0 -2 -2
88  2689  4955  14647  14671  2501  7687  7630  7490  10920
 END OF MULTIPLIER MR= 3414 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3414 IS  7490  10920

 3 -3 -2  1  2
 9754  5977  907  11903  8584  2175  4419  15879  16166  12830
 END OF MULTIPLIER MR= 6085 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 6085 IS  16166  12830

 3  2  2  0 -2
 9754  5977  9110  15668  5717  1074  11336  11361  387  3715
 END OF MULTIPLIER MR= 7985 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 7985 IS  387  3715
 2  3 -3  1 -1  5404  15652  3267  16648  13123  8089  11025  6347  14934  12862
 END OF MULTIPLIER MR= 5690 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 5690 IS  14934  12862

 3  3  3  1  1
 9754  5977  8941  8768  4748  1245  10675  12032  15996  3822
 END OF MULTIPLIER MR= 8387 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 8387 IS  15996  3822

 1  1 -1  3 -3
 88  2689  12756  382  2985  4147  304  13892  6823  16022
 END OF MULTIPLIER MR= 2713 AND BASE POINT = 88 ,-2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2713 IS  6823  16022

 2 -1 -3  3
 5404  15652  10348  5514  14362  11355  7035  2943
 END OF MULTIPLIER MR= 619 AND BASE POINT = 88 , 2689
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 619 IS  7035  2943

  _____ END OF GENERATOR POINT 88  2689 _____

 91  1785

 2  3 -2  3
1887  8282  7403  15727  2251  10863  296  3149
 END OF MULTIPLIER MR= 822 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 822 IS  296  3149

 2  2 -3 -2
1887  8282  16815  8000  9379  6809  7716  13541
 END OF MULTIPLIER MR= 761 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 761 IS  7716  13541

 3 -3  3 -3
5930  4115  3969  5254  11111  11159  11191  6136
 END OF MULTIPLIER MR= 900 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 900 IS  11191  6136
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 3  3  2  1
5930  4115  5721  1635  8781  4364  2829  10713
 END OF MULTIPLIER MR= 1191 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1191 IS  2829  10713

 1 -3 -2  2  1
 91  1785  14413  3760  8792  11751  5040  11275  4836  12120
 END OF MULTIPLIER MR= 1289 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS  4836  12120

 3 -3 -1 -2
5930  4115  3969  5254  4054  8829  8259  5199
 END OF MULTIPLIER MR= 873 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS  8259  5199

 1 -3  1
 91  1785  14413  3760  4811  832
 END OF MULTIPLIER MR= 29 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 29 IS  4811  832

 3  1  2
5930  4115  2857  431  10248  11847
 END OF MULTIPLIER MR= 156 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 156 IS  10248  11847


 1  1 -2 -3
91  1785  1180  228  8987  14622  12756  4988
 END OF MULTIPLIER MR= 375 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 375 IS  12756  4988

 1 -3  0  1  0
91  1785  14413  3760  126  11082  10756  9684  14991  5822
 END OF MULTIPLIER MR= 1379 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1379 IS  14991  5822

 1  0 -1 -2  2
91  1785  14978  11897  9168  5675  16459  16532  7429  2294
 END OF MULTIPLIER MR= 2340 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2340 IS  7429  2294

 1  2 -2 -1  1
91  1785  2944  15931  15157  2515  9541  4304  8896  11342
 END OF MULTIPLIER MR= 2983 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2983 IS  8896  11342

 1  3 -1  3 -3
91  1785  9404  2666  16786  4886  7235  16487  6470  4604
 END OF MULTIPLIER MR= 3399 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3399 IS  6470  4604

 1  3  0 -3 -1
91  1785  9404  2666  5643  4680  6363  566  12321  7703
 END OF MULTIPLIER MR= 3408 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3408 IS  12321  7703
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 1  3  0 -2 -2
 91  1785  9404  2666  5643  4680  13064  15065  7917  9912
 END OF MULTIPLIER MR= 3414 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3414 IS  7917  9912

  3 -3 -2  1  2
5930  4115  3969  5254  10391  1267  13423  12281  5889  1353
 END OF MULTIPLIER MR= 6085 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 6085 IS  5889  1353

 3  2  2  0 -2
 5930  4115  15662  14927  5553  8846  5717  16835  9464  7829
 END OF MULTIPLIER MR= 7985 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 7985 IS  9464  7829

 2  3 -3  1 -1
 1887  8282  7403  15727  8111  15307  4169  16689  7883  3198
 END OF MULTIPLIER MR= 5690 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 5690 IS  7883  3198

 3  3  3  1  1
5930  4115  5721  1635  2077  11806  9367  2621  15616  13624
 END OF MULTIPLIER MR= 8387 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 8387 IS  15616  13624

 1  1 -1  3 -3
 91  1785  1180  228  15459  16287  456  13699  10641  12482
 END OF MULTIPLIER MR= 2713 AND BASE POINT = 91 ,-1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2713 IS  10641  12482

 2 -1 -3  3
1887  8282  4548  3770  5936  12716  3476  10501
 END OF MULTIPLIER MR= 619 AND BASE POINT = 91 , 1785
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 619 IS  3476  10501

  _____ END OF GENERATOR POINT 91  1785 _____

 92  9878

 2  3 -2  3
 9503  16340  1908  4216  11126  318  14479  13052
 END OF MULTIPLIER MR= 822 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 822 IS  14479  13052

 2  2 -3 -2
9503  16340  530  10338  51  2482  2857  5214
 END OF MULTIPLIER MR= 761 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 761 IS  2857  5214

 3 -3  3 -3
1251  1229  4851  6852  7928  5280  13826  5924
 END OF MULTIPLIER MR= 900 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 900 IS  13826  5924
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 3  3  2  1
1251  1229  9763  9457  4956  5326  5010  16494
 END OF MULTIPLIER MR= 1191 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1191 IS  5010  16494

 1 -3 -2  2  1
 92  9878  13861  4530  14526  8091  10316  4131  14833  11688
 END OF MULTIPLIER MR= 1289 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS  14833  11688

 3 -3 -1 -2
1251  1229  4851  6852  16467  742  16301  2936
 END OF MULTIPLIER MR= 873 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS  16301  2936

 1 -3  1
92  9878  13861  4530  7124  11261
 END OF MULTIPLIER MR= 29 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 29 IS  7124  11261

  3  1  2
1251  1229  4263  16759  7320  15798
 END OF MULTIPLIER MR= 156 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 156 IS  7320  15798

 1  1 -2 -3
 92  9878  826  7662  4213  5132  9413  14815
 END OF MULTIPLIER MR= 375 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 375 IS  9413  14815
 1 -3  0  1  0
92  9878  13861  4530  4519  10066  11144  6602  12179  8925
 END OF MULTIPLIER MR= 1379 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1379 IS  12179  8925

 1  0 -1 -2  2
92  9878  5364  8396  9217  9760  7851  13078  858  8979
 END OF MULTIPLIER MR= 2340 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2340 IS  858  8979


 1  2 -2 -1  1
 92  9878  2059  3245  1833  11475  6663  9120  12251  9335
 END OF MULTIPLIER MR= 2983 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2983 IS  12251  9335

 1  3 -1  3 -3
92  9878  2060  10042  409  11553  1832  11108  16111  8105
 END OF MULTIPLIER MR= 3399 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3399 IS  16111  8105

 1  3  0 -3 -1
92  9878  2060  10042  13005  5014  16468  6702  11031  16895
 END OF MULTIPLIER MR= 3408 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3408 IS  11031  16895
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 1  3  0 -2 -2
92  9878  2060  10042  13005  5014  16554  13924  9238  16463
 END OF MULTIPLIER MR= 3414 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3414 IS  9238  16463

 3 -3 -2  1  2
1251  1229  4851  6852  9851  1762  15375  13311  10374  3460
 END OF MULTIPLIER MR= 6085 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 6085 IS  10374  3460

 3  2  2  0 -2
1251  1229  3803  16884  836  11209  3141  11189  9520  2041
 END OF MULTIPLIER MR= 7985 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 7985 IS  9520  2041

 2  3 -3  1 -1
 9503  16340  1908  4216  11748  11191  420  5460  5586  9914
 END OF MULTIPLIER MR= 5690 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 5690 IS  5586  9914

 3  3  3  1  1
1251  1229  9763  9457  1841  15065  7106  6850  852  16027
 END OF MULTIPLIER MR= 8387 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 8387 IS  852  16027

 1  1 -1  3 -3
 92  9878  826  7662  6254  4141  7875  9314  2396  8687
 END OF MULTIPLIER MR= 2713 AND BASE POINT = 92 ,-9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2713 IS  2396  8687

 2 -1 -3  3
9503  16340  8729  6274  844  12794  12626  5416
 END OF MULTIPLIER MR= 619 AND BASE POINT = 92 , 9878
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 619 IS  12626  5416

  _____ END OF GENERATOR POINT 92  9878 _____

 18  2408

 2  3 -2  3
4236  3689  15176  12584  12005  2759  5155  11287
 END OF MULTIPLIER MR= 822 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 822 IS  5155  11287

 2  2 -3 -2
4236  3689  7659  2767  726  11  2177  11993
 END OF MULTIPLIER MR= 761 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 761 IS  2177  11993

 3 -3  3 -3
9277  15578  3690  14636  5382  7774  10002  9551
 END OF MULTIPLIER MR= 900 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 900 IS  10002  9551
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 3  3  2  1
9277  15578  4160  14454  15556  15064  12881  12300
 END OF MULTIPLIER MR= 1191 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1191 IS  12881  12300

 1 -3 -2  2  1
18  2408  4465  789  15758  11849  504  6899  6914  3957
 END OF MULTIPLIER MR= 1289 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS  6914  3957

 3 -3 -1 -2
9277  15578  3690  14636  1335  5827  2403  16428
 END OF MULTIPLIER MR= 873 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS  2403  16428

 1 -3  1
 18  2408  4465  789  11125  6856
 END OF MULTIPLIER MR= 29 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 29 IS  11125  6856

 3  1  2
9277  15578  8507  3737  9911  6730
 END OF MULTIPLIER MR= 156 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 156 IS  9911  6730

 1  1 -2 -3
18  2408  14126  5620  15776  6899  3548  2588
 END OF MULTIPLIER MR= 375 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 375 IS  3548  2588

 1 -3  0  1  0
18  2408  4465  789  16053  8898  15305  14624  1758  15529
 END OF MULTIPLIER MR= 1379 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1379 IS  1758  15529

 1  0 -1 -2  2
 18  2408  13382  4382  16713  8605  16951  15726  14683  1673
 END OF MULTIPLIER MR= 2340 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2340 IS  14683  1673

 1  2 -2 -1  1
18  2408  12341  5947  8710  11487  1591  7069  3114  8783
 END OF MULTIPLIER MR= 2983 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2983 IS  3114  8783

 1  3 -1  3 -3
18  2408  13172  12740  4173  7271  10055  13188  10474  10315
 END OF MULTIPLIER MR= 3399 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3399 IS  10474  10315

 1  3  0 -3 -1
18  2408  13172  12740  3669  6289  14671  11  12906  6345
 END OF MULTIPLIER MR= 3408 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3408 IS  12906  6345
```

Appendix CH4.4 Result of the RR7SQSD addition/subtraction chain based EC point multiplication continued

```
 1  3  0 -2 -2
18  2408  13172  12740  3669  6289  9714  16953  7786  1052
 END OF MULTIPLIER MR= 3414 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 3414 IS  7786  1052

 3 -3 -2  1  2
9277  15578  3690  14636  4625  16613  12875  7899  4573  3182
 END OF MULTIPLIER MR= 6085 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 6085 IS  4573  3182

 3  2  2  0 -2
9277  15578  1899  13010  10700  10109  5170  8373  7200  3285
 END OF MULTIPLIER MR= 7985 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 7985 IS  7200  3285

 2  3 -3  1 -1
4236  3689  15176  12584  13437  16727  3858  1908  11906  12931
 END OF MULTIPLIER MR= 5690 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 5690 IS  11906  12931

 3  3  3  1  1
9277  15578  4160  14454  2691  15708  14044  16810  3315  15932
 END OF MULTIPLIER MR= 8387 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 8387 IS  3315  15932

  1  1 -1  3 -3
18  2408  14126  5620  11963  8121  15813  554  10108  23
 END OF MULTIPLIER MR= 2713 AND BASE POINT = 18 ,-2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 2713 IS  10108  23

 2 -1 -3  3
4236  3689  8738  10024  11493  9021  13694  16032
 END OF MULTIPLIER MR= 619 AND BASE POINT = 18 , 2408
AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 619 IS  13694  16032

  _____ END OF GENERATOR POINT 18  2408 _____
```

QB 4.5 CODE

Appendix CH4.5 Results of the message encryption/decryption program

```
 THE PLAN TEXT MESSAGE FOR THE CRYPTOGRAPHIC EXERCISE IS

        'THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG'
 THE CORRESPONDING NUMERIC CODE OF THE MESSAGE TO BE TRANSMITTED

1507          2704          2625          1201          527           2006
1827          2205          2715          1619          2113          2710
1524          2706          2314          1815          2702          311
2109         2717           605           20
     CURVE CONSTANTS ARE-53  218  17011
```

THE QUADRATIC RESIDUES OF THE ELLPTIC CURVE EQUATION ARE:-

1	4	9	16	25	36	49	64
81	100	121	144	169	196	225	256
289	324	361	400	441	484	529	576
625	676	729	784	841	900	961	1024
1089	1156	1225	1296	1369	1444	1521	1600
1681	1764	1849	1936	2025	2116	2209	2304
2401	2500	2601	2704	2809	2916	3025	3136
3249	3364	3481	3600	3721	3844	3969	4096
4225	4356	4489	4624	4761	4900	5041	5184
5329	5476	5625	5776	5929	6084	6241	6400
6561	6724	6889	7056	7225	7396	7569	7744
7921	8100	8281	8464	8649	8836	9025	9216
9409	9604	9801	10000	10201	10404	10609	10816
11025	11236	11449	11664	11881	12100	12321	12544
12769	12996	13225	13456	13689	13924	14161	14400
14641	14884	15129	15376	15625	15876	16129	16384
16641	16900	150	413	678	945	1214	1485
1758	2033	2310	2589	2870	3153	3438	3725
4014	4305	4598	4893	5190	5489	5790	6093
6398	6705	7014	7325	7638	7953	8270	8589
8910	9233	9558	9885	10214	10545	10878	11213
11550	11889	12230	12573	12918	13265	13614	13965
14318	14673	15030	15389	15750	16113	16478	16845
203	574	947	1322	1699	2078	2459	2842
3227	3614	4003	4394	4787	5182	5579	5978
6379	6782	7187	7594	8003	8414	8827	9242
9659	10078	10499	10922	11347	11774	12203	12634
13067	13502	13939	14378	14819	15262	15707	16154
16603	43	496	951	1408	1867	2328	2791
3256	3723	4192	4663	5136	5611	6088	6567
7048	7531	8016	8503	8992	9483	9976	10471
10968	11467	11968	12471	12976	13483	13992	14503
15016	15531	16048	16567	77	600	1125	1652
2181	2712	3245	3780	4317	4856	5397	5940
6485	7032	7581	8132	8685	9240	9797	10356
10917	11480	12045	12612	13181	13752	14325	14900
15477	16056	16637	209	794	1381	1970	2561
3154	3749	4346	4945	5546	6149	6754	7361
7970	8581	9194	9809	10426	11045	11666	12289
12914	13541	14170	14801	15434	16069	16706	334
975	1618	2263	2910	3559	4210	4863	5518
6175	6834	7495	8158	8823	9490	10159	10830
11503	12178	12855	13534	14215	14898	15583	16270

Appendix CH4.5 Results of the message encryption/decryption program

```
16959     639     1332     2027     2724     3423     4124     4827
5532     6239     6948     7659     8372     9087     9804    10523
11244    11967    12692    13419    14148    14879    15612    16347
73       812      1553     2296     3041     3788     4537     5288
6041     6796     7553     8312     9073     9836    10601    11368
12137    12908    13681    14456    15233    16012    16793      565
1350     2137     2926     3717     4510     5305     6102     6901
7702     8505     9310    10117    10926    11737    12550    13365
14182    15001    15822    16645      459     1286     2115     2946
3779     4614     5451     6290     7131     7974     8819     9666
10515    11366    12219    13074    13931    14790    15651    16514
368      1235     2104     2975     3848     4723     5600     6479
7360     8243     9128    10015    10904    11795    12688    13583
14480    15379    16280      172     1077     1984     2893     3804
4717     5632     6549     7468     8389     9312    10237    11164
12093    13024    13957    14892    15829    16768      698     1641
2586     3533     4482     5433     6386     7341     8298     9257
10218    11181    12146    13113    14082    15053    16026    17001
967      1946     2927     3910     4895     5882     6871     7862
8855     9850    10847    11846    12847    13850    14855    15862
16871     871     1884     2899     3916     4935     5956     6979
8004     9031    10060    11091    12124    13159    14196    15235
16276     308     1353     2400     3449     4500     5553     6608
7665     8724     9785    10848    11913    12980    14049    15120
16193     257     1334     2413     3494     4577     5662     6749
7838     8929    10022    11117    12214    13313    14414    15517
16622     718     1827     2938     4051     5166     6283     7402
8523     9646    10771    11898    13027    14158    15291    16426
552      1691     2832     3975     5120     6267     7416     8567
9720    10875    12032    13191    14352    15515    16680      836
2005     3176     4349     5524     6701     7880     9061    10244
11429    12616    13805    14996    16189      373     1570     2769
3970     5173     6378     7585     8794    10005    11218    12433
13650    14869    16090      302     1527     2754     3983     5214
6447     7682     8919    10158    11399    12642    13887    15134
16383     623     1876     3131     4388     5647     6908     8171
9436    10703    11972    13243    14516    15791       57     1336
2617     3900     5185     6472     7761     9052    10345    11640
12937    14236    15537    16840     1134     2441     3750     5061
6374     7689     9006    10325    11646    12969    14294    15621
16950     1270     2603     3938     5275     6614     7955     9298
10643    11990    13339    14690    16043      387     1744     3103
4464     5827     7192     8559     9928    11299    12672    14047
15424    16803     1173     2556     3941     5328     6717     8108
9501    10896    12293    13692    15093    16496      890     2297
3706     5117     6530     7945     9362    10781    12202    13625
15050    16477      895     2326     3759     5194     6631     8070
9511    10954    12399    13846    15295    16746     1188     2643
4100     5559     7020     8483     9948    11415    12884    14355
15828     292     1769     3248     4729     6212     7697     9184
10673    12164    13657    15152    16649     1137     2638     4141
5646     7153     8662    10173    11686    13201    14718    16237
747      2270     3795     5322     6851     8382     9915    11450
12987    14526    16067      599     2144     3691     5240     6791
8344     9899    11456    13015    14576    16139      693     2260
3829     5400     6973     8548    10125    11704    13285    14868
```

Appendix CH4.5 Results of the message encryption/decryption program

```
16453     1029     2618     4209     5802     7397     8994    10593
12194    13797    15402    17009     1607     3218     4831     6446
 8063     9682    11303    12926    14551    16178      796     2427
 4060     5695     7332     8971    10612    12255    13900    15547
  185     1836     3489     5144     6801     8460    10121    11784
13449    15116    16785     1445     3118     4793     6470     8149
 9830    11513    13198    14885    16574     1254     2947     4642
 6339     8038     9739    11442    13147    14854    16563     1263
 2976     4691     6408     8127     9848    11571    13296    15023
16752     1472     3205     4940     6677     8416    10157    11900
13645    15392      130     1881     3634     5389     7146     8905
10666    12429    14194    15961      719     2490     4263     6038
 7815     9594    11375    13158    14943    16730     1508     3299
 5092     6887     8684    10483    12284    14087    15892      688
 2497     4308     6121     7936     9753    11572    13393    15216
   30     1857     3686     5517     7350     9185    11022    12861
14702    16545     1379     3226     5075     6926     8779    10634
12491    14350    16211     1063     2928     4795     6664     8535
10408    12283    14160    16039      909     2792     4677     6564
 8453    10344    12237    14132    16029      917     2818     4721
 6626     8533    10442    12353    14266    16181     1087     3006
 4927     6850     8775    10702    12631    14562    16495     1419
 3356     5295     7236     9179    11124    13071    15020    16971
 1913     3868     5825     7784     9745    11708    13673    15640
  598     2569     4542     6517     8494    10473    12454    14437
16422     1398     3387     5378     7371     9366    11363    13362
15363      355     2360     4367     6376     8387    10400    12415
14432    16451     1461     3484     5509     7536     9565    11596
13629    15664      690     2729     4770     6813     8858    10905
12954    15005       47     2102     4159     6218     8279    10342
12407    14474    16543     1603     3676     5751     7828     9907
11988    14071    16156     1232     3321     5412     7505     9600
11697    13796    15897      989     3094     5201     7310     9421
11534    13649    15766      874     2995     5118     7243     9370
11499    13630    15763      887     3024     5163     7304     9447
11592    13739    15888     1028     3181     5336     7493     9652
11813    13976    16141     1297     3466     5637     7810     9985
12162    14341    16522     1694     3879     6066     8255    10446
12639    14834       20     2219     4420     6623     8828    11035
13244    15455      657     2872     5089     7308     9529    11752
13977    16204     1422     3653     5886     8121    10358    12597
14838       70     2315     4562     6811     9062    11315    13570
15827     1075     3336     5599     7864    10131    12400    14671
16944     2208     4485     6764     9045    11328    13613    15900
 1178     3469     5762     8057    10354    12653    14954      246
 2551     4858     7167     9478    11791    14106    16423     1731
 4052     6375     8700    11027    13356    15687     1009     3344
 5681     8020    10361    12704    15049      385     2734     5085
 7438     9793    12150    14509    16870     2222     4587     6954
 9323    11694    14067    16442     1808     4187     6568     8951
11336    13723    16112     1492     3885     6280     8677    11076
13477    15880     1274     3681     6090     8501    10914    13329
15746     1154     3575     5998     8423    10850    13279    15710
 1132     3567     6004     8443    10884    13327    15772     1208
 3657     6108     8561    11016    13473    15932     1382     3845
 6310     8777    11246    13717    16190     1654     4131     6610
 9091    11574    14059    16546     2024     4515     7008     9503
```

Appendix CH4.5 Results of the message encryption/decryption program

```
12000   14499   17000    2492    4997    7504   10013   12524
15037     541    3058    5577    8098   10621   13146   15673
 1191    3722    6255    8790   11327   13866   16407    1939
 4484    7031    9580   12131   14684     228    2785    5344
 7905   10468   13033   15600    1158    3729    6302    8877
11454   14033   16614    2186    4771    7358    9947   12538
15131     715    3312    5911    8512   11115   13720   16327
 1925    4536    7149    9764   12381   15000     610    3233
 5858    8485   11114   13745   16378    2002    4639    7278
 9919   12562   15207     843    3492    6143    8796   11451
14108   16767    2417    5080    7745   10412   13081   15752
 1414    4089    6766    9445   12126   14809     483    3170
 5859    8550   11243   13938   16635    2323    5024    7727
10432   13139   15848    1548    4261    6976    9693   12412
15133     845    3570    6297    9026   11757   14490     214
 2951    5690    8431   11174   13919   16666    2404    5155
 7908   10663   13420   16179    1929    4692    7457   10224
12993   15764    1526    4301    7078    9857   12638   15421
 1195    3982    6771    9562   12355   15150     936    3735
 6536    9339   12144   14951     749    3560    6373    9188
12005   14824     634    3457    6282    9109   11938   14769
  591    3426    6263    9102   11943   14786     620    3467
 6316    9167   12020   14875     721    3580    6441    9304
12169   15036     894    3765    6638    9513   12390   15269
 1139    4022    6907    9794   12683   15574    1456    4351
 7248   10147   13048   15951    1845    4752    7661   10572
13485   16400    2306    5225    8146   11069   13994   16921
 2839    5770    8703   11638   14575     503    3444    6387
 9332   12279   15228    1168    4121    7076   10033   12992
15953    1905    4870    7837   10806   13777   16750    2714
 5691    8670   11651   14634     608    3595    6584    9575
12568   15563    1549    4548    7549   10552   13557   16564
 2562    5573    8586   11601   14618     626    3647    6670
 9695   12722   15751    1771    4804    7839   10876   13915
16956    2988    6033    9080   12129   15180    1222    4277
 7334   10393   13454   16517    2571    5638    8707   11778
14851     915    3992    7071   10152   13235   16320    2396
 5485    8576   11669   14764     850    3949    7050   10153
13258   16365    2463    5574    8687   11802   14919    1027
 4148    7271   10396   13523   16652    2772    5905    9040
12177   15316    1446    4589    7734   10881   14030     170
 3323    6478    9635   12794   15955    2107    5272    8439
11608   14779     941    4116    7293   10472   13653   16836
 3010    6197    9386   12577   15770    1954    5151    8350
11551   14754     948    4155    7364   10575   13788   17003
 3209    6428    9649   12872   16097    2313    5542    8773
12006   15241    1467    4706    7947   11190   14435     671
 3920    7171   10424   13679   16936    3184    6445    9708
12973   16240    2498    5769    9042   12317   15594    1862
 5143    8426   11711   14998    1276    4567    7860   11155
14452     740    4041    7344   10649   13956     254    3565
 6878   10193   13510   16829    3139    6462    9787   13114
16443    2763    6096    9431   12768   16107    2437    5780
 9125   12472   15821    2161    5514    8869   12226   15585
 1935    5298    8663   12030   15399    1759    5132    8507
11884   15263    1633    5016    8401   11788   15177    1557
 4950    8345   11742   15141    1531    4934    8339   11746
```

Appendix CH4.5 Results of the message encryption/decryption program

```
 8133    15384     5626    12881     3127    10386      636     7899
15164     5420    12689     2949    10222      486     7763    15042
 5312    12595     2869    10156      434     7725    15018     5302
12599     2887    10188      480     7785    15092     5390    12701
 3003    10318      624     7943    15264     5576    12901     3217
10546      866     8199    15534     5860    13199     3529    10872
 1206     8553    15902     6242    13595     3939    11296     1644
 9005    16368     6722    14089     4447    11818     2180     9555
16932     7300    14681     5053    12438     2814    10203      583
 7976    15371     5757    13156     3546    10949     1343     8750
16159     6559    13972     4376    11793     2201     9622       34
 7459    14886     5304    12735     3157    10592     1018     8457
15898     6330    13775     4211    11660     2100     9553    17008
 7454    14913     5363    12826     3280    10747     1205     8676
16149     6613    14090     4558    12039     2511     9996      472
 7961    15452     5934    13429     3915    11414     1904     9407
16912     7408    14917     5417    12930     3434    10951     1459
 8980    16503     7017    14544     5062    12593     3115    10650
 1176     8715    16256     6788    14333     4869    12418     2958
10511     1055     8612    16171     6721    14284     4838    12405
 2963    10534     1096     8671    16248     6816    14397     4969
12554     3130    10719     1299     8892    16487     7073    14672
 5262    12865     3459    11066     1664     9275    16888     7492
15109     5717    13338     3950    11575     2191     9820      440
 8073    15708     6334    13973     4603    12246     2880    10527
 1165     8816    16469     7113    14770     5418    13079     3731
11396     2052     9721      381     8054    15729     6395    14074
 4744    12427     3101    10788     1466     9157    16850     7534
15231     5919    13620     4312    12017     2713    10422     1122
 8835    16550     7256    14975     5685    13408     4122    11849
 2567    10298     1020     8755    16492     7220    14961     5693
13438     4174    11923     2663    10416     1160     8917    16676
 7426    15189     5943    13710     4468    12239     3001    10776
 1542     9321       91     7874    15659     6435    14224     5004
12797     3581    11378     2166     9967      759     8564    16371
 7169    14980     5782    13597     4403    12222     3032    10855
 1669     9496      314     8145    15978     6802    14639     5467
13308     4140    11985     2821    10670     1510     9363      207
 8064    15923     6773    14636     5490    13357     4215    12086
 2948    10823     1689     9568      438     8321    16206     7082
14971     5851    13744     4628    12525     3413    11314     2206
10111     1007     8916    16827     7729    15644     6550    14469
 5379    13302     4216    12143     3061    10992     1914     9849
  775     8714    16655     7587    15532     6468    14417     5357
13310     4254    12211     3159    11120     2072    10037      993
 8962    16933     7895    15870     6836    14815     5785    13768
 4742    12729     3707    11698     2680    10675     1661     9660
  650     8653    16658     7654    15663     6663    14676     5680
13697     4705    12726     3738    11763     2779    10808     1828
 9861      885     8922    16961     7991    16034     7068    15115
 6153    14204     5246    13301     4347    12406     3456    11519
 2573    10640     1698     9769      831     8906    16983     8051
16132     7204    15289     6365    14454     5534    13627     4711
12808     3896    11997     3089    11194     2290    10399     1499
 9612      716     8833    16952     8062    16185     7299    15426
 6544    14675     5797    13932     5058    13197     4327    12470
 3604    11751     2889    11040     2182    10337     1483     9642
```

Appendix CH4.5 Results of the message encryption/decryption program

```
14258      7330       404     10491      3569     13660      6742     16837
 9923      3011     13112      6204     16309      9405      2503     12614
 5716     15831      8937      2045     12166      5278     15403      8519
 1637     11768      4890     15025      8151      1279     11420      4552
14697      7833       971     11122      4264     14419      7565       713
10874      4026     14191      7347       505     10676      3838     14013
 7179       347     10528      3700     13885      7061       239     10430
 3612     13807      6993       181     10382      3574     13779      6975
  173     10384      3586     13801      7007       215     10436      3648
13873      7089       307     10538      3760     13995      7221       449
10690      3922     14167      7403       641     10892      4134     14389
 7635       883     11144      4396     14661      7917      1175     11446
 4708     14983      8249      1517     11798      5070     15355      8631
 1909     12200      5482     15777      9063      2351     12652      5944
16249      9545      2843     13154      6456     16771     10077      3385
13706      7018       332     10659      3977     14308      7630       954
11291      4619     14960      8292      1626     11973      5311     15662
 9004      2348     12705      6053     16414      9766      3120     13487
 6845       205     10578      3942     14319      7687      1057     11440
 4814     15201      8579      1959     12352      5736     16133      9521
 2911     13314      6708       104     10513      3913     14326      7730
 1136     11555      4965     15388      8802      2218     12647      6067
16500      9924      3350     13789      7219       651     11096      4532
14981      8421      1863     12318      5764     16223      9673      3125
13590      7046       504     10975      4437     14912      8378      1846
12327      5799     16284      9760      3238     13729      7211       695
11192      4680     15181      8673      2167     12674      6172     16683
10185      3689     14206      7714      1224     11747      5261     15788
 9306      2826     13359      6883       409     10948      4478     15021
 8555      2091     12640      6180     16733     10277      3823     14382
 7932      1484     12049      5605     16174      9734      3296     13871
 7437      1005     11586      5158     15743      9319      2897     13488
 7070       654     11251      4839     15440      9032      2626     13233
 6831       431     11044      4648     15265      8873      2483     13106
 6720       336     10965      4585     15218      8842      2468     13107
 6737       369     11014      4650     15299      8939      2581     13236
 6882       530     11191      4843     15508      9164      2822     13493
 7155       819     11496      5164     15845      9517      3191     13878
 7556      1236     11929      5613     16310      9998      3688     14391
 8085      1781     12490      6190     16903     10607      4313     15032
 8742      2454     13179      6895       613     11344      5066     15801
 9527      3255     13996      7728      1462     12209      5947     16698
10440      4184     14941      8689      2439     13202      6956       712
11481      5241     16014      9778      3544     14323      8093      1865
12650      6426       204     10995      4777     15572      9358      3146
13947      7739      1533     12340      6138     16949     10751      4555
15372      9180      2990     13813      7627      1443     12272      6092
16925     10749      4575     15414      9244      3076     13921      7757
 1595     12446      6288       132     10989      4837     15698      9550
 3404     14271      8129      1989     12862      6726       592     11471
 5341     16224     10098      3974     14863      8743      2625     13520
 7406      1294     12195      6087     16992     10888      4786     15697
 9599      3503     14420      8328      2238     13161      7075       991
11920      5840     16773     10697      4623     15562      9492      3424
14369      8305      2243     13194      7136      1080     12037      5985
16946     10898      4852     15819      9777      3737     14710      8674
 2640     13619      7589      1561     12546      6522       500     11491
```

Appendix CH4.5 Results of the message encryption/decryption program

```
 5473   16468   10454    4442   15443    9435    3429   14436
 8434    2434   13447    7451    1457   12476    6486     498
11523    5539   16568   10588    4610   15645    9671    3699
14740    8772    2806   13853    7891    1931   12984    7028
 1074   12133    6183     235   11300    5356   16425   10485
 4547   15622    9688    3756   14837    8909    2983   14070
 8148    2228   13321    7405    1491   12590    6680     772
11877    5973      71   11182    5284   16399   10505    4613
15734    9846    3960   15087    9205    3325   14458    8582
 2708   13847    7977    2109   13254    7390    1528   12679
 6821     965   12122    6270     420   11583    5737   16904
11062    5222   16395   10559    4725   15904   10074    4246
15431    9607    3785   14976    9158    3342   14539    8727
 2917   14120    8314    2510   13719    7919    2121   13336
 7542    1750   12971    7183    1397   12624    6842    1062
12295    6519     745   11984    6214     446   11691    5927
  165  11416    5658   16913   11159    5407   16668   10920
 5174   16441   10699    4959   16232   10496    4762   16041
10311    4583   15868   10144    4422   15713    9995    4279
15576    9864    4154   15457    9751    4047   15356    9656
 3958   15273    9579    3887   15208    9520    3834   15161
 9479    3799   15132    9456    3782   15121    9451    3783
15128    9464    3802   15153    9495    3839   15196    9544
 3894   15257    9611    3967   15336    9696    4058   15433
 9799    4167   15548    9920    4294   15681   10059    4439
15832   10216    4602   16001   10391    4783   16188   10584
 4982   16393   10795    5199   16616   11024    5434   16857
11271    5687     105   11536    5958     382   11819    6247
  677  12120    6554     990   12439    6879    1321   12776
 7222    1670   13131    7583    2037   13504    7962    2422
13895    8359    2825   14304    8774    3246   14731    9207
 3685   15176    9658    4142   15639   10127    4617   16120
10614    5110   16619   11119    5621     125   11642    6150
  660  12183    6697    1213   12742    7262    1784   13319
 7845    2373   13914    8446    2980   14527    9065    3605
15158    9702    4248   15807   10357    4909   16474   11030
 5588     148   11721    6285     851   12430    7000    1572
13157    7733    2311   13902    8484    3068   14665    9253
 3843   15446   10040    4636   16245   10845    5447      51
11668    6276     886   12509    7123    1739   13368    7988
 2610   14245    8871    3499   15140    9772    4406   16053
10691    5331   16984   11628    6274     922   12583    7235
 1889   13556    8214    2874   14547    9211    3877   15556
10226    4898   16583   11259    5937     617   12310    6994
 1680   13379    8069    2761   14466    9162    3860   15571
10273    4977   16694   11402    6112     824   12549    7265
 1983   13714    8436    3160   14897    9625    4355   16098
10832    5568     306   12057    6799    1543   13300    8048
 2798   14561    9315    4071   15840   10600    5362     126
11903    6671    1441   13224    7998    2774   14563    9343
 4125   15920   10706    5494     284   12087    6881    1677
13486    8286    3088   14903    9709    4517   16338   11150
 5964     780   12609    7429    2251   14086    8912    3740
15581   10413    5247      83   11932    6772    1614   13469
 8315    3163   15024    9876    4730   16597   11455    6315
 1177   13052    7918    2786   14667    9539    4413   16300
11178    6058     940   12835    7721    2609   14510    9402
```

Appendix CH4.5 Results of the message encryption/decryption program

```
13056    11792    10530     9270     8012     6756     5502     4250
3000     1752      506    16273    15031    13791    12553    11317
10083     8851     7621     6393     5167     3943     2721     1501
283     16078    14864    13652    12442    11234    10028     8824
7622     6422     5224     4028     2834     1642      452    16275
15089    13905    12723    11543    10365     9189     8015     6843
5673     4505     3339     2175     1013    16864    15706    14550
13396    12244    11094     9946     8800     7656     6514     5374
4236     3100     1966      834    16715    15587    14461    13337
12215    11095     9977     8861     7747     6635     5525     4417
14343    13495    12649    11805    10963    10123     9285     8449
15136    14416    13698    12982    12268    11556    10846    10138
9432     8728     8026     7326     6628     5932     5238     4546
3856     3168     2482     1798     1116      436    16769    16093
15419    14747    14077    13409    12743    12079    11417    10757
10099     9443     8789     8137     7487     6839     6193     5549
4907     4267     3629     2993     2359     1727     1097      469
16854    16230    15608    14988    14370    13754    13140    12528
11918    11310    10704    10100     9498     8898     8300     7704
7110     6518     5928     5340     4754     4170     3588     3008
2430     1854     1280      708      138    16581    16015    15451
14889    14329    13771    13215    12661    12109    11559    11011
10465     9921     9379     8839     8301     7765     7231     6699
6169     5641     5115     4591     4069     3549     3031     2515
2001     1489      979      471    16976    16472    15970    15470
14972    14476    13982    13490    13000    12512    12026    11542
11060    10580    10102     9626     9152     8680     8210     7742
7276     6812     6350     5890     5432     4976     4522     4070
3620     3172     2726     2282     1840     1400      962      526
92     16671    16241    15813    15387    14963    14541    14121
13703    13287    12873    12461    12051    11643    11237    10833
10431    10031     9633     9237     8843     8451     8061     7673
7287     6903     6521     6141     5763     5387     5013     4641
4271     3903     3537     3173     2811     2451     2093     1737
13002    12682    12364    12048    11734    11422    11112    10804
10498    10194     9892     9592     9294     8998     8704     8412
8122     7834     7548     7264     6982     6702     6424     6148
5874     5602     5332     5064     4798     4534     4272     4012
3754     3498     3244     2992     2742     2494     2248     2004
1762     1522     1284     1048      814      582      352      124
16909    16685    16463    16243    16025    15809    15595    15383
15173    14965    14759    14555    14353    14153    13955    13759
13565    13373    13183    12995    12809    12625    12443    12263
12085    11909    11735    11563    11393    11225    11059    10895
10733    10573    10415    10259    10105     9953     9803     9655
9509     9365     9223     9083     8945     8809     8675     8543
8413     8285     8159     8035     7913     7793     7675     7559
7445     7333     7223     7115     7009     6905     6803     6703
6605     6509     6415     6323     6233     6145     6059     5975
5893     5813     5735     5659     5585     5513     5443     5375
5309     5245     5183     5123     5065     5009     4955     4903
4853     4805     4759     4715     4673     4633     4595     4559
4525     4493     4463     4435     4409     4385     4363     4343
4325     4309     4295     4283     4273     4265     4259     4255
4253
```

Appendix CH4.5 Results of the message encryption/decryption program

Again the curves rational points

2 4835	4 16541	7 3799	8 2632	10 9691
12 10031	13 16051	15 12516	16 12088	18 13186
19 9968	21 8173	22 7965	24 12986	25 12332
27 12618	29 4603	30 11849	33 5600	37 3030
42 1188	43 14253	44 10474	45 15296	46 4691
47 9130	55 7007	56 143	58 13948	59 5048
60 4832	62 10629	63 841	66 8655	67 15587
72 13994	73 14948	74 16998	76 3339	77 10711
78 1959	80 10575	81 13609	83 9230	85 12119
88 13594	90 14153	92 10123	94 11914	95 9634
97 8804	99 9924	101 16674	107 5996	108 14893
109 682	111 5328	112 4525	115 3883	120 14043
121 5789	124 3938	125 11383	126 1629	127 11618
129 5936	132 11640	133 8916	134 10066	140 9048
141 15295	142 14068	143 14046	145 8922	146 11350
149 3948	150 4928	151 13385	152 13877	157 14503
159 6276	162 4225	163 11877	165 6676	166 7629
167 4881	169 3970	171 6117	173 3588	175 9700
176 2179	177 15572	180 10650	183 4108	184 16561
186 14819	188 342	189 1779	193 11651	194 11168
196 6187	199 10962	201 6837	204 6711	208 1398
209 3201	212 12396	214 5649	215 897	218 14209
226 606	234 2647	235 8469	237 10601	243 6211
245 434	246 8513	247 13022	248 8612	250 7835
252 3856	258 11130	260 15862	261 11060	263 4855
264 11142	266 4691	267 16992	270 14767	271 5308
273 9443	274 15473	275 1542	276 14734	278 8813
279 9984	281 6	282 11984	283 12680	285 8079
286 14814	288 2343	291 3938	293 4167	294 8892
295 14108	296 14681	299 16631	300 11615	301 10259
302 14050	305 16967	307 5906	308 15282	311 4188
313 4403	316 16190	318 3951	320 14439	321 2769
322 13689	323 5656	324 7756	325 8314	327 14469
328 8230	331 11943	333 5874	337 6479	338 15026
339 3355	343 11244	344 7401	345 10474	348 1470
349 7473	351 2793	354 9924	356 12616	357 2465
358 16956	359 16067	366 13609	367 13056	368 1322
370 12109	372 11155	373 4510	374 13589	375 9100
378 3588	379 9224	380 11350	383 9891	384 8281
385 10375	386 719	387 16922	388 2573	389 3442
391 575	392 10134	393 15627	394 237	395 6881
397 9709	398 2893	399 3001	401 1898	403 5857
406 7040	414 620	418 1723	420 6379	421 13300
422 7671	423 11296	424 14388	425 13156	426 3205
428 9579	429 7341	432 9967	433 11085	434 15579
439 4239	441 16304	442 1460	444 8253	445 3613
447 8034	453 12809	455 15961	457 10251	458 15219
461 15209	465 14601	466 834	467 14555	471 13506
472 14278	474 6485	475 4279	477 15472	480 3653
481 9884	483 6332	485 11127	487 12407	488 14676
490 7425	491 5947	492 8359	493 8159	496 14509
498 13000	499 12521	500 2292	501 4134	505 6623
508 5890	514 12227	515 1755	518 12315	525 14566
526 13960	528 12647	530 3788	532 8494	533 16604
534 4544	535 2413	539 16019	540 1299	543 8086

Appendix CH4.5 Results of the message encryption/decryption program

```
3554  9379    3555  1078    3556  7524    3558  6364    3559 15517
3561  1381    3563   587    3564  6233    3572  7599    3574  3691
3576 12628    3577  6391    3580 14030    3582  8229    3585 13181
3586 13627    3587 16365    3591   294    3593 12986    3596  1661
3597 16373    3599  7835    3601  8372    3602  4210    3603 12807
3605 11607    3606  9881    3609  5145    3612  2698    3614 13103
3615   144    3616   778    3617 16091    3618 14796    3619  7745
3621  8159    3622 16257    3623  9931    3624  7919    3628 10914
3629 11678    3631   843    3632  5907    3635 14775    3639  3005
3642  8680    3643  1062    3645  4753    3647 14216    3651  9362
3653  5905    3655 13490    3658  3238    3659 12237    3660   984
3661  4501    3662 14098    3663  3574    3665  9811    3666  5525
3667 14774    3668  3076    3670 16568    3671 14580    3672 13141
3674 15811    3677  6616    3678  7232    3679 16477    3681  4619
3686  4975    3688 15368    3696  3619    3697  2337    3698 11659
3699  9032    3700  8405    3701  9857    3702  8951    3705    23
3707  6090    3709 11768    3711 15029    3712 10746    3713 15094
3715 14655    3718   636    3719 13338    3721 16059    3722 16235
3723  2200    3726  1486    3727  9330    3728  2525    3729 15575
3730  1970    3731  8434    3733 10530    3739  5070    3741  4236
3742 14090    3744 16529    3745 14336    3746 15713    3750  2975
3755  6194    3756   536    3759 14130    3760  5605    3761  3688
3763 12438    3764  5467    3769  3331    3770  8325    3772   361
3774  3795    3776  9478    3778 14632    3779  7629    3782 11804
3783  2650    3785 13574    3789 13514    3794  3028    3795 15496
3797  2065    3798  2200    3799 15868    3800 12468    3807 13001
3809   864    3811  1195    3813  1674    3817 16442    3820 13040
3821  8877    3827 14196    3828  1848    3829  7553    3831   706
3837 15908    3839 16378    3841  9785    3846  1080    3847 14673
3848 10747    3854  4333    3855 11145    3858  2286    3860  1683
3861 10956    3863 10120    3864  2286    3865 16670    3866  3449
3867 10651    3868  1055    3870  4562    3872 16994    3876  9447
3878  1213    3881 13026    3882   987    3885  8604    3886 15402
3887 14628    3890 12771    3891  3546    3892 12406    3895  7762
3897  5432    3900 14426    3901  1221    3903  9059    3906 16025
3907  1553    3911  3032    3918  3929    3921  8731    3922  5978
3923  1021    3924  1009    3925 16418    3926  1600    3927 15266
3928  8098    3929  6799    3932 12736    3934 13396    3935  4320
3937  4613    3938 11190    3939  1626    3942 11017    3943 11562
3947 15968    3951  1507    3952  3939    3956  8199    3958 14673
3959  2021    3960  2086    3961 12993    3962  1154    3965  3671
3966   671    3967  7269    3969  4788    3970 13593    3971  9321
3972  9078    3974 13898    3976 14188    3977 14352    3978 14745
3980 10790    3982   737    3985 14296    3987  9263    3988  1562
3991  2843    3992  4472    3994  6802    3995  4787    3996  9413
3999 10859    4000 13153    4004 13324    4005  6811    4006 14369
4007  6446    4009 10609    4014 14743    4016 14635    4017  3920
4018  9214    4020  4100    4021 11317    4023  2342    4024  3453
4025 11698    4028 15092    4029  4515    4031 13284    4032  9388
4035 10949    4039    14    4040  9530    4042  5074    4046  2519
4050 11269    4051  4084    4054  4641    4055 15778    4057  4893
4059 14088    4060 11711    4061  3913    4063 13056    4064  8050
4065 14435    4066  5161    4070 12751    4072  6071    4073 14739
4074  5747    4075  7421    4078  8096    4079  4709    4080 13603
4085   671    4086  7252    4088  9848    4090  3613    4093  7874
4094  1591    4095  4506    4098  3233    4099 15339    4103  7755
4104 13603    4106 11918    4108  1266    4111  1344    4113  4869
```

Appendix CH4.5 Results of the message encryption/decryption program

```
4115  6154      4119  2260      4120  4080      4122  5605      4123 14223
4125 16625      4126   748      4128  3794      4129 14906      4130  6798
4134  4190      4136 15908      4137 12001      4140  9956      4144 11562
4145  1345      4146 16921      4148   320      4151  4517      4155   624
4157   113      4158  9099      4160  6356      4161 14404      4167  6357
4168 15427      4170 10606      4172  7177      4173   806      4174  1049
4175  3476      4176  9717      4179  8923      4181  9962      4190  1949
4191  2663      4192 10454      4197 12573      4198  8364      4202 11485
4206 12957      4207 12470      4208  7821      4214 15058      4215   592
4218  6117      4219 11674      4220  5874      4221 11251      4224 12737
4228  9877      4229 16874      4230   496      4231   254      4232 12577
4234   617      4236  5988      4237  2012      4238 13092      4240  8332
4241 15035      4242  7583      4244 11951      4247  1410      4248  3911
4249 13285      4251 12670      4252  8041      4256  1337      4258 16223
4259  1087      4261  9794      4266 14120      4269  5155      4271 14588
4272  6522      4274   654      4275  2025      4277   405      4283  4475
4284  9065      4287 12670      4295 16759      4296 16916      4299 10197
4302  3393      4306  3195      4307 13315      4308  1242      4310  9873
4311  5485      4312  2626      4315  7177      4316   905      4317  9294
4319  1999      4321  2361      4323  6151      4326  7725      4327  3566
4329  1905      4330 14963      4333 10188      4334 11196      4336  7332
4337  2927      4340  9291      4344  7495      4351 14431      4352  7565
4354  1760      4358  7988      4360 17001      4361  9425      4364 13692
4369  7837      4371  4479      4372 12779      4373 16936      4374 11819
4375  6047      4376  7542      4380  3896      4382  8086      4383  5454
4385 15716      4386 16784      4388 16696      4394  4977      4395  8573
4399  8813      4402 14743      4403  7791      4404  6347      4408 15599
4409  6566      4410 15173      4414 11997      4416  8910      4417 15355
4419  2361      4421   682      4422  1841      4423 13781      4428  4283
4429  6334      4430  4088      4431 15821      4432  6635      4435  3607
4439 14828      4440  6271      4441 14474      4442  3693      4443  7493
4444  1909      4445  4928      4447     9      4448 13955      4449   975
4450 13771      4451  1927      4453 12289      4454 14180      4460  4246
4461  5035      4463 13493      4466 15861      4467 13385      4468  6892
4476  9564      4477   486      4478 13663      4480  7220      4481 15770
4485  6796      4486  9091      4488  5466      4489 12327      4490 13368
4491  9111      4492 12073      4493  6084      4494   472      4498  4903
4502 11774      4503  7010      4505 11445      4506 13709      4508 12194
4510  9932      4511 15660      4512  6683      4514  4101      4515 13813
4516 12006      4517 12031      4518 13630      4519 10913      4524 11555
4526 14884      4528  1957      4530  3645      4532 13965      4533 10514
4535 11059      4536   152      4542  9414      4543  9909      4546  2285
4550  6803      4551  6794      4552  5470      4553  7430      4556   851
4558 16210      4559  9445      4560  3131      4562  8205      4563 14693
4567  7071      4568 14618      4573  2073      4579  2874      4580 16095
4582  4136      4584  1285      4587 10538      4588  7020      4591  2009
4594  7183      4595  8763      4599  8912      4600  7305      4602   747
4605   382      4607 13499      4609  1703      4611 10485      4613 13139
4616  7271      4618  4651      4619  9690      4622 12613      4624  4814
4625 11569      4627 11425      4628  8073      4629 16666      4631 15595
4632 10432      4633 14712      4634  3736      4635 11972      4636 11279
4640 15951      4646 15419      4648  3598      4649   975      4650  5859
4652 15534      4655 13991      4656 12926      4662  3466      4663  2384
4666 12169      4667  9931      4668  3156      4671 11430      4674 16870
4675  4324      4676  3579      4677 10277      4679 16758      4680  9180
4682 14960      4683  3426      4684 15951      4685 14770      4688  4986
```

Appendix CH4.5 Results of the message encryption/decryption program

```
7145 15282      7146 16608      7147  4389      7148  8352      7150 13387
7151  6190      7153  8271      7154 12325      7156  3237      7158  6042
7160  3101      7161  1137      7162 10434      7164 14706      7165  5340
7166  1404      7167 14693      7170 11327      7171  6803      7174 11178
7180 11378      7182  4272      7183 14252      7187  6478      7189   681
7190 11512      7192  4894      7193  8460      7194 13803      7197 10325
7200 12124      7202 10627      7205 14441      7206 15473      7208 14417
7210  4742      7212  4944      7214 10554      7215 11310      7216  6697
7218  5339      7220 12625      7221  5421      7222  7782      7223  3400
7224 12868      7225  6047      7228  3513      7230  1647      7231 15486
7232  1615      7234  1344      7236 15534      7237   820      7238 11942
7242  6153      7244  1701      7245 12004      7248  5244      7249  7270
7250 12345      7254  3389      7255  6124      7256 16414      7257 16569
7260 11095      7262  7747      7263  6478      7265 15372      7266  4838
7267 10396      7268 13942      7270 11613      7273 12864      7274 16062
7279 13079      7280  2176      7281 15158      7283   861      7284   719
7286 10711      7287  6369      7289  5476      7291 16003      7294  2864
7295 16809      7298 15212      7299  5983      7302  9368      7304  4961
7307 16952      7309 10765      7310 11805      7311 15497      7313  2025
7315  7821      7316   572      7317 15294      7320 16327      7322 11101
7323  3816      7324  7903      7326 11116      7330  4793      7333  6718
7337  1404      7338  8303      7341  8363      7342  6461      7344 13594
7347  6723      7348 13698      7349  1205      7350  5406      7351 16533
7352  1823      7354  5827      7356    45      7357  5687      7361 15160
7362 14935      7364 11103      7365  1322      7366  9444      7373  4343
7379 15602      7383 13942      7387  4374      7388  6324      7393 13005
7394 16344      7396  9240      7398  2627      7399 16900      7400 12373
7402   205      7403  3312      7404  2195      7405 16720      7408  2786
7410    45      7411 13222      7414 16270      7415  4509      7418 16858
7419 16289      7420   552      7421  9481      7423 16269      7424  4810
7426 12680      7427 12907      7428 16909      7430  6707      7431  5126
7433  1763      7436  7783      7437 14173      7439 15147      7443  9126
7445  4517      7447  4480      7448 12198      7452 10573      7457  8146
7458 10922      7464   749      7466 11833      7468  8693      7470 12495
7471 14466      7472  5978      7474 14346      7479  7451      7482  3309
7483 10353      7484  2366      7485  6109      7486  1213      7494 12001
7496  9797      7500 11659      7504  8835      7505  5759      7507 12807
7508  4308      7511 15847      7512  8827      7513  3608      7514 15223
7516 14321      7517  5734      7518  6733      7522 14266      7525  7374
7528  2660      7529  4740      7531  8332      7532 14757      7535   484
7537  9849      7539  9325      7540  6214      7541 10464      7542   282
7543 16578      7544 13161      7549 15547      7550  9884      7551 16054
7553 12798      7555  1820      7559 16237      7566 11778      7569 11917
7570 15897      7571 12579      7573 10438      7574 15155      7576  9443
7578  7078      7579 16998      7582  6187      7584  4305      7585   757
7587 13104      7590 13149      7592  8351      7593 15646      7595 12593
7605  6924      7606  8520      7607  8003      7608 15470      7611  8905
7613  4514      7618  4477      7620 12085      7622  4330      7623 11964
7625  4905      7628  5534      7629  5304      7630  4729      7633  1221
7639 13877      7640  2209      7641 11548      7646 13589      7647  2164
7648 16210      7649  6330      7651 15231      7658  9185      7659  6255
7660 11039      7661  8319      7662  5600      7663  9005      7665  2586
7667 15706      7668  3900      7669  4134      7670  4177      7671 14755
7672  3197      7673  3059      7678 14758      7680 11054      7681  8839
7682 16026      7685 14562      7687 15951      7689   864      7693 16929
7694  6522      7698 10416      7699  6199      7700 15923      7701   930
7703 16202      7704  8400      7705  5506      7708  1904      7709  9243
7717  9125      7718 13477      7721 12246      7724  7763      7725  3119
```

Appendix CH4.5 Results of the message encryption/decryption program

```
7726  2427    7728  8108    7729  6401    7730 10629    7733  1679
7734 16278    7737 12019    7738  1952    7739  5806    7740  2562
7742 15246    7745 10813    7746 12599    7747 13140    7748 10138
7749  7592    7750   206    7751 14854    7752 13600    7753  6271
7755 13242    7759 12993    7760 10936    7761   440    7763  5769
7764 10375    7765  1740    7767 15316    7769  6799    7771 11361
7775 12618    7777  5310    7778 10146    7780 11095    7781  2510
7782 10127    7784  5517    7785  9673    7786 16058    7788   126
7791 15443    7793  4625    7795 13281    7800 15756    7805  2348
7806 11511    7808 11138    7809  7284    7810 11291    7813  5669
7816 16220    7819  7541    7823  8760    7826  7782    7827  1197
7829 15180    7830 12735    7837 16564    7839  6519    7840 15695
7841 11108    7845  3306    7846 12033    7847   794    7848   159
7849  2948    7852  3419    7856 16131    7860 13248    7864  9245
7865 14404    7867  1821    7868  9157    7872  4641    7873  4179
7876  8281    7878  4525    7880   885    7883 14747    7886  7592
7889  7014    7892   709    7894  9349    7897  1548    7899  9668
7901  1019    7902 12509    7903  4309    7905  9425    7907 10991
7908  9809    7909  8649    7910 14690    7912  5087    7913   975
7915 10158    7916   205    7917  3461    7918  5002    7920   965
7923  8173    7925  5846    7926 11414    7930  5680    7932 12826
7933 16956    7934  9517    7935  4780    7936  6147    7938  3536
7941 14595    7944   792    7946  2871    7947  3270    7948  6720
7951 11044    7952  2632    7953  6819    7955  8982    7960 16350
7961 10806    7962   991    7963 13473    7964 12811    7969 14547
7970   935    7980 15333    7985  9200    7986  9660    7989 11096
7990 13154    7991  3779    7993 12962    8000  3986    8001 14523
8002  4423    8007  4148    8009 11862    8010 10344    8011 15235
8012  5013    8014 11609    8015  5100    8017 14550    8020 13531
8022  6311    8024  3779    8025 10164    8026  4642    8027  1280
8030  7295    8033   143    8035  7903    8036  7415    8038   686
8040 14950    8043  6190    8046  3590    8049  1833    8052  1593
8053 15710    8055  4827    8056 12254    8059  7942    8061   452
8067  4558    8068  1889    8070  5836    8074  8533    8078 13628
8081 13379    8086   650    8087  6679    8088 14510    8089   816
8091 13826    8092 11009    8096  4126    8097  3554    8098  2347
8103  8746    8105   148    8106  7405    8107 16054    8112  6942
8113 12679    8118  2624    8119 15063    8124  8967    8125  1121
8126 16918    8127  6639    8128 10510    8129 13954    8130  3207
8135  4028    8142  2269    8144  2292    8145  9057    8146  7942
8147  1382    8149  6196    8151 14215    8153  5534    8154  6074
8157  1154    8162  6561    8163  2789    8165  7764    8169  5400
8170  6378    8171 13727    8172 14752    8175  3662    8177  9061
8179 12013    8181  7918    8184  6372    8185  9241    8188 12232
8189  7657    8190 12872    8191 12217    8192  6078    8195 15386
8200 11415    8204 10965    8206  1548    8207  3139    8209 15686
8210 11046    8211  7967    8212  4220    8213 10487    8214 13629
8215 12549    8216  1762    8218  3759    8219  2824    8220 16496
8221  5604    8225 12804    8226  2993    8227  3791    8230 12961
8233  2785    8234 11500    8236 10389    8237 10889    8238 11444
8242 15824    8244  5846    8248  4764    8250  7711    8251  2392
8252  5075    8260 12794    8262  3457    8266 13948    8267  2976
8269 15796    8271  2361    8274  4510    8276  2705    8278  6192
8279  9923    8282 10121    8283  6426    8284  6032    8285 10225
8289   235    8291  4345    8292  7457    8298  9260    8299 16735
8300 15590    8301  4662    8303  4119    8305  8371    8308  6401
8311   528    8316  6741    8317 10149    8318 15433    8322  5222
```

Appendix CH4.5 Results of the message encryption/decryption program

8323 1535	8324 4931	8325 11768	8328 15532	8331 6391
8332 15537	8333 662	8334 2732	8338 4934	8342 14090
8344 16708	8345 15618	8346 16472	8347 5936	8349 14733
8350 2265	8351 13745	8352 5034	8354 4299	8356 8777
8357 16240	8358 16937	8360 14564	8361 1029	8364 4510
8366 10224	8367 2292	8370 16309	8371 5135	8372 4498
8374 14757	8375 3173	8377 12851	8380 203	8382 4125
8383 5361	8384 1744	8385 274	8388 9934	8392 6435
8397 14775	8398 1910	8399 5919	8400 3756	8401 13485
8402 4881	8403 12704	8406 5885	8408 1284	8410 7081
8411 6189	8414 10519	8416 4507	8421 12612	8424 10877
8426 9693	8427 311	8429 5196	8431 4	8432 8203
8433 15066	8440 11271	8441 669	8442 4881	8443 15831
8447 13429	8449 14195	8450 15746	8451 3884	8453 5864
8454 14887	8455 12260	8456 6535	8458 6280	8460 8796
8462 8581	8465 6199	8466 2421	8467 10605	8469 6842
8470 426	8472 16932	8473 12670	8475 8729	8478 4253
8479 16773	8480 11088	8485 9441	8489 5391	8490 4419
8491 1328	8492 4267	8493 15338	8495 8646	8496 15608
8500 4961	8501 10313	8502 10072	8504 16969	8505 1383
8507 12948	8508 8664	8509 7915	8511 15928	8516 269
8517 15470	8520 7667	8521 5435	8522 5865	8523 6162
8524 7177	8526 6076	8528 12690	8530 12389	8532 11658
8533 13393	8534 12554	8535 5936	8536 12952	8537 13470
8538 5028	8539 1127	8541 13620	8542 7991	8543 3238
8544 14324	8546 13497	8548 554	8551 877	8553 11803
8554 16149	8555 1549	8556 9539	8557 8992	8558 4471
8559 11225	8561 13640	8566 7596	8571 5673	8573 7785
8576 290	8579 15792	8580 4866	8583 5746	8587 6563
8588 2229	8592 16843	8593 1698	8595 14743	8598 15226
8600 1681	8602 12937	8603 416	8604 15967	8605 2214
8607 9435	8610 15043	8612 10574	8613 13447	8614 16364
8615 5385	8616 13783	8619 14082	8620 1920	8621 4068
8622 6069	8623 15152	8627 4560	8628 3191	8632 334
8634 7807	8636 12419	8637 8767	8641 16074	8642 12309
8644 12741	8645 2193	8646 14308	8649 9310	8650 14649
8652 14564	8655 11460	8656 2466	8661 1434	8663 8845
8665 5064	8667 15508	8670 15944	8674 9693	8675 14356
8676 6915	8678 2078	8679 5057	8680 7278	8685 4124
8687 13948	8688 6641	8689 13985	8692 16447	8697 9368
8698 11394	8699 8100	8700 5055	8701 15953	8702 13454
8703 7344	8704 9756	8706 932	8707 14880	8708 6375
8710 10730	8711 11378	8714 8377	8718 13825	8719 7665
8720 16956	8721 9931	8722 12692	8723 14204	8726 716
8727 16819	8730 5275	8732 8736	8734 3023	8736 15892
8738 15388	8739 10311	8740 14800	8741 11882	8745 4931
8746 9121	8749 10716	8751 8281	8754 12867	8756 3742
8758 11742	8761 11721	8762 16801	8763 13199	8764 3608
8766 185	8767 431	8768 1734	8770 14680	8772 12181
8774 8550	8777 5105	8781 11520	8792 12442	8793 657
8795 8121	8796 13866	8797 4488	8798 11051	8802 2296
8803 1840	8806 6781	8808 583	8811 9846	8812 15708
8813 2991	8815 11543	8817 11213	8818 836	8821 14353
8823 10536	8824 83	8825 14154	8827 13603	8829 15132
8832 5293	8834 6702	8836 13214	8837 2002	8838 2378
8841 4134	8842 5886	8843 8080	8848 708	8853 4192
8856 14518	8857 5066	8858 4051	8859 12855	8862 10375

Appendix CH4.5 Results of the message encryption/decryption program

```
8864 15061      8866 15056      8867 12741      8868  9027      8871  8876
8873  7733      8875  4098      8876  6365      8880  6422      8882 13456
8883  3721      8886 13178      8890  1070      8893 14280      8894  6039
8896  6509      8898  7737      8900  1414      8901 10046      8903  4952
8905  3031      8906  3793      8908 10667      8913  7115      8916  4636
8917  3389      8919  3309      8922   143      8924 11562      8926  9774
8936 15211      8938 14871      8942 13203      8944 14994      8948 16760
8950 12120      8951   319      8954  5735      8955 15176      8956 13302
8957  6389      8958  3728      8960   671      8969  9778      8974 10157
8977  5262      8978   816      8980  9884      8981  2755      8983  8895
8984 16939      8986  4959      8987 12669      8988  5330      8991 13589
8992  3125      8994 16723      8995  3224      8997 15636      8998  3975
9000  8000      9007 12051      9013  2732      9014  9386      9015 16566
9016  4251      9017  1620      9019  3659      9020 12669      9021  4525
9025  2630      9027  1194      9029  1698      9030  1757      9031 13614
9033 11904      9036 15645      9038 15908      9039 16560      9041  4180
9042  6564      9045 13976      9047  4101      9049  2422      9050  6322
9051  2632      9053  3115      9055  5315      9057 12376      9058    64
9059 16350      9061 14510      9063  2694      9065  4624      9067  8173
9070  7042      9071 12606      9073  5983      9076  5106      9078  4345
9080  5175      9081  5751      9082 14880      9084  2889      9086   431
9088   792      9091  5418      9092 11481      9093 16488      9094  6087
9095  1726      9098  4997      9099  5549      9100  9743      9103   506
9104 16311      9106 14479      9107  1737      9111 14316      9112 13707
9113 12546      9116 12255      9117 16425      9118 16084      9119  5574
9122  1105      9123 10313      9125  2297      9127  5302      9129  5605
9131  8423      9132  2263      9133 13154      9134  3278      9135 12794
9136  6283      9137  7459      9139  6428      9141 16447      9144 15167
9145  5660      9146 16632      9148  6936      9149  3532      9150  3838
9152 16569      9153  2725      9154 10062      9155 13114      9156 16448
9157  2066      9158 10318      9162  9443      9164  2496      9166 15896
9167  5441      9171  4095      9172 12275      9175 11533      9177 10795
9179  5493      9180 15099      9183 10214      9184 16538      9185 14947
9186 13301      9189 16760      9190 15532      9191 15125      9192   101
9193  6231      9195  7595      9196  7543      9199   836      9200 10929
9201 11982      9202  6834      9203 11660      9204  5045      9205  4714
9206 14673      9207 15521      9209 12618      9210  6353      9217  4052
9218  2171      9219 10629      9220 13877      9222   732      9228 16652
9232  2290      9233  6187      9235  9623      9239  4922      9240  7111
9243   678      9244  5163      9247  6196      9248  6634      9250  9920
9251  4077      9252   381      9254 13685      9255  3036      9260 13190
9261 11358      9262  2297      9264  7511      9270  8416      9271    56
9273 10349      9276  7410      9277  7333      9278  1322      9279  7206
9282  7344      9283 15263      9284   409      9285  4300      9287 12000
9288   126      9289  2286      9290 15390      9291  8939      9292 16883
9294 16888      9295  3549      9296   625      9297 10504      9299 13045
9300  6310      9302 12680      9304  2258      9305  1603      9306  9558
9309 10393      9311  8819      9312 13497      9315  9646      9316  5600
9317 14944      9320  9026      9322 12559      9324 10947      9325  5466
9326  5046      9328  6874      9330 13339      9331 14989      9333  1066
9338  9184      9339   915      9341   719      9342  6378      9345 14207
9350  7921      9353  5737      9357  3700      9358 16998      9359  4922
9361 16543      9362  8426      9364  3979      9365 13531      9368  7932
9369 15125      9371  6643      9372 15443      9373  9288      9375  9501
9376  9121      9378 13365      9383  4051      9384  4116      9385  6430
9387  4283      9388 12291      9389  6619      9390  4724      9398   289
9399  1984      9405 10542      9407    54      9408  5806      9409  8378
```

Appendix CH4.5 Results of the message encryption/decryption program

```
9413  8638    9415 10808    9419 10010    9421  5863    9423  1299
9424 16582    9425  7182    9429 10146    9431 16564    9435  2948
9438 11307    9441 13972    9442  9953    9444 10749    9445  1208
9450  7650    9451  5669    9452 11968    9455 14396    9456  5125
9458   844    9462  3154    9467 13829    9468  3794    9470 13014
9473 12812    9475 15972    9478  2208    9479  8767    9480  5911
9485 16874    9490  2200    9491 10025    9493   658    9497 16034
9499  5120    9501  4422    9506  7402    9507 14344    9509  3458
9511  5384    9513  3986    9514  9091    9515  4672    9516 13996
9517 11215    9518  2660    9519  5537    9520 13189    9521  7107
9522 12583    9529 11009    9531 15473    9532 14575    9534  3270
9535 16696    9536  3576    9539  5024    9540  1697    9541 14618
9542  9561    9543 14985    9545 15121    9551 15637    9552 14839
9553 10126    9555  2515    9558 15282    9559 13015    9560 10996
9566  2186    9568   333    9569  1228    9572 14940    9575 11464
9576  8621    9578 10786    9579 13594    9580  8656    9583  4846
9584 15208    9585  9243    9586   165    9587  6329    9589  6695
9591  9378    9593 12736    9595  6741    9599 15707    9600 13959
9601 10528    9602  7420    9608  8871    9609  1726    9610 14952
9612  2391    9613  9872    9614  8895    9616  4875    9617 15495
9618 13775    9619  2284    9625 12309    9626  8100    9627  5631
9628 11052    9629  4508    9630 16496    9631 14424    9632 11933
9633 14900    9635 12661    9637  8937    9639  1972    9640  7082
9644  9113    9646  1679    9647  6160    9648 10711    9654 14445
9655 10528    9657  4598    9658 14181    9659  5656    9662  1841
9664 13193    9666  4752    9668  4839    9671  6821    9673 11279
9678 12906    9679  5537    9680  3355    9681  1606    9682  4637
9683 12528    9684  5714    9686  8543    9689  4440    9690 12404
9692 10393    9693 10343    9695 15309    9698  6883    9700  3001
9702 13225    9703  3791    9704 11070    9706  7785    9707 15928
9708 14872    9710  9199    9712  3536    9713  7585    9714 13617
9718 16768    9719 15604    9721  9541    9722  4261    9723  2880
9724   529    9727  4177    9728   459    9729 10670    9731  6990
9732  3181    9733 15226    9735 12544    9738 10387    9739 12575
9740  1266    9741  2887    9742   183    9743 11052    9750  2306
9753  1271    9758 10014    9763  1583    9765  4379    9766  7280
9767  1098    9772  8033    9773  2946    9775  8170    9776 14154
9777 12538    9779   241    9780 10845    9782  3508    9786  2995
9788  3242    9789 14947    9793   350    9794  3962    9796  2231
9797  9597    9799  7652    9801 13469    9802 15637    9804  9541
9809  9990    9812  5658    9817  7335    9818  1826    9820 13873
9823 11463    9825 10670    9826  3795    9829  6467    9831 15590
9832  6904    9834  3286    9836  7923    9837 12525    9838  7286
9841  5282    9843 10066    9844  3040    9846  4426    9848  5009
9853 12304    9854  9575    9855 13107    9856  7330    9857  1825
9858   321    9859 12771    9861 12854    9863 16144    9864  7028
9868  7492    9869  2366    9872 12239    9874  5641    9876  5985
9877  1905    9881 11548    9882  4209    9883  3099    9884 16014
9885 15845    9887  1699    9888 11296    9889  8025    9890  4346
9896  4470    9897 12040    9903  4469    9905  6237    9906  6125
9907  8714    9908  9794    9910 11791    9911 13560    9913  9803
9914  2774    9916 15985    9918  2463    9919  7674    9925   289
9929  3146    9931 15716    9934  1328    9937 17008    9940  9646
9943 12177    9944 16686    9945  2317    9946 12984    9950 14264
9951   361    9955 10992    9956 11314    9958  7036    9960 14074
9965 14885    9966 15750    9969 12221    9975  7531    9976   506
9978   246    9980 13663    9981  2246    9982  1823    9984 10094
```

Appendix CH4.5 Results of the message encryption/decryption program

```
 9987  5194      9992   181      9995 10349      9996 10513      9997  9513
 9998  8328     10000 12868     10001  4977     10005 13024     10010  3434
13686  1522     13687  3412     13688  1743     13689 13097     13695 14632
13696  9310     13697  7945     13708  7729     13710  7479     13711 14206
13715  6479     13716  6640     13717  3609     13718  1629     13721 11994
13722 16311     13724  9332     13725  4225     13729  1047     13731 15239
13732  2826     13733 12577     13736  2880     13737  8929     13740  7435
13741 16026     13745  2434     13747  1019     13754  5766     13757  4858
13758  2721     13762  7487     13763  6535     13765 12780     13766 16748
13769  2199     13771 15324     13772  5885     13773  1081     13774 15556
13776   488     13778  4778     13779  6356     13781   789     13782  7232
13783 13800     13784  2755     13786 10117     13787   610     13788 16095
13791  6435     13792  6274     13795  1660     13796 11849     13799 15321
13800  9659     13801  8083     13802  8227     13804  3076     13806  1957
13809 13017     13816 16154     13819 11962     13820  9404     13823  8285
13824  5183     13827  7700     13829  8363     13832  5411     13836  9885
13837 15801     13839 14776     13842 10466     13844  2673     13846  2053
13847 16948     13849 15604     13853  3772     13855  2224     13856  8003
13857 11219     13860 13189     13862  8658     13863   928     13864  2839
13865 12973     13867  9160     13868 10708     13870 10342     13875  3426
13879  7468     13880 13876     13881  3993     13883  3461     13888  6443
13889  5516     13890  6701     13892 11290     13893 16916     13896 11735
13897  5552     13898 13525     13899  6051     13900  6110     13901  8972
13902  9811     13903 15800     13906   479     13907 16185     13909  9869
13913  1454     13917 11985     13920 13919     13924  4921     13925  1224
13927 16621     13929  2137     13930 16358     13933  7552     13936   606
13938 11140     13940  6792     13941 16871     13942 12988     13943  9136
13944  9580     13945  5316     13950 10875     13952  3681     13955  7657
13956 15958     13958  1652     13961  7967     13962  7610     13967  5651
13971  1999     13972  4602     13980  4039     13981 14971     13986  9851
13987  5201     13988 16555     13989  6594     13991 16635     13993 14628
13995   760     13998  5058     14002  3700     14004  6619     14006 16369
14007   424     14010  7710     14013 11219     14014  3756     14016  4363
14017  6903     14018  7106     14020  7435     14023  9496     14028   518
14030 12490     14031 10052     14033 12600     14034  9592     14035   950
14036  8161     14037  7416     14038  4827     14040  1711     14041 14341
14043  1657     14047 11673     14049  8445     14051 14480     14052  2724
14053   654     14054  1097     14057 10705     14058 13324     14059  1284
14061  8035     14062 15425     14066 10790     14067  4365     14069 14341
14070  6659     14071 12293     14072  7577     14075  5741     14076 12652
14080  4753     14082 14111     14083   488     14084  1205     14085  2990
14089  4067     14090 16463     14092 13159     14093 14192     14096  5370
14097  1005     14100 16794     14102  3331     14107  9048     14109 12495
14110 11651     14112  9032     14113  6021     14115 14622     14116 14659
14120  9774     14121  5114     14123 15461     14124 10138     14125  4650
14129  1415     14131  2207     14132 14606     14134 10833     14139  9945
14141  6739     14144 16059     14145  1885     14146  8158     14148  1693
14149  2562     14150  6602     14152   591     14153 13178     14154 11060
14156   709     14158 14786     14162  3269     14165  4419     14169   901
14170  8483     14172  7955     14173  6720     14175  8070     14176 12131
14181 11999     14182 10559     14184 14968     14188 12169     14189 16583
14190    73     14191  9165     14196 13374     14197 13904     14199 14284
14200 12376     14201 10746     14205  5155     14207 13698     14208  8416
14209 11450     14211  7007     14214 11705     14215 12582     14217  6550
14218  2424     14219 12315     14225 13992     14228  6517     14231 10949
14233  2106     14234 15234     14238 12639     14239  1087     14241  2525
14242  6635     14244  8146     14245 11489     14247  4356     14248  6148
14249 13533     14252 13845     14257 15626     14258  5763     14259 13409
```

Appendix CH4.5 Results of the message encryption/decryption program

```
14260 15262     14261  6796     14262  5577     14263  7334     14265 12722
14267  3224     14269 13772     14271 14662     14273 16265     14274 14070
14836 14915     14839 13011     14842  5183     14843  2075     14844  3723
14845  2424     14846 14302     14847  7358     14848  6025     14849  4975
16042 15325     16043 10299     16048  9521     16049 11536     16051  4033
16053  2525     16054  9892     16055  2643     16056  7638     16057 12577
16060  5406     16061  6158     16063 12410     16064 15870     16066  5148
16067 13287     16068 13315     16071  1837     16075 13487     16077  7100
16081 10577     16083  6356     16086  2175     16087  1572     16090  7267
16092  3791     16093  5907     16094  8276     16095 12365     16097 14182
16100 14451     16105 12114     16107  7174     16109 10746     16111  6741
16112 10527     16114  8842     16115 14884     16119   163     16120  5576
16122  5214     16126 12211     16134  7698     16135 14041     16136 11930
16138 11666     16139 13984     16140  6696     16141 10319     16142 14628
16143 10226     16149 12439     16150  4769     16151  3331     16153  9209
16154 13376     16156  2719     16157 16059     16162 10127     16163 12151
16164 13497     16165  1633     16166 15155     16167 11061     16169   431
16170 11784     16172 16496     16175 13681     16176  8655     16183 16942
16184 13144     16187 13491     16188 11173     16189 16447     16190  4304
16192  6143     16193  7706     16194 16383     16195  9998     16197  6644
16198 11008     16199  6232     16200  3112     16201  6355     16203  8048
16206 14527     16207  3203     16209 12724     16211  9032     16212  5882
16213 16743     16215  4549     16225  5482     16227  2653     16230  4195
16233 14880     16236  3648     16240 16499     16241  1927     16242 14621
16243   740     16246  1779     16249  6482     16253 13347     16254 11267
16256 14593     16257 16343     16262  8299     16263 11102     16265  5850
16268 14771     16269   624     16270  7573     16272  3790     16274  3547
16275  7101     16276  9912     16278 15709     16279  9514     16281  2549
16282  4705     16283 14398     16284  6185     16288  7970     16289 13296
16290  1881     16294    37     16295  8294     16296  6737     16297  4753
16298  5963     16302  3579     16304  4020     16305 13937     16307  7746
16308 16214     16312  2226     16313 16921     16316   383     16317   306
16319 12022     16320  1763     16322  9420     16323 14026     16330  6247
16332  6355     16333  1077     16334  9694     16336  2091     16337 16119
16338  9138     16341 11942     16343  3683     16347  1296     16352  2938
16354 15621     16356  5579     16358 15699     16360 12609     16361 11800
16363 11022     16366  4385     16368  1229     16369  9179     16370  3393
16372  3536     16374  5004     16376 12131     16377  4628     16378 16581
16379  4096     16382 12950     16384  8907     16387 10112     16388 12864
16391 13650     16392 16497     16393 14192     16398 10313     16399 11149
16400  1698     16401 11009     16403   121     16407  7939     16408 11834
16414  5702     16417  3629     16419 13616     16422  5224     16425 13683
16427 11002     16428 13657     16430  2954     16433 14520     16434 11008
16437 13916     16439  1230     16440 11278     16447  9773     16450  1007
16452  3669     16455  9811     16456  1230     16459 14311     16460  3588
16461 16062     16462 12012     16463  6591     16464  3929     16466 13797
16476 11722     16481  5681     16485 11350     16487  1021     16488 13788
16492  8030     16493  2616     16496  3657     16499   138     16500 11548
16503 16301     16504 15532     16505  7230     16506 15860     16508  2754
16509  3986     16510  6378     16511 12741     16513  8743     16515  1510
16517  8943     16521  8601     16523   842     16524 13509     16525  7622
16526 10968     16527  4384     16531 13369     16532 11616     16534  8701
16535 16397     16536 15633     16541  3270     16542 12373     16547  8677
16549 14632     16550  3076     16552  8002     16553  2809     16558  9924
16559 16501     16561  6919     16562  7232     16563 14275     16564 13609
16566 11059     16567  6724     16569  8008     16570  1911     16571 11775
16576  2530     16578  9630     16580  2033     16581 13714     16584 14417
16585 16418     16587  3279     16588 10552     16590 16308     16592  3248
```

Appendix CH4.5 Results of the message encryption/decryption program

```
16594  8636     16596  3938     16598  2971     16600   620     16601  3082
16602  6555     16603 13409     16604  3128     16609  9366     16610 11665
16612 15497     16613  8067     16614  7346     16616 16536     16617  9536
16620 10694     16622 10474     16624 10131     16625  4177     16626  6196
16627 12794     16628  2939     16629  6602     16631 13960     16637 13531
16642 12874     16643  1679     16644  4088     16645  8034     16646  2102
16648  6228     16651 14495     16652   471     16653  4345     16657 12809
16658  7957     16659  9111     16660  6243     16661  8636     16662  3681
16664 12005     16665 14252     16666  3442     16667 11306     16668 14240
16669  2366     16670  2963     16674  2724     16675  2732     16676  5659
16678  7430     16680 13020     16682  8868     16686 12642     16688  1823
16690 14926     16691  8586     16693  1165     16695 15074     16698  6827
16699  4691     16700  5810     16701  6610     16702  6112     16705  3003
16708  3120     16711  5344     16712  8807     16713 14949     16714 13506
16715 15305     16716 14655     16718   574     16719  7685     16723 16225
16726 16733     16727  9877     16728  9243     16730 14567     16731  5189
16733 10277     16735 13998     16738  2948     16740  2983     16744  1023
16746 14813     16747  8771     16748 16143     16750  1133     16751  9181
16752 10237     16753 13336     16754 16564     16758  5669     16760 13905
16763  5975     16764  7017     16767  6030     16768  2933     16769 13174
16770  4227     16772  1410     16773  2216     16775  3192     16776 14916
16778  8386     16779 15431     16784  7131     16785  3269     16787  6268
16791 15911     16792  3380     16794  6233     16795   573     16798 12868
16799 11644     16800  2866     16803  1816     16804 14888     16808  6447
16809 14658     16810  7429     16811  9945     16812  8728     16813 14786
16815 10146     16819  1061     16821  4859     16823 13649     16824 11134
16825 10982     16828 10430     16830 13760     16832  7348     16833 12454
16835  6516     16837 13506     16841  6093     16842  2737     16843 11678
16846 12822     16853 15696     16854  5970     16856  3349     16859 15443
16860  4651     16861  8763     16863    23     16865 13960     16868  1887
16869 15239     16871  4356     16873 14510     16874  1727     16875  5806
16876  3373     16879  4417     16880  1089     16881  2434     16884 14295
16885  2673     16893  1459     16897  8298     16898  9848     16899 13154
16901 16112     16902  3645     16904  8816     16905 12573     16908   371
16911  9937     16912 12809     16913  5161     16918 10323     16919 13887
16920    43     16922  9693     16930  8034     16933  8561     16935  3329
16936  7524     16938  5608     16939 10327     16943 14764     16945 14621
16946   126     16948  6523     16953  4931     16954  3916     16955   816
16958  5004     16960 10592     16961  1553     16967  3442     16976  2660
16979  9335     16980 11794     16982   772     16983 11642     16985  2714
16987 16091     16989 12166     16990   792     16996  3300     17001  7136
17003 16350     17004 14102     17005 15111     17008   620     17010 14564

                    ORDER OF FIELD IS 21223
```

```
   1    1507         1507 16417
   2    2704         2704   393
   3    2625         2628  2950
   4    1201         1204 15324
   5     527          528 12647
   6    2006         2006  9629
   7    1827         1828  1936
   8    2205         2205 14215
   9    2715         2715 13753
```

Appendix CH4.5 Results of the message encryption/decryption program

```
  10    1619         1619   915
  11    2113         2113 10358
  12    2710         2710   401
  13    1524         1525  4099
  14    2706         2708 11651
  15    2314         2314 12327
  16    1815         1816  6343
  17    2702         2702 11279
  18     311          311  4188
  19    2109         2109 15967
  20    2717         2718  3716
  21     605          610  3309
  22      20           21  8173
CODED INPUT MESSAGE    THE CURVE EMBEDDMENT
   1    1507        1507 16417
   2    2704        2704   393
   3    2625        2628  2950
   4    1201        1204 15324
   5     527         528 12647
   6    2006        2006  9629
   7    1827        1828  1936

   8    2205        2205 14215
   9    2715        2715 13753
  10    1619        1619   915
  11    2113        2113 10358
  12    2710        2710   401
  13    1524        1525  4099
  14    2706        2708 11651
  15    2314        2314 12327
  16    1815        1816  6343
  17    2702        2702 11279
  18     311         311  4188
  19    2109        2109 15967
  20    2717        2718  3716
  21     605         610  3309
  22      20          21  8173
```

```
SET OF SPECIMEN MULTIPLIERS
 1289  5       873  4

FOR THE GENERATOR POINT = 8360  14564

 THE RESULTS OF THE MULTIPLY-BY-7 POINT MULTIPLICATION OPERATION ARE AS
FOLLOWS:-

 1 -3 -2  2  1

 8360  14564  5865  16280  4916  3405  13532  13881  978  301
     END OF MULTIPLIER MR= 1289 AND BASE POINT = 8360    14564

AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS  978  301

   THEREFORE MICHAEL'S PUBLIC KEY IS  978  301
```

Appendix CH4.5 Results of the message encryption/decryption program

```
SET OF SPECIMEN MULTIPLIERS
 1289  5       873  4

FOR THE GENERATOR POINT = 8360  14564

 THE RESULTS OF THE MULTIPLY-BY-7 POINT MULTIPLICATION OPERATION ARE AS
FOLLOWS:-

 1 -3 -2  2  1

 8360  14564  5865  16280  4916  3405  13532  13881  978  301
     END OF MULTIPLIER MR= 1289 AND BASE POINT = 8360    14564

AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS  978  301
FOR THE GENERATOR POINT = 8360  14564

 THE RESULTS OF THE MULTIPLY-BY-7 POINT MULTIPLICATION OPERATION ARE AS
FOLLOWS:-

 3 -3 -1 -2

 5187  16942  3955  14672  16809  13832  16530  5539
     END OF MULTIPLIER MR= 873 AND BASE POINT = 8360   -14564

AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS  16530  5539

  THEREFORE LILIAN'S PUBLIC KEY IS 16530  5539

_____________________________________________________________________________
FOR THE GENERATOR POINT = 16530  5539

 THE RESULTS OF THE MULTIPLY-BY-7 POINT MULTIPLICATION OPERATION ARE AS
FOLLOWS:-

 1 -3 -2  2  1

 16530  5539  3640  11137  15162  8977  174  4371  11027  8726
     END OF MULTIPLIER MR= 1289 AND BASE POINT = 16530    5539

AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS  11027  8726

    LILIAN'S TRANSFOPRMED PUBLIC KEY IS 11027  8726

_____________________________________________________________________________
FOR THE GENERATOR POINT = 978  301


THE RESULTS OF THE MULTIPLY-BY-7 POINT MULTIPLICATION OPERATION ARE AS
FOLLOWS:-

 1 -3 -2  2  1

 978  301  11556  6722  8533  4987  8903  15679  3516  14281
     END OF MULTIPLIER MR= 1289 AND BASE POINT = 978    301

AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 1289 IS  3516  14281
```

Appendix CH4.5 Results of the message encryption/decryption program

```
FOR THE GENERATOR POINT = 978  301

 THE RESULTS OF THE MULTIPLY-BY-7 POINT MULTIPLICATION OPERATION ARE AS
FOLLOWS:-

 3 -3 -1 -2

 11122  16627  7357  2554  9924  9992  11027  8726
     END OF MULTIPLIER MR= 873 AND BASE POINT = 978   -301

AND FROM REFERENCE PROGRAM OPERATION [nP] FOR 873 IS  11027  8726

   MICHAEL'S TRANSFORMED PUBLIC KEY IS  11027  8726

_____________________________________________________________________________
  1    7842         2346
  2   11750           16
  3    1608        12565
  4   12555         9185
  5    6926        12655
  6   13202         9812
  7    8918         7380
  8    1253         4343
  9   13617         9536
 10   14606         2515
 11     954         8446
 12    9041        12792
 13   12250          259
 14   10761           42
 15    8380        10376
 16    3664        11782
 17   11786        16088
 18    4520         7286
 19   12668        11909
 20    7545         8423
 21    1926         8348
 22   13933         2298
     BELOW IS YOUR SCRAMBLED MESSAGE

 BLK/N0    ENCODED MESSAGE       EMBEDDED MESSAGE        ENCRYPTED  MESSAGE

  1           1507                  1507 16417             7842  2346
  2           2704                  2704   393            11750    16
  3           2625                  2628  2950             1608 12565
  4           1201                  1204 15324            12555  9185
  5            527                   528 12647             6926 12655
   Appendix CH4.5                                         QB 4.5 CODE
_____________________________________________________________________________

  6           2006                  2006  9629            13202  9812
  7           1827                  1828  1936             8918  7380
  8           2205                  2205 14215             1253  4343
  9           2715                  2715 13753            13617  9536
 10           1619                  1619   915            14606  2515
 11           2113                  2113 10358              954  8446
 12           2710                  2710   401             9041 12792
 13           1524                  1525  4099            12250   259
```

Appendix CH4.5 Results of the message encryption/decryption program

```
  14          2706                2708 11651            10761    42
  15          2314                2314 12327             8380 10376
  16          1815                1816  6343             3664 11782
  17          2702                2702 11279            11786 16088
  18           311                 311  4188             4520  7286
  19          2109                2109 15967            12668 11909
  20          2717                2718  3716             7545  8423
  21           605                 610  3309             1926  8348
  22            20                  21  8173            13933  2298
_____________________END OF ENCRYPTION_____________________

         Cross checking the points of the cipher

 2  4835         4 16541         7  3799         8  2632        10  9691
12 10031        13 16051        15 12516        16 12088        18 13186
19  9968        21  8173        22  7965        24 12986        25 12332
27 12618        29  4603        30 11849        33  5600        37  3030
42  1188        43 14253        44 10474        45 15296        46  4691
47  9130        55  7007        56   143        58 13948        59  5048
60  4832        62 10629        63   841        66  8655        67 15587
72 13994        73 14948        74 16998        76  3339        77 10711
78  1959        80 10575        81 13609        83  9230        85 12119
88 13594        90 14153        92 10123        94 11914        95  9634
97  8804        99  9924       101 16674       107  5996       108 14893
109   682       111  5328       112  4525       115  3883       120 14043
121  5789       124  3938       125 11383       126  1629       127 11618
129  5936       132 11640       133  8916       134 10066       140  9048
141 15295       142 14068       143 14046       145  8922       146 11350
149  3948       150  4928       151 13385       152 13877       157 14503
159  6276       162  4225       163 11877       165  6676       166  7629
167  4881       169  3970       171  6117       173  3588       175  9700
176  2179       177 15572       180 10650       183  4108       184 16561
186 14819       188   342       189  1779       193 11651       194 11168
196  6187       199 10962       201  6837       204  6711       208  1398
209  3201       212 12396       214  5649       215   897       218 14209
226   606       234  2647       235  8469       237 10601       243  6211
245   434       246  8513       247 13022       248  8612       250  7835
252  3856       258 11130       260 15862       261 11060       263  4855
264 11142       266  4691       267 16992       270 14767       271  5308
273  9443       274 15473       275  1542       276 14734       278  8813
279  9984       281     6       282 11984       283 12680       285  8079
286 14814       288  2343       291  3938       293  4167       294  8892
295 14108       296 14681       299 16631       300 11615       301 10259
302 14050       305 16967       307  5906       308 15282       311  4188
313  4403       316 16190       318  3951       320 14439       321  2769
322 13689       323  5656       324  7756       325  8314       327 14469
328  8230       331 11943       333  5874       337  6479       338 15026
339  3355       343 11244       344  7401       345 10474       348  1470
349  7473       351  2793       354  9924       356 12616       357  2465
358 16956       359 16067       366 13609       367 13056       368  1322
370 12109       372 11155       373  4510       374 13589       375  9100
378  3588       379  9224       380 11350       383  9891       384  8281
385 10375       386   719       387 16922       388  2573       389  3442
391   575       392 10134       393 15627       394   237       395  6881
397  9709       398  2893       399  3001       401  1898       403  5857
406  7040       414   620       418  1723       420  6379       421 13300
422  7671       423 11296       424 14388       425 13156       426  3205
428  9579       429  7341       432  9967       433 11085       434 15579
```

Appendix CH4.5 Results of the message encryption/decryption program

```
439  4239      441 16304      442  1460      444  8253      445  3613
447  8034      453 12809      455 15961      457 10251      458 15219
461 15209      465 14601      466   834      467 14555      471 13506
472 14278      474  6485      475  4279      477 15472      480  3653
481  9884      483  6332      485 11127      487 12407      488 14676
490  7425      491  5947      492  8359      493  8159      496 14509
498 13000      499 12521      500  2292      501  4134      505  6623
508  5890      514 12227      515  1755      518 12315      525 14566
526 13960      528 12647      530  3788      532  8494      533 16604
534  4544      535  2413      539 16019      540  1299      543  8086
545  5988      546 15543      548  7955      555  8058      556  2289
558 12411      559  6531      560  5731      563  6420      566  6691
567 10913      569 15423      571 12946      572  8058      574 13876
576  1779      579  8631      582 16382      584 16130      585  2083
590  6699      591 14924      592  5157      593 11489      596  8684
597  4489      602 16569      603  1045      610  3309      615 12835
618 13048      619 11530      620 14954      621  8630      622 16154
623 10387      629  4719      631 12227      632 15291      633  1431
635 13789      639  5983      640  3028      641  4501      643 10231
644  1796      647 10579      652 12027      654 10458      656 14330
658  6102      659 13807      661  3733      662 11433      663  8118
667  8405      670  2946      671  8385      672 11413      674  2477
676  7337      679  8574      680  7145      681  8483      682  9398
683 16547      687  6759      689   729      690  5004      691 10786
692  1641      695  8495      696 14468      697  5883      699  2734
700 11328      701  6882      704  2770      707  4136      708  6171
709  7269      710 16056      714 10028      715   255      720  2258
721 13298      722   790      723  2935      726  6742      727 15657
729 14097      730 14184      732 10893      733  6450      734  3608
736 11932      737  4296      740  9165      744   591      745 11476
747  4680      752 15850      754 12382      757 16619      758  2708
759  3180      760  6242      761 16672      762 16933      763  7056
764  2037      767  8636      769  8655      770  4619      771 10077
776  8477      777  7420      779  4776      781 15698      785  3647
788 13823      789  1603      790  1334      791  3617      792  5486
796 14376      797 16392      798  9529      799 13951      801 11537
803  3246      804  4660      806  7942      808  9237      810  8574
814  6424      819  3298      821  6964      825 13800      828 13771
829 14536      831  5453      832  1570      834 12039      835 14621
836 11694      837  4324      838 14167      840  3389      842  5846
843  7756      847  6199      851  1201      854  6994      855 10410
858 16272      860 14979      864  3943      866  9445      868  2251
869  6208      872  6212      874  6652      875 12125      878 13906
879 15743      880  6549      882  1526      884  1634      886 15470
889  5188      890 16038      892  1286      895   520      898  5230
899  6776      902  5563      903  7416      905  3327      906 14369
907  5409      908  1548      912 16599      913   825      917  5930
918  6832      921  2408      928 11955      929  7577      931  4441
932  7293      934 14158      936 14868      937  8230      938  3896
939  3879      941  8164      944 12938      948 14103      949 14535
952  6186      953  7169      954 11503  954 11503  8446 A HIT
955  2928      956  5363      959 16476      960  7790      962   546
963  9544      966  4236      967 12957      968 11490      970  9992
971 10193      974  6401      980 12028      981  4965      983 10916
985 16537      986 15688      987 11939      991  6604      993 16522
996  3779      997  8823      998  5812      999 12776     1000  8320
1005 14897     1008  5061     1009 15553     1011 10058     1012  9368
1014  1003     1018 16529     1020  3764     1023  8548     1025  3047
```

Appendix CH4.5 Results of the message encryption/decryption program

```
1026  7048      1029 14326      1030  9550      1037  9988      1040 12230
1041 15427      1042 14776      1043 10139      1044 13548      1045  7254
1046  4039      1047  5802      1048 10118      1052 10881      1054 12330
1056 11554      1057 12519      1058  9696      1059 16883      1060  9078
1062 15218      1064  9397      1068 10592      1069 12988      1074  3920
1077  8118      1080  6792      1083  8227      1088  1306      1090 10537
1091  9414      1093 13125      1094 12445      1097 13740      1098 12162
1104  3299      1105 14757      1106  6660      1107 12251      1109  4587
1114  2704      1115  9956      1119 10271      1120 11378      1121  6691
1124 11804      1125 10697      1127  1230      1131  3793      1136  8630
1138 16645      1139  5025      1144  8573      1146   724      1148  3740
1151  7125      1153 10878      1154 13902      1155   110      1158  1290
1159  5364      1162  6467      1168  3792      1169  9397      1172  9579
1174  2490      1176 16785      1179  2223      1181   598      1182   319
1186  4300      1187  2182      1188  9208      1189  7874      1190  5001
1191 15651      1193 11556      1194 10609      1195 15269      1199  6634
1200   336      1204 15324      1206  9317      1207  4961      1209 10139
1211 16616      1214  9163      1215  5802      1220  6093      1224 16633
1227   544      1228  8912      1229 13233      1230  5534      1232 12521
1237  6058      1238  9833      1240  2260      1242 13516      1245  4527
1246 13822      1249  3939      1250  6028      1252  9047      1257  4783
1258 16051      1259  6948      1261 10324      1264  2037      1266  2107
1275  9893      1276  6045      1277  7710      1278  7561      1282 13230
1284  5357      1285 10830      1286 14634      1288  8445      1303 12238
1305 16771      1308  8022      1309 12221      1310  9642      1315   332
1321  4239      1322   724      1324  4545      1327 11751      1329  3802
1330 10691      1332  4888      1334  2440      1336 14718      1338 15264
1339   708      1340  1408      1341 10775      1342  6381      1343  3581
1346 13914      1347  1660      1349 15762      1353 16054      1354  9192
1356  8658      1361  1009      1363  6282      1364  5842      1366  6067
1370 15959      1372  9629      1373  7137      1375 16579      1376  7592
1381 16139      1382  7636      1383   890      1384  4795      1385 14204
1387  7031      1389 14367      1390 11008      1391 12737      1394  3530
1395  1873      1397 14184      1401  6039      1403  9136      1404  9696
1409  4181      1410  7280      1411  8073      1412  1274      1414  7667
1418  9258      1423  2866      1424 12546      1427 12262      1432  5651
1434 14191      1435 10099      1437  4052      1438  6610      1441 12846
1443  2551      1444  5167      1445  5057      1451  9743      1452  5083
1453 12433      1457  2224      1459 16658      1461   693      1463  8539
1468 10049      1469  8361      1473 15335      1475  7350      1476 16791
1479  4053      1480  1611      1481 11534      1483 11416      1484  6144
1485  9575      1489  6355      1490 14972      1491   387      1494 13071
1495  5549      1498  1641      1507 16417      1509  4862      1510 13380
1514 13193      1515 10031      1517  2462      1519  8714      1525  4099
1529  8000      1530 11752      1532 15911      1537  8718      1540   513
1543  2573      1544 15455      1547 16042      1548  3447      1550 12304
1553  3529      1554 11978      1555  2671      1559  1110      1561  1854
1564  1522      1566  1840      1568  7273      1570  4279      1571  4354
572 12544      1575   676      1580  9625      1582  4420      1585  6530
1591 13658      1592 14474      1594 12170      1597 15286      1598  3962
1599  1042      1602  8919      1603  4294      1608  4734  1608  4734
12565 A HIT
1611  4055      1612  5651      1613 14778      1614  6742      1615  3123
1616  2229      1617  6470      1618 15451      1619   915      1620  3964
1621  8796      1622  4747      1623 11259      1624 11948      1626 15888
1631  4947      1636  5932      1639  5158      1643  6268      1644  5046
1645  5737      1647 15380      1649 12274      1650  1335      1652  7903
1654  3593      1656   525      1657  2874      1658 13973      1661 11076
```

Appendix CH4.5 Results of the message encryption/decryption program

```
1663 15140      1664  6761      1665  9136      1667  1949      1668  4863
1669  5610      1670 15863      1674  8445      1675  2395      1683 10044
1684   269      1685  2832      1688 13254      1690 14132      1691 12011
1692  4094      1693   205      1696 11363      1697  3731      1699  2224
1701  5841      1703 12787      1705  5391      1707  9785      1709  6343
1714  8984      1716  7737      1717  9224      1722  3245      1723 13646
1724  7710      1725  7665      1728 10716      1730 15139      1732  7891
1735   368      1739 16280      1741  1222      1743 16396      1744 12583
1745 10283      1746  2161      1748  2951      1750 14900      1751  7923
1754   409      1758 15639      1759  2443      1761  1522      1763  1127
1766  5726      1767  8071      1770  5528      1771  9793      1772 16541
1774  7299      1775  4450      1777 13736      1778  1998      1779  8868
1780  4488      1781  9162      1784 13949      1785  1048      1786  8744
1788  7459      1789  8512      1793  8658      1796 11691      1797 11800
1798 16641      1800  7791      1802  6954      1808  7040      1809 15918
1810  5914      1812   692      1816  6343      1817  6190      1820 15117
1823 11037      1824 13868      1825  5105      1826  2417      1828  1936
1830  4677      1831 16363      1836 16474      1838  8612      1840 16668
1842  1513      1843  3839      1846   551      1848 10204      1851  7141
1853 15707      1855  7182      1858  5649      1862 16272      1865  8851
1867  6349      1868 10046      1869  1660      1870 15647      1872 10530
1873  3286      1875 10563      1877  1162      1878 16503      1879    33
1882 10736      1883 15152      1887   900      1888 12829      1892 11504
1893  1017      1895 11757      1896  3465      1899     6      1900  3785
1901 14937      1903 12337      1904 15926      1905  8871      1908  9813
1910 15698      1911  9530      1912  5013      1915 11555      1918  1600
1920 15730      1923  3172      1925 15040      1927  2155      1928  1946
1930  8579      1933 14296      1936  4251      1941 12017      1942  8579
1943  1731      1948  7511      1950  2860      1951  8489      1952 14906
1955 15880      1958 16310      1959  6031      1962 16204      1963  7782
1964  3091      1965   452      1969 10967      1970 13150      1971 11095
1972  9562      1973 12709      1974 11061      1976 12126      1980 15942
1982  3884      1983 14901      1985  4039      1987  3374      1989 12471
1991  6792      1993  1789      1994  1606      1995   324      1997  2395
2000 12988      2002  3856      2003  7548      2004  2378      2005  3120
2006  9629      2008 10345      2009  5194      2010  7577      2011 14756
2012 16373      2013 13119      2014  1873      2017 12525      2022 11807
2023 10149      2024 13538      2025 11658      2027 10736      2030 14872
2032  7644      2034  4713      2035   541      2036 15324      2041   981
2042  3128      2043  3082      2045 15633      2046  3662      2048  4280
2050  2712      2051 10237      2052 10151      2053  8223      2054 14355
2055  4187      2057  8292      2063 11315      2064 15043      2065 14253
2068  9809      2071  7416      2073 15233      2076 10926      2077   873
2080  9165      2082  3321      2086  5975      2088 12980      2091  8393
2095   841      2099  3407      2104  4846      2105  8907      2109 15967
2110  4325      2113 10358      2115   591      2121  1343      2122 13557
2123  7193      2124  9321      2125 14710      2126  8227      2127  2549
2128 11132      2129 16983      2130 14776      2131 13570      2132 14059
2134  3006      2135  1707      2143 10151      2145 15584      2148 14839
2150  4982      2152   909      2154  1013      2155  7908      2156 10023
2158  6523      2159 13657      2160  8483      2161 14458      2163  3324
2164 13813      2165  2950      2167 13502      2168  9063      2169 16380
5112  7740      5116 10451      5118  8555      5119    38      5120 10742
5122 14642      5123 15534      5124 12769      5126 10118      5132 11553
5137  7402      5140 16820      5141  4518      5143  2645      5145  8804
5146  6129      5147 15495      5148 11900      5154 10708      5157 12525
5165  8151      5166 11149      5167  2214      5168  1447      5170  4950
5172 14082      5174  5631      5177 14734      5180  6775      5183  5222
```

Appendix CH4.5 Results of the message encryption/decryption program

5184 2208	5191 8320	5197 9548	5199 16959	5203 10900
5205 10471	5206 8755	5207 15813	5208 1549	5210 14699
5219 4537	5221 8274	5223 16604	5224 369	5226 16555
5228 5472	5230 858	5231 12745	5235 16747	5240 8332
5246 6643	5247 2708	5248 15840	5251 12255	5252 2374
5257 12689	5258 9212	5260 14160	5263 11572	5264 5341
5265 1395	5273 14872	5276 3744	5278 16942	5279 2422
5280 15384	5281 14693	5283 10362	5284 7246	5285 14119
5286 658	5287 2175	5289 4020	5290 6803	5291 12993
5295 2933	5297 6955	5299 12674	5300 8678	5301 3227
5302 14392	5304 3286	5305 15349	5306 3644	5307 8117
5309 7391	5311 14839	5313 6799	5316 5176	5317 10268
5319 5062	5322 13776	5324 4846	5325 15686	5326 16946
5327 12755	5329 13240	5333 2855	5335 1874	5336 1606
5338 605	5341 2141	5342 3345	5344 14356	5352 10254
5354 15899	5355 9998	5356 5183	5357 16139	5358 15425
5359 8015	5363 2855	5366 3702	5367 4485	5371 16545
5380 12001	5384 14153	5389 10697	5390 15699	5392 14209
5393 8764	5394 6521	5398 2530	5399 6074	5402 11982
5405 10847	5407 480	5410 7627	5411 6047	5414 5606
5415 4517	5420 12946	5423 2025	5424 7923	5427 7987
5428 7101	5429 6559	5431 15452	5433 16281	5435 1057
5438 10037	5439 16723	5447 12874	5449 10929	5450 10634
5451 3088	5452 1756	5453 13998	5456 416	5457 7383
5460 14754	5462 15167	5463 2565	5464 6103	5466 2897
5468 6213	5469 8171	5470 4862	5475 13938	5476 6976
5477 1221	5480 4198	5482 3325	5483 156	5484 14111
5486 9584	5488 7821	5490 7594	5491 12629	5492 12103
5493 7551	5496 1401	5497 12867	5498 8871	5505 796
5506 5842	5507 950	5508 6766	5510 5263	5511 1334
5513 864	5514 3647	5515 11439	5519 9914	5521 10506
5522 8377	5526 15177	5527 13525	5529 10186	5532 1888
5535 5370	5538 13825	5539 7926	5540 16190	5542 7405
5545 2311	5549 4798	5554 13301	5555 9006	5556 789
5557 2041	5558 13078	5559 15707	5560 5295	5562 2648
5565 11224	5566 5828	5568 15538	5570 1994	5574 11527
5575 9233	5580 16545	5581 58	5584 1954	5585 8714
5587 1034	5588 5626	5592 16820	5593 13114	5594 37
5598 13955	5601 3783	5603 1760	5604 6963	5605 4366
5607 15772	5608 12304	5609 8480	5611 4714	5612 7562
5615 7888	5617 5978	5618 3683	5619 3962	5620 1542
5622 1353	5624 6955	5625 9077	5626 6108	5627 15399
5628 14900	5631 6068	5632 3085	5634 9152	5635 12629
5636 15088	5637 14954	5638 4126	5640 15765	5642 9546
5644 5435	5645 13822	5646 11657	5647 1213	5649 12419
5650 7487	5652 12110	5653 8038	5659 8794	5663 7856
5665 5882	5666 1344	5668 3469	5670 2137	5672 9575
5673 7223	5676 14672	5683 5791	5684 610	5685 15807
5686 4087	5688 5514	5692 13991	5697 6766	5698 11046
5699 14049	5700 9984	5701 10226	5702 15587	5703 14886
5704 12544	5705 5946	5706 152	5708 11444	5710 11902
6343 13079	6350 4788	6351 752	6352 2016	6356 16276
6357 4052	6363 8769	6364 13995	6366 11181	6367 4575
6371 1443	6375 16442	6376 11466	6377 14480	6379 15333
6382 4330	6383 6963	6387 8513	6389 9878	6393 1863
6394 7869	6395 11383	6396 10164	6397 7737	6400 3668
6401 15664	6407 12500	6409 5105	6411 12047	6415 10465
6417 5549	6419 16312	6420 12803	6425 8428	6429 14143

Appendix CH4.5 Results of the message encryption/decryption program

```
6430 12144      6432   897      6433 10413      6434  4488      6435 12720
6441 12797      6442 11642      6444 12217      6445  4843      6446 14509
6447  5828      6449 10621      6451 11559      6452  9514      6453 12654
6454 11658      6459  1710      6460  9743      6468 10399      6469  9299
6473  5148      6474 12546      6477   575      6479 14130      6480  2606
6482  8000      6484 15921      6485 10898      6486 12781      6489   553
6492 16536      6496  2779      6497 13590      6498  8916      6502  2569
6503 16087      6504 13864      6505 15152      6506 11251      6507 16201
6508  3910      6510  5810      6511  9962      6512  5692      6513 11710
6516  9466      6518 12278      6519  6892      6522 13135      6523  9873
6526  1484      6528  2195      6531  8843      6533  5041      6534 10716
6537  2504      6538 11735      6540  4300      6541  9655      6544 13279
6548 13853      6549 13889      6550   740      6553 16751      6554 15093
6564  6634      6565 11286      6567  7665      6568  2946      6570  8670
6571  5928      6574  8317      6576 10612      6577 10796      6578  3884
6582 11345      6584  3584      6585   122      6587 10962      6589 14294
6593  1523      6594  6143      6597 15660      6600  8720      6601  1280
6602 12312      6605  1444      6606  7265      6608 13092      6610  7155
6617 16257      6618  2403      6621  9850      6623 15646      6624    70
6629 10841      6632  7420      6634 11934      6635  1711      6636  7622
6639 15224      6640  7205      6642  1840      6643 10111      6644   435
6645  9851      6648 12918      6650 10387      6655 14215      6656  4898
6659     9      6660 16883      6661  8417      6663  6408      6664 13991
6666 15221      6667 15063      6668 16792      6669  1487      6670 12865
6671 10149      6673 12244      6676  1614      6679  7358      6680 11162
6682  9723      6686 16798      6687  6120      6688  8810      6689  2586
6692  4788      6693  8274      6696 10626      6698  1931      6700 11296
6701 11674      6702 14779      6704  7523      6706 16053      6708 13433
6709  1127      6713  8419      6715  6739      6716  6039      6718 13196
6719  8436      6721  4804      6731  3481      6733  9444      6735  3010
6739 14321      6740  5681      6742 10786      6745   413      6746  9065
6747  7981      6749 12329      6753 15703      6756  9767      6757  1633
6759  6297      6760  6562      6764 11102      6766 11261      6767   758
6768 15612      6769   152      6770 13196      6772 16257      6773 14130
6774  9948      6776 10597      6778  5691      6779 15133      6780 13371
6782 16250      6786  8728      6787  4711      6789   767      6790  3662
6793 16586      6795   793      6798  2086      6804   758      6807  2229
6811   269      6812   134      6815 15217      6817  5391      6818  9626
6819 16442      6821  2565      6823 15064      6824   708      6826  8330
6827  4304      6829  5933      6830  7961      6831  8514      6834 11623
6835  8126      6836 10152      6838 13007      6840 15452      6842 12356
6843  4114      6846  9723      6848  6386      6850 11496      6851 14012
6856  4303      6858 13308      6860 12065      6861  5979      6863  7428
6864  8127      6866  8064      6867  1273      6868 10694      6870  3729
6871  9864      6872  6675      6878   319      6879  1035      6880 10020
6882   172      6883  7046      6886  4033      6887  5057      6891 10984
6892  8945      6894 15759      6896    20      6897   423      6899  1507
6903 14204      6904  4096      6905 10260      6910  1837      6912  3001
6916 15696      6917  1603      6921  4447      6923  7106      6924  7361
6925 12558      6926 13559  6926  13559  12655 A HIT
8130  3207      8135  4028      8142  2269      8144  2292      8145  9057
8146  7942      8147  1382      8149  6196      8151 14215      8153  5534
8154  6074      8157  1154      8162  6561      8163  2789      8165  7764
8169  5400      8170  6378      8171 13727      8172 14752      8175  3662
8177  9061      8179 12013      8181  7918      8184  6372      8185  9241
8188 12232      8189  7657      8190 12872      8191 12217      8192  6078
8195 15386      8200 11415      8204 10965      8206  1548      8207  3139
8209 15686      8210 11046      8211  7967      8212  4220      8213 10487
```

Appendix CH4.5 Results of the message encryption/decryption program

```
8214 13629      8215 12549      8216  1762      8218  3759      8219  2824
8220 16496      8221  5604      8225 12804      8226  2993      8227  3791
8230 12961      8233  2785      8234 11500      8236 10389      8237 10889
8238 11444      8242 15824      8244  5846      8248  4764      8250  7711
8251  2392      8252  5075      8260 12794      8262  3457      8266 13948
8267  2976      8269 15796      8271  2361      8274  4510      8276  2705
8278  6192      8279  9923      8282 10121      8283  6426      8284  6032
8285 10225      8289   235      8291  4345      8292  7457      8298  9260
8299 16735      8300 15590      8301  4662      8303  4119      8305  8371
8308  6401      8311   528      8316  6741      8317 10149      8318 15433
8322  5222      8323  1535      8324  4931      8325 11768      8328 15532
8331  6391      8332 15537      8333   662      8334  2732      8338  4934
8342 14090      8344 16708      8345 15618      8346 16472      8347  5936
8349 14733      8350  2265      8351 13745      8352  5034      8354  4299
8356  8777      8357 16240      8358 16937      8360 14564      8361  1029
8364  4510      8366 10224      8367  2292      8370 16309      8371  5135
8372  4498      8374 14757      8375  3173      8377 12851      8380   203
8380  203  10376 A HIT
8382  4125      8383  5361      8384  1744      8385   274      8388  9934
8392  6435      8397 14775      8398  1910      8399  5919      8400  3756
8401 13485      8402  4881      8403 12704      8406  5885      8408  1284
8410  7081      8411  6189      8414 10519      8416  4507      8421 12612
8424 10877      8426  9693      8427   311      8429  5196      8431     4
8432  8203      8433 15066      8440 11271      8441   669      8442  4881
8443 15831      8447 13429      8449 14195      8450 15746      8451  3884
8453  5864      8454 14887      8455 12260      8456  6535      8458  6280
8460  8796      8462  8581      8465  6199      8466  2421      8467 10605
8469  6842      8470   426      8472 16932      8473 12670      8475  8729
8478  4253      8479 16773      8480 11088      8485  9441      8489  5391
8490  4419      8491  1328      8492  4267      8493 15338      8495  8646
8496 15608      8500  4961      8501 10313      8502 10072      8504 16969
8505  1383      8507 12948      8508  8664      8509  7915      8511 15928
8516   269      8517 15470      8520  7667      8521  5435      8522  5865
8523  6162      8524  7177      8526  6076      8528 12690      8530 12389
8532 11658      8533 13393      8534 12554      8535  5936      8536 12952
8537 13470      8538  5028      8539  1127      8541 13620      8542  7991
8543  3238      8544 14324      8546 13497      8548   554      8551   877
8553 11803      8554 16149      8555  1549      8556  9539      8557  8992
8558  4471      8559 11225      8561 13640      8566  7596      8571  5673
8573  7785      8576   290      8579 15792      8580  4866      8583  5746
8587  6563      8588  2229      8592 16843      8593  1698      8595 14743
8598 15226      8600  1681      8602 12937      8603   416      8604 15967
8605  2214      8607  9435      8610 15043      8612 10574      8613 13447
8614 16364      8615  5385      8616 13783      8619 14082      8620  1920
8621  4068      8622  6069      8623 15152      8627  4560      8628  3191
8632   334      8634  7807      8636 12419      8637  8767      8641 16074
8642 12309      8644 12741      8645  2193      8646 14308      8649  9310
8650 14649      8652 14564      8655 11460      8656  2466      8661  1434
8663  8845      8665  5064      8667 15508      8670 15944      8674  9693
8675 14356      8676  6915      8678  2078      8679  5057      8680  7278
8685  4124      8687 13948      8688  6641      8689 13985      8692 16447
8697  9368      8698 11394      8699  8100      8700  5055      8701 15953
8702 13454      8703  7344      8704  9756      8706   932      8707 14880
8708  6375      8710 10730      8711 11378      8714  8377      8718 13825
8719  7665      8720 16956      8721  9931      8722 12692      8723 14204
8726   716      8727 16819      8730  5275      8732  8736      8734  3023
8736 15892      8738 15388      8739 10311      8740 14800      8741 11882
8745  4931      8746  9121      8749 10716      8751  8281      8754 12867
```

Appendix CH4.5 Results of the message encryption/decryption program

```
8756  3742      8758 11742      8761 11721      8762 16801      8763 13199
8764  3608      8766   185      8767   431      8768  1734      8770 14680
8772 12181      8774  8550      8777  5105      8781 11520      8792 12442
8793   657      8795  8121      8796 13866      8797  4488      8798 11051
8802  2296      8803  1840      8806  6781      8808   583      8811  9846
8812 15708      8813  2991      8815 11543      8817 11213      8818   836
8821 14353      8823 10536      8824    83      8825 14154      8827 13603
8829 15132      8832  5293      8834  6702      8836 13214      8837  2002
8838  2378      8841  4134      8842  5886      8843  8080      8848   708
8853  4192      8856 14518      8857  5066      8858  4051      8859 12855
8862 10375      8864 15061      8866 15056      8867 12741      8868  9027
8871  8876      8873  7733      8875  4098      8876  6365      8880  6422
8882 13456      8883  3721      8886 13178      8890  1070      8893 14280
8894  6039      8896  6509      8898  7737      8900  1414      8901 10046
8903  4952      8905  3031      8906  3793      8908 10667      8913  7115
8916  4636      8917  3389      8919  3309      8922   143      8924 11562
8926  9774      8936 15211      8938 14871      8942 13203      8944 14994
8948 16760      8950 12120      8951   319      8954  5735      8955 15176
8956 13302      8957  6389      8958  3728      8960   671      8969  9778
8974 10157      8977  5262      8978   816      8980  9884      8981  2755
8983  8895      8984 16939      8986  4959      8987 12669      8988  5330
8991 13589      8992  3125      8994 16723      8995  3224      8997 15636
8998  3975      9000  8000      9007 12051      9013  2732      9014  9386
9015 16566      9016  4251      9017  1620      9019  3659      9020 12669
9021  4525      9025  2630      9027  1194      9029  1698      9030  1757
9031 13614      9033 11904      9036 15645      9038 15908      9039 16560
9041  4180  9041  4180  12792 A HIT
9042  6564      9045 13976      9047  4101      9049  2422      9050  6322
9051  2632      9053  3115      9055  5315      9057 12376      9058    64
9059 16350      9061 14510      9063  2694      9065  4624      9067  8173
9070  7042      9071 12606      9073  5983      9076  5106      9078  4345
9080  5175      9081  5751      9082 14880      9084  2889      9086   431
9088   792      9091  5418      9092 11481      9093 16488      9094  6087
9095  1726      9098  4997      9099  5549      9100  9743      9103   506
9104 16311      9106 14479      9107  1737      9111 14316      9112 13707
9113 12546      9116 12255      9117 16425      9118 16084      9119  5574
9122  1105      9123 10313      9125  2297      9127  5302      9129  5605
9131  8423      9132  2263      9133 13154      9134  3278      9135 12794
9136  6283      9137  7459      9139  6428      9141 16447      9144 15167
9145  5660      9146 16632      9148  6936      9149  3532      9150  3838
9152 16569      9153  2725      9154 10062      9155 13114      9156 16448
9157  2066      9158 10318      9162  9443      9164  2496      9166 15896
9167  5441      9171  4095      9172 12275      9175 11533      9177 10795
9179  5493      9180 15099      9183 10214      9184 16538      9185 14947
9186 13301      9189 16760      9190 15532      9191 15125      9192   101
9193  6231      9195  7595      9196  7543      9199   836      9200 10929
9201 11982      9202  6834      9203 11660      9204  5045      9205  4714
9206 14673      9207 15521      9209 12618      9210  6353      9217  4052
9218  2171      9219 10629      9220 13877      9222   732      9228 16652
9232  2290      9233  6187      9235  9623      9239  4922      9240  7111
10484  6851     10485  1425     10486  1654     10491 11383     10494 10120
10495 11763     10497 10697     10499 16680     10500  1846     10501  4188
10503  6366     10506  9934     10507  2342     10515 15523     10518  6637
10519  6881     10520  4675     10522 16516     10525  5167     10526  4668
10529 15235     10533 14759     10534 10694     10536  8030     10539   150
10540 14224     10541  6765     10542  3731     10544  8416     10548  2783
10549  9027     10555 15336     10556  1014     10557 16773     10559 10893
10560 14420     10562  1862     10563  2880     10564   993     10565  1510
```

Appendix CH4.5 Results of the message encryption/decryption program

```
10570 16791     10572 11555     10574  3972     10577 13629     10578  1292
10581  6222     10582 15069     10583  4951     10584   493     10585  1899
10586 11499     10587 13383     10590  8682     10591 16339     10594  5087
10596 13529     10597 11642     10598 11394     10602 11500     10604   626
10605  4556     10607  5524     10609  6147     10610 12421     10612  5406
10613 15511     10614 12679     10615  4959     10617 16707     10618  6761
10619   828     10621  7640     10622  6220     10626  1935     10627 15388
10628 15939     10629  6619     10630  3894     10631 10095     10632 13614
10633 12768     10634  3977     10637 15298     10639  7752     10640 14077
10642 13189     10644 14809     10645   596     10648  3760     10649 11168
10650   113     10651 13798     10655 16971     10656  5517     10657 10353
10658  9517     10663  3700     10664 16637     10665  5853     10667  6942
10668  3181     10672  7965     10674 13485     10675  6389     10676 11238
10678 11586     10682  4598     10692  1846     10694 12166     10696  4744
10701  1236     10702  2454     10704  7974     10705   828     10708 11085
10709  6821     10710  1334     10712  3647     10714  1661     10715 13388
10716  9970     10719 16314     10725 10500     10726  1457     10730  8685
10731  7833     10733 11754     10737 12260     10739  4096     10740  7078
10741  8824     10742  1185     10743 12682     10747 12198     10750  1791
10751 13805     10752 12137     10753  8409     10754 14954     10756  9912
10757  7922     10759 10911     10760  3224     10761 14518  10761  14518  42 A
HIT
10762  7587     10763  7408     10764  4159     10765 12376     10766  2861
10767 10939     10770   463     10771 12509     10772  4888     10773 15266
10775  2874     10776   234     10777 11066     10778 10136     10780  8121
10785 15763     10786  7999     10796  5734     10797  3844     10801  5681
10802  4261     10804  5312     10808  7791     10812  5810     10813  2466
10815  5783     10817  6486     10818  9893     10820  8320     10823  9567
10829 10730     10837 10118     10841  1018     10848  1801     10850  3717
10851 16762     10858  2471     10859  5238     10860 14609     10861  7622
10863 14451     10865 11734     10866  7814     10867 10813     10872  3784
10873 13957     10880 10271     10881 12431     10882  4284     10884 11882
10888   652     10889  5308     10891  2248     10892  2919     10895  9953
10896 15729     10897  5001     10899  6162     10900  6052     10904 11567
10905  3108     10911 16979     10914 16536     10916  9321     10918 12087
10920   782     10921 14066     10923  1140     10924 12438     10928  7807
10930 14906     10933  3749     10934  2193     10935 16582     10936  9722
10937 12587     10938 12952     10940  8570     10943  2494     10946  2359
10950 13006     10951  1489     10956 11762     10957 12587     10958  3197
10959  4086     10962  5003     10964  1876     10965 12914     10966  8470
10969  5542     10970 14562     10972  8073     10976  9774     10978 14474
10979  4259     10983 13178     10985   483     10986  3606     10988  5422
10989  9939     10990 13329     10991  4272     10992 16814     10993  3863
10995 11102     10998  9185     11003 16472     11005  4304     11006  2708
11007 16452     11010  4346     11011  9432     11012   314     11013  4627
11014  1964     11015  3725     11017 10938     11020 12946     11025  5784
11026 15750     11028 14955     11029  9317     11032  9984     11039  8503
11041  1837     11043 12461     11044  2487     11046 14989     11047  1941
11048  3349     11049  9364     11056 13799     11057  1977     11059  6447
11060  9530     11062 16781     11064 16468     11066  4140     11068 14592
11074  2936     11075 11968     11078 10061     11080 15279     11084 14796
11085  4033     11089  9722     11090 16711     11093 14296     11094  2916
11675  4793     11676   714     11677  9310     11678  4744     11682   883
11684 10614     11686  2296     11687 16909     11691 13197     11693  3896
11695 12069     11699 14843     11705 16860     11709  7842     11710  1929
11712  5251     11713  8676     11714 14066     11715  8424     11716  6211
11718 15077     11719  2869     11720   381     11722  4136     11725  6733
```

Appendix CH4.5 Results of the message encryption/decryption program

```
11726  1009     11727  5115     11729  7874     11730  3040     11731 16283
11732   991     11733 13771     11736 13200     11739 13743     11741  7504
11744 10024     11745  3755     11746 15730     11748  4117     11749  2544
11750  9102  11750  9102  16 A HIT
11751  2505     11753  7438     11756  9956     11757 12403     11758  2146
11762  7082     11764 13347     11765  2209     11766  2453     11769  8804
11771  5911     11773  6386     11775  4552     11782   185     11783   680
11786 10967  11786  10967  16088 A HIT
11790  7347     11791 12033     11793  3706     11799  3569     11800  6750
11802 15427     11803  6535     11806  9903     11807  9907     11808 10609
11809  1236     11810  3939     11812  4726     11815  6183     11817  8540
11818  7265     11821 15180     11822 13734     11823  1876     11824  2936
11825 10396     11828 13985     11829  1186     11830  5906     11831  3431
11833   874     11834 13998     11835 12356     11838 12957     11839  6435
11841  6033     11842  6878     11844  5885     11848 16352     11850  7967
11853  4777     11854 11178     11857 14756     11858  2443     11860 11883
11861 14429     11862 16203     11863 12128     11864 16344     11868 12048
11870  3296     11874  5759     11878  7657     11880 16851     11881  7569
11885 11990     11886  6351     11887  1254     11888  8170     11889  6059
11896   961     11901  6720     11903 16546     11904  3226     11906  3100
11908  7278     11911  4159     11914  7347     11916  2386     11919 14427
11920  3920     11921 11805     11922  9474     11925 10913     11926  6428
11929  4827     11932 10244     11933 12874     11935 15858     11939  2155
11941 15955     11942  4234     11943  7517     11944 14428     11945  7164
11946  5158     11947  2217     11949 10418     11950   270     11952 14093
11953  5629     11954  2212     11955 13237     11956  2020     11957 12405
11958  6471     11959  2933     11966 14944     11968  6563     11969 12781
11973 14737     11974   709     11975  3030     11976 15955     11977 11889
11978 11283     11979  9078     11980 12047     11984  2958     11985  8772
11986  9644     11987 15939     11991   678     11994 13779     11996 16179
11997  9722     12003 13045     12005   929     12008   165     12013   993
12014  4559     12015 10939     12018  6705     12020  2489     12021  9198
12026 16632     12027 13348     12030  4648     12032 16364     12042  4439
12044  1158     12045  3264     12046 12797     12048  2530     12053 12659
12061  3683     12062 10360     12063   308     12065 14029     12071  1567
12074 16590     12075 11798     12076  9687     12077 13729     12080 13356
12081 10024     12082 11157     12083 12673     12085 14667     12086  8086
12087 15499     12088  6661     12090  2462     12096  7574     12099 14369
12101  9520     12103  4595     12104  6193     12105 11804     12107  5532
12109  1257     12111  2646     12112  8133     12114  3237     12118 13648
12121  2705     12122 11334     12123   427     12125   784     12126  7922
12127  2004     12128 12587     12131 14211     12133  2562     12134    37
12135  9429     12140  4803     12143  9822     12145  5882     12146 10099
12147 13969     12148  4122     12150 10138     12151  8300     12154 12688
12155  9859     12158  2617     12160 10911     12161  4933     12162  2785
12163 13730     12165  8715     12166  7624     12168  3885     12171  9126
12172 11799     12178 10226     12182  5118     12183   692     12184 12119
12185  2832     12186 14367     12187  6038     12195  1059     12199 14837
12200  1101     12203  4094     12204  1852     12212 10554     12213  3604
12218  2950     12220  5318     12221   626     12222  3461     12227 15635
12230  5988     12233 10299     12235  6252     12238  7630     12239  9299
12241 16698     12242 16945     12243 14677     12249 16529     12250  6053
12250  6053  259 A HIT
12251 15602     12252  8443     12253 10399     12259  9923     12263 16750
12266 14479     12272  4118     12273   284     12274 15699     12276 13433
12278 14399     12279 14434     12282  1863     12283  9040     12284  5323
12285 11115     12287  7677     12288  9778     12289 16478     12290   173
12292 15569     12293  7816     12294 16612     12295 15108     12297  2497
12298  6721     12299  6426     12304  4236     12305 13652     12306  8395
```

Appendix CH4.5 Results of the message encryption/decryption program

```
12309 13362     12310  3444     12311 11608     12313 10440     12315 11884
12316 12279     12318  6034     12319  7101     12321  8146     12325  8695
12326  1263     12327 10947     12328 16042     12330  8615     12333 10218
12334  8813     12335  7269     12338 10499     12339 13357     12340  1798
12341 16246     12345  3839     12346 15612     12348  9365     12349  2810
12351  1558     12352  1758     12355 16159     12357 14039     12359  8170
12360 16567     12361  6910     12362  4997     12364 10260     12365  4853
12366 15744     12367  3567     12369  6856     12371 10947     12373 12326
12374  1019     12375  8303     12376 13724     12377 13557     12379  6869
12380  3824     12381  2791     12383 12226     12386   756     12389  9323
12390  7722     12391  1423     12392  3660     12393 13385     12394 14229
12395 16157     12396  2955     12399  1598     12401  9937     12402  8727
12403  3273     12404  7953     12405  8393     12406  2282     12407  2564
12408  4952     12409  6756     12410 16601     12413 15129     12414 14988
12415 13786     12416  4928     12418  1383     12419  9705     12421   169
12422  4533     12426  5242     12428   165     12429  4020     12432  7742
12433  4297     12438  6605     12439  1938     12441  2987     12442  4237
12443 12495     12444  8509     12448  3040     12449 14194     12450 14671
12455  3011     12457 11149     12458  5874     12460 16317     12465  6266
12467 13698     12468   859     12472  9998     12473  7262     12474  4447
12476  3613     12479  2438     12480  3874     12481   682     12484 13744
12494  3766     12495   436     12496  7358     12497 12811     12498 14733
12501  8548     12504  7492     12505  2080     12506  1724     12508  7445
12513 14345     12515   902     12516 10702     12517 11488     12518 11719
12521  1884     12526   368     12531 10486     12533 10054     12535 11664
12536  5573     12537 13279     12538  2135     12539 12690     12544  4651
12545 16181     12546  4451     12550 16473     12551  8455     12553  5348
12555 15326  12555  15326  9185 A HIT
12556  4068     12558  7403     12559  8003     12560  4619     12561 16942
12562  4997     12566  8763     12568  3018     12569 12492     12570 10242
12572 13292     12575 16188     12576 12188     12577  3028     12579 15140
12581 13056     12584 15662     12585 13913     12586  6301     12589  1205
12590  3645     12591  6567     12592 14800     12593 11354     12598  9407
12599 16026     12601   974     12603  2161     12609  4507     12610 16918
12612 10277     12614  6311     12615  5119     12616  7926     12617  4037
12620  7082     12622  6117     12626  1530     12628  5550     12630  7005
12632  1492     12636  2995     12638 12901     12639 11451     12640  1289
12644  8405     12646 15978     12649  2175     12650 14479     12651 11846
12653  3459     12654  4708     12656  1054     12663  6296     12665 10419
12667  4711     12669  3385     12671  3976     12674   105     12676  2341
12677  5443     12680  6464     12681  4003     12688 14059     12690 14800
12691  5482     12693 13224     12694  6708     12695  1731     12697  5422
12698   828     12699  8657     12706  1172     12707 11192     12708  4871
12709  4501     12710 15742     12711  4952     12716 16028     12717  8794
12719   215     12720 13045     12724  5028     12726  2222     12729  9491
12732 14582     12736  6883     12738 11224     12739 14944     12741  7169
12742  8539     12743  2016     12744 15571     12745  7977     12747  8130
12748  7779     12749 14629     12750   901     12754 15759     12756  8780
12760  5028     12761 12535     12765  9778     12767 10676     12768  1740
12769  4113     12771  5911     12774  5632     12775 13797     12777  1370
12778  7573     12779 11503     12781  4968     12784 10892     12786 16366
12787  7888     12788   678     12791  7430     12792   782     12794  5357
12795 13314     12796 16400     12798  5924     12799  1664     12800 16685
12806 14950     12807 10588     12810 15687     12812 14375     12816 10151
12820   744     12826 15903     12827  2860     12829 11902     12830  1846
12831   181     12838 16655     12841 16480     12845  4068     12848  7374
13475 13048     13476 11480     13479  9753     13484  9923     13485  1194
13486  6343     13488 14967     13489  1740     13490 10955     13491  3744
```

Appendix CH4.5 Results of the message encryption/decryption program

```
13492  9242     13495 11968     13496   400     13497 15031     13498  4261
13500 12178     13502 15234     13505 13985     13506 15766     13510  8557
13512 16418     13514  2647     13515   886     13519  7601     13520 14052
13521  5607     13523 12902     13524  9932     13525 16858     13528 12884
13530  7524     13531 16747     13532  4494     13533 12032     13534  2296
13535  3992     13541 13789     13543  9784     13549 11645     13550 11678
13551 14033     13552 14814     13553  5066     13554   777     13556  7012
13558  1237     13561 12811     13563 16921     13564  9782     13572  6812
13573  9242     13580  3256     13582 11408     13585 13258     13587   608
13588 11735     13589 14068     13591 14655     13592 13179     13594 14480
13596 16578     13597   860     13598   624     13599 10916     13602  1873
13603  7473     13610  1400     13612  1021     13618  7709     13619 16884
13621  9127     13622  3181     13623  4249     13625  2705     13626   866
13627  2785     13629 10060     13633  9629     13635  1027     13636  5350
13637 14869     13639  2072     13640  3929     13641  4766     13642  6821
13643  1306     13645 14989     13646 15161     13647 13315     13648  3031
13649  7374     13650 12185     13651 12836     13654 16739     13658   571
13659  8345     13663 16582     13664  6233     13665   887     13666  1395
13669 16933     13670  2167     13672  6697     13675  9233     13676 14469
13678 11206     13680 16967     13681  6334     13682  7409     13683  4598
13685  9953     13686  1522     13687  3412     13688  1743     13689 13097
13695 14632     13696  9310     13697  7945     13708  7729     13710  7479
13711 14206     13715  6479     13716  6640     13717  3609     13718  1629
13721 11994     13722 16311     13724  9332     13725  4225     13729  1047
13731 15239     13732  2826     13733 12577     13736  2880     13737  8929
13740  7435     13741 16026     13745  2434     13747  1019     13754  5766
13757  4858     13758  2721     13762  7487     13763  6535     13765 12780
13766 16748     13769  2199     13771 15324     13772  5885     13773  1081
13774 15556     13776   488     13778  4778     13779  6356     13781   789
13782  7232     13783 13800     13784  2755     13786 10117     13787   610
13788 16095     13791  6435     13792  6274     13795  1660     13796 11849
13799 15321     13800  9659     13801  8083     13802  8227     13804  3076
13806  1957     13809 13017     13816 16154     13819 11962     13820  9404
13823  8285     13824  5183     13827  7700     13829  8363     13832  5411
13836  9885     13837 15801     13839 14776     13842 10466     13844  2673
13846  2053     13847 16948     13849 15604     13853  3772     13855  2224
13856  8003     13857 11219     13860 13189     13862  8658     13863   928
13864  2839     13865 12973     13867  9160     13868 10708     13870 10342
13875  3426     13879  7468     13880 13876     13881  3993     13883  3461
13888  6443     13889  5516     13890  6701     13892 11290     13893 16916
13896 11735     13897  5552     13898 13525     13899  6051     13900  6110
13901  8972     13902  9811     13903 15800     13906   479     13907 16185
13909  9869     13913  1454     13917 11985     13920 13919     13924  4921
13925  1224     13927 16621     13929  2137     13930 16358     13933  7552
13933  7552  2298 A HIT
13936   606     13938 11140     13940  6792     13941 16871     13942 12988
13943  9136     13944  9580     13945  5316     13950 10875     13952  3681
13955  7657     13956 15958     13958  1652     13961  7967     13962  7610
13967  5651     13971  1999     13972  4602     13980  4039     13981 14971
13986  9851     13987  5201     13988 16555     13989  6594     13991 16635
13993 14628     13995   760     13998  5058     14002  3700     14004  6619
14006 16369     14007   424     14010  7710     14013 11219     14014  3756
14016  4363     14017  6903     14018  7106     14020  7435     14023  9496
14028   518     14030 12490     14031 10052     14033 12600     14034  9592
14594 15598     14595  9194     14596 10554     14597 13525     14600  6162
14601 12412     14602 15425     14603 11291     14604  7271     14605  9027
14606  1821  14606  1821  2515 A HIT
```

Appendix CH4.5 Results of the message encryption/decryption program

```
14609  1704     14610  7487     14611 14578     14614 16644     14617  2209
14620 10266     14622 10201     14624  1833     14625 10187     14627  4447
14628 16939     14630  3115     14634   600     14636  7538     14638   905
14641  2520     14643  5759     14644  4468     14645  5665     14651 11052
14652  4012     14653  5763     14656  2093     14657 15388     14659  7543
14667  2000     14669 15637     14671  6742     14672  3806     14673 16472
14676  1775     14678  1071     14685   789     14687 16268     14690 12019
14692 14372     14693  5099     14695 14235     14698  1758     14699   289
14701  6773     14702 15946     14705  1382     14706   186     14710   222
14711  2884     14712  8421     14714 14304     14720  6591     14722  5235
14723 13083     14724 16500     14726 11394     14727  5577     14730 12346
14731 15659     14733  3546     14737  2616     14738 12194     14742  8910
14743 10284     14746  6070     14747 12400     14748 16131     14749  5802
14751  6841     14752  3237     14753 10260     14754 10349     14756  7870
14757  3856     14759 10139     14760   309     14762  9315     14764 16344
14766 10528     14767  9646     14768 11588     14769  7793     14772  9233
14775 14436     14776 12194     14778  9397     14779  6733     14781  4246
14783  3368     14784 11881     14785  9760     14791 11805     14793 13543
14794 10545     14797  7040     14798 16651     14800 15860     14802  7271
14808 16555     14812  4423     14813   757     14814  4787     14815  7346
14816 11511     14817  8465     14819  8434     14820  6151     14821  1641
14822 12969     14823 10846     14824  5537     14825  7157     14826 12799
14827 15325     14830 12260     14831     6     14832 10158     14833  2409
14836 14915     14839 13011     14842  5183     14843  2075     14844  3723
14845  2424     14846 14302     14847  7358     14848  6025     14849  4975
14850  9876     14851  6247     14853   841     14854 13511     14855  5922
14857  9977     14859  5518     14860  5713     14861  7796     14862  9392
14864  8910     14872  1574     14877  5380     14878 10734     14879  6151
14880  6514     14881  9042     14882 15172     14884 14184     14885  2749
14886  9613     14887  4969     14889  3736     14890 13443     14891  5568
14900 14319     14903 14253     14905 11040     14907 16909     14908 15612
14910  6400     14911 13976     14912 11704     14914  1893     14915  9224
14916  6523     14917 13361     14918 11388     14922 12415     14925  8612
14926  8798     14927  2016     14930 10396     14933  2865     14935  6850
14937  5858     14938  7917     14939  5649     14941 13279     14942  4949
14943   506     14944  7664     14945 16730     14947 13485     14948  9960
14952  2104     14956 13695     14961 14138     14962  6267     14965 13214
14966  4279     14970  8861     14974 10754     14976  9465     14977  9843
14978  8386     14979  1209     14983  2037     14985  2702     14986 16773
14991  3289     14994 11178     14997   726     14999  8480     15000 10708
15002 16179     15003  1694     15013  3847     15014 13141     15016  6304
15017  4793     15021 10393     15024 11806     15033 16269     15035  9231
15038 13433     15039  3614     15045  4763     15046 12120     15047  8486
15048 14947     15050 12498     15052 13249     15053  5509     15054  3560
15055 14848     15058 13895     15059 13859     15063 13140     15064 13909
15067 16399     15068   905     15069 16994     15072 12614     15073  3605
15079 14071     15080  2573     15081 16994     15082 13734     15083  4929
15086  1483     15088  3580     15090 13586     15091  1395     15092 15181
15093  8109     15100  5729     15101  4195     15102  4128     15106  3312
15107  3007     15109 16209     15111  1836     15112 14567     15115 10042
15121   270     15122   979     15125  7310     15126 11738     15128  9934
15131 15414     15132 11069     15134  4617     15135  7899     15139  4366
15140  6149     15141  8946     15142  1735     15144 11313     15145  4717
15146  7661     15148  6153     15149  9209     15154  2864     15160  3949
15164 14996     15167  4640     15168 12169     15170  1242     15171  3912
15173  5975     15175  9495     15176  8972     15177 10665     15178  6040
15179  5085     15180  4692     15181  5770     15182  9659     15186 11500
```

Appendix CH4.5 Results of the message encryption/decryption program

```
15756 10949     15759 16464     15760  7626     15761  7024     15765 10984
15766  8653     15768  7496     15769  3718     15771  8015     15772 11758
15773 10751     15777  5800     15781  4051     15782 14703     15784  3747
15790  4118     15792 14983     15796 16198     15797 13201     15798  9715
15799  2866     15800 16256     15801  2368     15802  4632     15804 15928
15805  6252     15809 16500     15810 10520     15812  5769     15815 13085
15816 11908     15817 10900     15820  8061     15821 15499     15824  2992
15825  2427     15826 14260     15827 12198     15828 14091     15831 11944
15833 16916     15835  5143     15836  3634     15837 13043     15840  9682
15841  6833     15842  5372     15843  9978     15844  7756     15845 11040
15847  8914     15849  4272     15850  1907     15852  8663     15854  7673
15855 14378     15857  4437     15861  7274     15864 11800     15865  9549
15866 12227     15867  7926     15871  9587     15872  6896     15873 16443
15878  9156     15880   148     15881 11061     15882 10353     15883  6610
15884  8058     15885 16760     15888  7866     15891 15590     15899  7716
15900  9121     15901  2958     15902 12096     15903 11300     15904 12435
15905 15016     15906  8767     15907  2870     15908  2590     15909  4014
15912  6334     15913  6956     15916 13836     15923 15659     15928  2749
15929  8905     15933 10640     15935  1391     15936 16814     15940  2975
15941  5318     15946   159     15947  5372     15948 14221     15950 14913
15953 10400     15955 11273     15959 15768     15960 13324     15961  6093
15963 11990     15964 16198     15967  3083     15968  4632     15969  9042
15971  1670     15972  9804     15976 10790     15977  1855     15978 11475
15979 11366     15981  4484     15984 12400     15986   565     15988  9932
15989  2772     15992  4777     15993  2082     15994 10592     15995  4770
15997 12295     15998  5032     16000  9188     16002   988     16003  4336
16004  4312     16005   836     16010  5085     16015 12669     16016  8330
16017  9399     16018  5092     16021 12838     16022  8406     16026 13106
16027 11453     16028  7038     16032 13853     16037  7344     16041 12550
16042 15325     16043 10299     16048  9521     16049 11536     16051  4033
16053  2525     16054  9892     16055  2643     16056  7638     16057 12577
16060  5406     16061  6158     16063 12410     16064 15870     16066  5148
16067 13287     16068 13315     16071  1837     16075 13487     16077  7100
16081 10577     16083  6356     16086  2175     16087  1572     16090  7267
16092  3791     16093  5907     16094  8276     16095 12365     16097 14182
16100 14451     16105 12114     16107  7174     16109 10746     16111  6741
16112 10527     16114  8842     16115 14884     16119   163     16120  5576
16122  5214     16126 12211     16134  7698     16135 14041     16136 11930
16138 11666     16139 13984     16140  6696     16141 10319     16142 14628
16143 10226     16149 12439     16150  4769     16151  3331     16153  9209
16154 13376     16156  2719     16157 16059     16162 10127     16163 12151
16164 13497     16165  1633     16166 15155     16167 11061     16169   431
16170 11784     16172 16496     16175 13681     16176  8655     16183 16942
16184 13144     16187 13491     16188 11173     16189 16447     16190  4304
16192  6143     16193  7706     16194 16383     16195  9998     16197  6644
16198 11008     16199  6232     16200  3112     16201  6355     16203  8048
16206 14527     16207  3203     16209 12724     16211  9032     16212  5882
16213 16743     16215  4549     16225  5482     16227  2653     16230  4195
16233 14880     16236  3648     16240 16499     16241  1927     16242 14621
16243   740     16246  1779     16249  6482     16253 13347     16254 11267
16256 14593     16257 16343     16262  8299     16263 11102     16265  5850
16268 14771     16269   624     16270  7573     16272  3790     16274  3547
16275  7101     16276  9912     16278 15709     16279  9514     16281  2549
16282  4705     16283 14398     16284  6185     16288  7970     16289 13296
16290  1881     16294    37     16295  8294     16296  6737     16297  4753
```

Appendix CH4.5 Results of the message encryption/decryption program

```
16298  5963     16302  3579     16304  4020     16305 13937     16307  7746
16308 16214     16312  2226     16313 16921     16316   383     16317   306
16319 12022     16320  1763     16322  9420     16323 14026     16330  6247
16332  6355     16333  1077     16334  9694     16336  2091     16337 16119
16338  9138     16341 11942     16343  3683     16347  1296     16352  2938
16354 15621     16356  5579     16358 15699     16360 12609     16361 11800
16363 11022     16366  4385     16368  1229     16369  9179     16370  3393
16372  3536     16374  5004     16376 12131     16377  4628     16378 16581
16379  4096     16382 12950     16384  8907     16387 10112     16388 12864
16391 13650     16392 16497     16393 14192     16398 10313     16399 11149
16400  1698     16401 11009     16403   121     16407  7939     16408 11834
16414  5702     16417  3629     16419 13616     16422  5224     16425 13683
16427 11002     16428 13657     16430  2954     16433 14520     16434 11008
16437 13916     16439  1230     16440 11278     16447  9773     16450  1007
16452  3669     16455  9811     16456  1230     16459 14311     16460  3588
16461 16062     16462 12012     16463  6591     16464  3929     16466 13797
16476 11722     16481  5681     16485 11350     16487  1021     16488 13788
16492  8030     16493  2616     16496  3657     16499   138     16500 11548
16503 16301     16504 15532     16505  7230     16506 15860     16508  2754
16509  3986     16510  6378     16511 12741     16513  8743     16515  1510
16517  8943     16521  8601     16523   842     16524 13509     16525  7622
16526 10968     16527  4384     16531 13369     16532 11616     16534  8701
16535 16397     16536 15633     16541  3270     16542 12373     16547  8677
16549 14632     16550  3076     16552  8002     16553  2809     16558  9924
16559 16501     16561  6919     16562  7232     16563 14275     16564 13609
16566 11059     16567  6724     16569  8008     16570  1911     16571 11775
16576  2530     16578  9630     16580  2033     16581 13714     16584 14417
16585 16418     16587  3279     16588 10552     16590 16308     16592  3248
16594  8636     16596  3938     16598  2971     16600   620     16601  3082
16602  6555     16603 13409     16604  3128     16609  9366     16610 11665
16612 15497     16613  8067     16614  7346     16616 16536     16617  9536
16620 10694     16622 10474     16624 10131     16625  4177     16626  6196
16627 12794     16628  2939     16629  6602     16631 13960     16637 13531
16642 12874     16643  1679     16644  4088     16645  8034     16646  2102
16648  6228     16651 14495     16652   471     16653  4345     16657 12809
16658  7957     16659  9111     16660  6243     16661  8636     16662  3681
16664 12005     16665 14252     16666  3442     16667 11306     16668 14240
16669  2366     16670  2963     16674  2724     16675  2732     16676  5659
16678  7430     16680 13020     16682  8868     16686 12642     16688  1823
16690 14926     16691  8586     16693  1165     16695 15074     16698  6827
16699  4691     16700  5810     16701  6610     16702  6112     16705  3003
16708  3120     16711  5344     16712  8807     16713 14949     16714 13506
16715 15305     16716 14655     16718   574     16719  7685     16723 16225
16726 16733     16727  9877     16728  9243     16730 14567     16731  5189
16733 10277     16735 13998     16738  2948     16740  2983     16744  1023
16746 14813     16747  8771     16748 16143     16750  1133     16751  9181
16752 10237     16753 13336     16754 16564     16758  5669     16760 13905
16763  5975     16764  7017     16767  6030     16768  2933     16769 13174
16770  4227     16772  1410     16773  2216     16775  3192     16776 14916
16778  8386     16779 15431     16784  7131     16785  3269     16787  6268
16791 15911     16792  3380     16794  6233     16795   573     16798 12868
16799 11644     16800  2866     16803  1816     16804 14888     16808  6447
16809 14658     16810  7429     16811  9945     16812  8728     16813 14786
16815 10146     16819  1061     16821  4859     16823 13649     16824 11134
16825 10982     16828 10430     16830 13760     16832  7348     16833 12454
16835  6516     16837 13506     16841  6093     16842  2737     16843 11678
16846 12822     16853 15696     16854  5970     16856  3349     16859 15443
16860  4651     16861  8763     16863    23     16865 13960     16868  1887
16869 15239     16871  4356     16873 14510     16874  1727     16875  5806
```

Appendix CH4.5 Results of the message encryption/decryption program

```
14918 11388     14922 12415     14925  8612     14926  8798     14927  2016
14930 10396     14933  2865     14935  6850     14937  5858     14938  7917
14939  5649     14941 13279     14942  4949     14943   506     14944  7664
14945 16730     14947 13485     14948  9960     14952  2104     14956 13695
14961 14138     14962  6267     14965 13214     14966  4279     14970  8861
14974 10754     14976  9465     14977  9843     14978  8386     14979  1209
14983  2037     14985  2702     14986 16773     14991  3289     14994 11178
14997   726     14999  8480     15000 10708     15002 16179     15003  1694
15013  3847     15014 13141     15016  6304     15017  4793     15021 10393
15024 11806     15033 16269     15035  9231     15038 13433     15039  3614
15045  4763     15046 12120     15047  8486     15048 14947     15050 12498
15052 13249     15053  5509     15054  3560     15055 14848     15058 13895
15059 13859     15063 13140     15064 13909     15067 16399     15068   905
16876  3373     16879  4417     16880  1089     16881  2434     16884 14295
16885  2673     16893  1459     16897  8298     16898  9848     16899 13154
16901 16112     16902  3645     16904  8816     16905 12573     16908   371
16911  9937     16912 12809     16913  5161     16918 10323     16919 13887
16920    43     16922  9693     16930  8034     16933  8561     16935  3329
16936  7524     16938  5608     16939 10327     16943 14764     16945 14621
16946   126     16948  6523     16953  4931     16954  3916     16955   816
16958  5004     16960 10592     16961  1553     16967  3442     16976  2660
16979  9335     16980 11794     16982   772     16983 11642     16985  2714
16987 16091     16989 12166     16990   792     16996  3300     17001  7136
17003 16350     17004 14102     17005 15111     17008   620     17010 14564

   TOTAL NUMBER OF HITS= 13 AND THE ORDER OF THE GROUP IS = 0
   1       0     0  A MISS
   2   11750  9102 A HIT
   3    1608  4734 A HIT
   4   12555 15326 A HIT
   5    6926 13559 A HIT
   6   13202 12559 A HIT
   7       0     0  A MISS
   8       0     0  A MISS
   9       0     0  A MISS
  10   14606  1821 A HIT
  11     954 11503 A HIT
  12    9041  4180 A HIT
  13   12250  6053 A HIT
  14   10761 14518 A HIT
  15    8380   203 A HIT
  16       0     0  A MISS
  17   11786 10967 A HIT
  18       0     0  A MISS
  19       0     0  A MISS
  20       0     0  A MISS
  21       0     0  A MISS
  22   13933  7552 A HIT
```

Appendix CH4.5 Results of the message encryption/decryption program

```
STATE OF CIPHER TEXT IN RELATION TO THE POINTS ON THE ELLIPTIC CURVE
1  7842  2346  1507  16417
2  11750  16  2704  393
3  1608  12565  2628  2950
4  12555  9185  1204  15324
5  6926  12655  528  12647
6  13202  9812  2006  9629
7  8918  7380  1828  1936
8  1253  4343  2205  14215
9  13617  9536  2715  13753
10  14606  2515  1619  915
11  954  8446  2113  10358
12  9041  12792  2710  401
13  12250  259  1525  4099
14  10761  42  2708  11651
15  8380  10376  2314  12327
16  3664  11782  1816  6343
17  11786  16088  2702  11279
18  4520  7286  311  4188
19  12668  11909  2109  15967
20  7545  8423  2718  3716
21  1926  8348  610  3309
22  13933  2298  21  8173

       -----THE DECIPHERED MESSAGE-------


  1    1507       1507 16417       7842  2346       1507 16417
  2    2704       2704   393      11750    16       2704   393
  3    2625       2628  2950       1608 12565       2628  2950
  4    1201       1204 15324      12555  9185       1204 15324
  5     527        528 12647       6926 12655        528 12647
  6    2006       2006  9629      13202  9812       2006  9629
  7    1827       1828  1936       8918  7380       1828  1936
  8    2205       2205 14215       1253  4343       2205 14215
  9    2715       2715 13753      13617  9536       2715 13753
 10    1619       1619   915      14606  2515       1619   915
 11    2113       2113 10358        954  8446       2113 10358
 12    2710       2710   401       9041 12792       2710   401
 13    1524       1525  4099      12250   259       1525  4099
 14    2706       2708 11651      10761    42       2708 11651
 15    2314       2314 12327       8380 10376       2314 12327
 16    1815       1816  6343       3664 11782       1816  6343
 17    2702       2702 11279      11786 16088       2702 11279
 18     311        311  4188       4520  7286        311  4188
 19    2109       2109 15967      12668 11909       2109 15967
 20    2717       2718  3716       7545  8423       2718  3716
 21     605        610  3309       1926  8348        610  3309
 22      20         21  8173      13933  2298         21  8173
```

Appendix CH4.5 Results of the message encryption/decryption program

```
 2  4835        4 16541        7  3799        8  2632       10  9691
12 10031       13 16051       15 12516       16 12088       18 13186
19  9968       21  8173  21  8173  8173 A HIT
22  7965       24 12986       25 12332       27 12618       29  4603
30 11849       33  5600       37  3030       42  1188       43 14253
44 10474       45 15296       46  4691       47  9130       55  7007
56   143       58 13948       59  5048       60  4832       62 10629
63   841       66  8655       67 15587       72 13994       73 14948
74 16998       76  3339       77 10711       78  1959       80 10575
81 13609       83  9230       85 12119       88 13594       90 14153
92 10123       94 11914       95  9634       97  8804       99  9924
101 16674     107  5996      108 14893      109   682      111  5328
112  4525     115  3883      120 14043      121  5789      124  3938
125 11383     126  1629      127 11618      129  5936      132 11640
133  8916     134 10066      140  9048      141 15295      142 14068
143 14046     145  8922      146 11350      149  3948      150  4928
151 13385     152 13877      157 14503      159  6276      162  4225
163 11877     165  6676      166  7629      167  4881      169  3970
171  6117     173  3588      175  9700      176  2179      177 15572
180 10650     183  4108      184 16561      186 14819      188   342
189  1779     193 11651      194 11168      196  6187      199 10962
201  6837     204  6711      208  1398      209  3201      212 12396
214  5649     215   897      218 14209      226   606      234  2647
235  8469     237 10601      243  6211      245   434      246  8513
247 13022     248  8612      250  7835      252  3856      258 11130
260 15862     261 11060      263  4855      264 11142      266  4691
267 16992     270 14767      271  5308      273  9443      274 15473
275  1542     276 14734      278  8813      279  9984      281     6
282 11984     283 12680      285  8079      286 14814      288  2343
291  3938     293  4167      294  8892      295 14108      296 14681
299 16631     300 11615      301 10259      302 14050      305 16967
307  5906     308 15282      311  4188  311  4188  4188 A HIT
313  4403     316 16190      318  3951      320 14439      321  2769
322 13689     323  5656      324  7756      325  8314      327 14469
328  8230     331 11943      333  5874      337  6479      338 15026
339  3355     343 11244      344  7401      345 10474      348  1470
349  7473     351  2793      354  9924      356 12616      357  2465
358 16956     359 16067      366 13609      367 13056      368  1322
370 12109     372 11155      373  4510      374 13589      375  9100
378  3588     379  9224      380 11350      383  9891      384  8281
385 10375     386   719      387 16922      388  2573      389  3442
391   575     392 10134      393 15627      394   237      395  6881
397  9709     398  2893      399  3001      401  1898      403  5857
406  7040     414   620      418  1723      420  6379      421 13300
422  7671     423 11296      424 14388      425 13156      426  3205
428  9579     429  7341      432  9967      433 11085      434 15579
439  4239     441 16304      442  1460      444  8253      445  3613
447  8034     453 12809      455 15961      457 10251      458 15219
461 15209     465 14601      466   834      467 14555      471 13506
472 14278     474  6485      475  4279      477 15472      480  3653
481  9884     483  6332      485 11127      487 12407      488 14676
```

Appendix CH4.5 Results of the message encryption/decryption program

```
490  7425      491  5947      492  8359      493  8159      496 14509
498 13000      499 12521      500  2292      501  4134      505  6623
508  5890      514 12227      515  1755      518 12315      525 14566
526 13960      528 12647  528 12647  12647 A HIT
530  3788      532  8494      533 16604      534  4544      535  2413
539 16019      540  1299      543  8086      545  5988      546 15543
548  7955      555  8058      556  2289      558 12411      559  6531
560  5731      563  6420      566  6691      567 10913      569 15423
571 12946      572  8058      574 13876      576  1779      579  8631
582 16382      584 16130      585  2083      590  6699      591 14924
592  5157      593 11489      596  8684      597  4489      602 16569
603  1045      610  3309  610  3309  3309 A HIT
615 12835      618 13048      619 11530      620 14954      621  8630
622 16154      623 10387      629  4719      631 12227      632 15291
633  1431      635 13789      639  5983      640  3028      641  4501
643 10231      644  1796      647 10579      652 12027      654 10458
656 14330      658  6102      659 13807      661  3733      662 11433
663  8118      667  8405      670  2946      671  8385      672 11413
674  2477      676  7337      679  8574      680  7145      681  8483
682  9398      683 16547      687  6759      689   729      690  5004
691 10786      692  1641      695  8495      696 14468      697  5883
699  2734      700 11328      701  6882      704  2770      707  4136
708  6171      709  7269      710 16056      714 10028      715   255
720  2258      721 13298      722   790      723  2935      726  6742
727 15657      729 14097      730 14184      732 10893      733  6450
734  3608      736 11932      737  4296      740  9165      744   591
745 11476      747  4680      752 15850      754 12382      757 16619
758  2708      759  3180      760  6242      761 16672      762 16933
763  7056      764  2037      767  8636      769  8655      770  4619
771 10077      776  8477      777  7420      779  4776      781 15698
785  3647      788 13823      789  1603      790  1334      791  3617
792  5486      796 14376      797 16392      798  9529      799 13951
801 11537      803  3246      804  4660      806  7942      808  9237
810  8574      814  6424      819  3298      821  6964      825 13800
828 13771      829 14536      831  5453      832  1570      834 12039
835 14621      836 11694      837  4324      838 14167      840  3389
842  5846      843  7756      847  6199      851  1201      854  6994
855 10410      858 16272      860 14979      864  3943      866  9445
868  2251      869  6208      872  6212      874  6652      875 12125
878 13906      879 15743      880  6549      882  1526      884  1634
886 15470      889  5188      890 16038      892  1286      895   520
898  5230      899  6776      902  5563      903  7416      905  3327
906 14369      907  5409      908  1548      912 16599      913   825
917  5930      918  6832      921  2408      928 11955      929  7577
931  4441      932  7293      934 14158      936 14868      937  8230
938  3896      939  3879      941  8164      944 12938      948 14103
949 14535      952  6186      953  7169      954 11503      955  2928
956  5363      959 16476      960  7790      962   546      963  9544
966  4236      967 12957      968 11490      970  9992      971 10193
974  6401      980 12028      981  4965      983 10916      985 16537
986 15688      987 11939      991  6604      993 16522      996  3779
997  8823      998  5812      999 12776     1000  8320     1005 14897
1008  5061    1009 15553     1011 10058     1012  9368     1014  1003
1018 16529    1020  3764     1023  8548     1025  3047     1026  7048
1029 14326    1030  9550     1037  9988     1040 12230     1041 15427
1042 14776    1043 10139     1044 13548     1045  7254     1046  4039
1047  5802    1048 10118     1052 10881     1054 12330     1056 11554
```

Appendix CH4.5 Results of the message encryption/decryption program

```
1057 12519      1058  9696      1059 16883      1060  9078      1062 15218
1064  9397      1068 10592      1069 12988      1074  3920      1077  8118
1080  6792      1083  8227      1088  1306      1090 10537      1091  9414
1093 13125      1094 12445      1097 13740      1098 12162      1104  3299
1105 14757      1106  6660      1107 12251      1109  4587      1114  2704
1115  9956      1119 10271      1120 11378      1121  6691      1124 11804
1125 10697      1127  1230      1131  3793      1136  8630      1138 16645
1139  5025      1144  8573      1146   724      1148  3740      1151  7125
1153 10878      1154 13902      1155   110      1158  1290      1159  5364
1162  6467      1168  3792      1169  9397      1172  9579      1174  2490
1176 16785      1179  2223      1181   598      1182   319      1186  4300
1187  2182      1188  9208      1189  7874      1190  5001      1191 15651
1193 11556      1194 10609      1195 15269      1199  6634      1200   336
1204 15324  1204  15324  15324 A HIT
1206  9317      1207  4961      1209 10139      1211 16616      1214  9163
1215  5802      1220  6093      1224 16633      1227   544      1228  8912
1229 13233      1230  5534      1232 12521      1237  6058      1238  9833
1240  2260      1242 13516      1245  4527      1246 13822      1249  3939
1250  6028      1252  9047      1257  4783      1258 16051      1259  6948
1261 10324      1264  2037      1266  2107      1275  9893      1276  6045
1277  7710      1278  7561      1282 13230      1284  5357      1285 10830
1286 14634      1288  8445      1303 12238      1305 16771      1308  8022
1309 12221      1310  9642      1315   332      1321  4239      1322   724
1324  4545      1327 11751      1329  3802      1330 10691      1332  4888
1334  2440      1336 14718      1338 15264      1339   708      1340  1408
1341 10775      1342  6381      1343  3581      1346 13914      1347  1660
1349 15762      1353 16054      1354  9192      1356  8658      1361  1009
1363  6282      1364  5842      1366  6067      1370 15959      1372  9629
1373  7137      1375 16579      1376  7592      1381 16139      1382  7636
1383   890      1384  4795      1385 14204      1387  7031      1389 14367
1390 11008      1391 12737      1394  3530      1395  1873      1397 14184
1401  6039      1403  9136      1404  9696      1409  4181      1410  7280
1411  8073      1412  1274      1414  7667      1418  9258      1423  2866
1424 12546      1427 12262      1432  5651      1434 14191      1435 10099
1437  4052      1438  6610      1441 12846      1443  2551      1444  5167
1445  5057      1451  9743      1452  5083      1453 12433      1457  2224
1459 16658      1461   693      1463  8539      1468 10049      1469  8361
1473 15335      1475  7350      1476 16791      1479  4053      1480  1611
1481 11534      1483 11416      1484  6144      1485  9575      1489  6355
1490 14972  1491  387  1494 13071  1495  5549  1498 1641  1507 16417  1507  16417
16417 A HIT
1509  4862      1510 13380      1514 13193      1515 10031      1517  2462
1519  8714      1525  4099  1525  4099  4099 A HIT
1529  8000      1530 11752      1532 15911      1537  8718      1540   513
1543  2573      1544 15455      1547 16042      1548  3447      1550 12304
1553  3529      1554 11978      1555  2671      1559  1110      1561  1854
1564  1522      1566  1840      1568  7273      1570  4279      1571  4354
1572 12544      1575   676      1580  9625      1582  4420      1585  6530
1591 13658      1592 14474      1594 12170      1597 15286      1598  3962
1599  1042      1602  8919      1603  4294      1608  4734      1611  4055
1612  5651      1613 14778      1614  6742      1615  3123      1616  2229
1617  6470      1618 15451      1619   915  1619  915  915 A HIT
```

Appendix CH4.5 Results of the message encryption/decryption program

```
1620  3964      1621  8796      1622  4747      1623 11259      1624 11948
1626 15888      1631  4947      1636  5932      1639  5158      1643  6268
1644  5046      1645  5737      1647 15380      1649 12274      1650  1335
1652  7903      1654  3593      1656   525      1657  2874      1658 13973
1661 11076      1663 15140      1664  6761      1665  9136      1667  1949
1668  4863      1669  5610      1670 15863      1674  8445      1675  2395
1683 10044      1684   269      1685  2832      1688 13254      1690 14132
1691 12011      1692  4094      1693   205      1696 11363      1697  3731
1699  2224      1701  5841      1703 12787      1705  5391      1707  9785
1709  6343      1714  8984      1716  7737      1717  9224      1722  3245
1723 13646      1724  7710      1725  7665      1728 10716      1730 15139
1732  7891      1735   368      1739 16280      1741  1222      1743 16396
1744 12583      1745 10283      1746  2161      1748  2951      1750 14900
1751  7923      1754   409      1758 15639      1759  2443      1761  1522
1763  1127      1766  5726      1767  8071      1770  5528      1771  9793
1772 16541      1774  7299      1775  4450      1777 13736      1778  1998
1779  8868      1780  4488      1781  9162      1784 13949      1785  1048
1786  8744      1788  7459      1789  8512      1793  8658      1796 11691
1797 11800      1798 16641      1800  7791      1802  6954      1808  7040
1809 15918      1810  5914      1812   692      1816  6343  1816  6343  6343 A
HIT
1817  6190      1820 15117      1823 11037      1824 13868      1825  5105
1826  2417      1828  1936  1828  1936  1936 A HIT
1830  4677      1831 16363      1836 16474      1838  8612      1840 16668
1842  1513      1843  3839      1846   551      1848 10204      1851  7141
1853 15707      1855  7182      1858  5649      1862 16272      1865  8851
1867  6349      1868 10046      1869  1660      1870 15647      1872 10530
1873  3286      1875 10563      1877  1162      1878 16503      1879    33
1882 10736      1883 15152      1887   900      1888 12829      1892 11504
1893  1017      1895 11757      1896  3465      1899     6      1900  3785
1901 14937      1903 12337      1904 15926      1905  8871      1908  9813
1910 15698      1911  9530      1912  5013      1915 11555      1918  1600
1920 15730      1923  3172      1925 15040      1927  2155      1928  1946
1930  8579      1933 14296      1936  4251      1941 12017      1942  8579
1943  1731      1948  7511      1950  2860      1951  8489      1952 14906
1955 15880      1958 16310      1959  6031      1962 16204      1963  7782
1964  3091      1965   452      1969 10967      1970 13150      1971 11095
1972  9562      1973 12709      1974 11061      1976 12126      1980 15942
1982  3884      1983 14901      1985  4039      1987  3374      1989 12471
1991  6792      1993  1789      1994  1606      1995   324      1997  2395
2000 12988      2002  3856      2003  7548      2004  2378      2005  3120
2006  9629  2006  9629  9629 A HIT
2008 10345      2009  5194      2010  7577      2011 14756      2012 16373
2013 13119      2014  1873      2017 12525      2022 11807      2023 10149
2024 13538      2025 11658      2027 10736      2030 14872      2032  7644
2034  4713      2035   541      2036 15324      2041   981      2042  3128
2043  3082      2045 15633      2046  3662      2048  4280      2050  2712
2051 10237      2052 10151      2053  8223      2054 14355      2055  4187
2057  8292      2063 11315      2064 15043      2065 14253      2068  9809
2071  7416      2073 15233      2076 10926      2077   873      2080  9165
```

Appendix CH4.5 Results of the message encryption/decryption program

```
2082  3321      2086  5975      2088 12980      2091  8393      2095   841
2099  3407      2104  4846      2105  8907      2109 15967  2109  15967
15967 A HIT
2110  4325      2113 10358  2113  10358  10358 A HIT
2115   591      2121  1343      2122 13557      2123  7193      2124  9321
2125 14710      2126  8227      2127  2549      2128 11132      2129 16983
2130 14776      2131 13570      2132 14059      2134  3006      2135  1707
2143 10151      2145 15584      2148 14839      2150  4982      2152   909
2154  1013      2155  7908      2156 10023      2158  6523      2159 13657
2160  8483      2161 14458      2163  3324      2164 13813      2165  2950
2167 13502      2168  9063      2169 16380      2170 11960      2171 16926
2172  4102      2173 11489      2175 15004      2177  7910      2178  1965
2180 14567      2181  9539      2182 10442      2190 12022      2191 14059
2194  5785      2195  4714      2196  9709      2197 16641      2202  4385
2204   113      2205 14215  2205  14215  14215 A HIT
2206  3890      2209 16157      2210  5631      2211  6623      2218  2768
2221  3575      2222 10390      2223  8892      2225 15063      2226  2430
2229 16041      2230  8455      2232   658      2233  5372      2234  2860
2237 14418      2238   929      2240  4404      2241 16307      2242  4353
2245    45      2246 13780      2247 15495      2249  6721      2250  4092
2252  4632      2255  5042      2257  1952      2259 13107      2260 16374
2261 12030      2262 16198      2263 13114      2265  8832      2266  8386
2271 13301      2272  9687      2274 12315      2275 13754      2277  6053
2278 14778      2280  1306      2282  7273      2283 10643      2284 11283
2286  1429      2288  8980      2290 16348      2291  7955      2293  8424
2294 16093      2297  3333      2298 10354      2300  4131      2303  4807
2305 12472      2306 15061      2307 14181      2309 14796      2310   562
2311 15266      2312  4514      2314 12327  2314  12327  12327 A HIT
2316 14738      2324 12438      2328 10311      2330  7634      2338  5064
2341 11784      2343 12817      2346 16837      2347 11989      2348  8395
2349  2208      2353 11289      2355 15371      2357  8395      2360 13942
2361  9100      2362 10929      2366 13507      2367 15692      2368  7402
2369  6125      2370  8220      2373  1731      2374  6861      2375  3094
2376 12217      2379 16223      2380  7009      2382  5266      2383  1280
2389  6315      2390 15618      2394  6643      2396  1077      2399 14161
2403 13800      2404  4407      2405 15167      2407 16157      2408 11982
2409  7728      2410 12270      2412 14684      2415  9597      2417  9721
2419 16780      2420 16012      2421  7123      2423  6204      2425   307
2426  8194      2427  2596      2429 10916      2430 16944      2431  5630
2434  5319      2436 16904      2437  1120      2439 13334      2441 10275
2443  7495      2444  6201      2446 11676      2447 10878      2448 10961
2451 12442      2453  4120      2454  4229      2459 12429      2460 14648
2462 10400      2463  8729      2464  6721      2465 10719      2468  9549
2470  5389      2472  2926      2481  2342      2484  6053      2486 10197
2487  5576      2488  2780      2490  6750      2491  9588      2492  1276
2495 10203      2496 12023      2500 15035      2501 12640      2502  5948
2503   967      2507  1086      2508  1404      2512 13557      2513 15821
2515  8393      2516 11628      2519 11929      2520 16289      2521  8487
2523 13116      2526  4459      2528  4662      2529  7905      2533 15761
2536 14894      2539 13076      2541  9512      2545  5625      2549 13660
2551 12851      2553 13663      2554  6645      2555  1537      2557 13876
2558  2618      2561  6478      2564  1762      2567  1434      2568 10224
2573 16154      2574  4609      2575  6878      2576  7993      2577 16636
```

Appendix CH4.5 Results of the message encryption/decryption program

```
2578 16723      2579   190      2580 16933      2581 16657      2583  3310
2584 11762      2586  5151      2587 14499      2588  1256      2589 12845
2590 10164      2591 15194      2593  7346      2594  6878      2596 11060
2600 15150      2601 13914      2603   333      2604  9498      2605 15111
2607 15809      2611 10196      2613 10123      2616  4977      2617  8403
2619 11933      2628  2950  2628  2950  2950 A HIT
2630  7659      2631  2534      2632 15508      2633  8131      2635 13400
2640  3661      2641  3323      2643 14962      2644 12255      2645  3712
2646    57      2649 11471      2650  4866      2651  4725      2652 15333
2654 10120      2655  9570      2657  2586      2659  7540      2660 15033
2662  2161      2663  9687      2666 13130      2667  5749      2671  5349
2680 14017      2681 10040      2683  2422      2685  5546      2691  4195
2692  4433      2695 15716      2697  1605      2700  8581      2701  1661
2702 11279  2702  11279  11279 A HIT
2704   393  2704  393  393 A HIT
2705 13887      2708 11651  2708  11651  11651 A HIT
2709 15436      2710   401  2710  401  401 A HIT
2713  2561      2714  6832      2715 13753  2715  13753  13753 A HIT
2716 15860      2718  3716  2718  3716  3716 A HIT
2720  9209      2721  6750      2724  4631      2726 12419      2727  3207
2729  7540      2730 11811      2731  8749      2732  1889      2733  6370
2734  2269      2737   322      2740 12839      2742  5999      2745  7007
2747  7489      2748 12954      2749 11283      2750   368      2751  5184
2754  2243      2755 13825      2756 14321      2758 12058      2760 12867
2762  1181      2764  9048      2768   929      2769 15140      2771 11737
2774 11429      2775  8377      2777 10500      2784 13468      2785 13120
2787   352      2789  7206      2791 14002      2792  2616      2794 14042
2795 15646      2797  9097      2798 12038      2799  1089      2805  1905
2806  8539      2808 16608      2810  5563      2814 15228      2816  4571
2819 16495      2821 10154      2823  3839      2830 10100      2831  7905
2833  5085      2835 13789      2836 13112      2838  2694      2839  6591
2841  6247      2842  9794      2845 10211      2846  5435      2849   606
2852 10904      2855  5574      2856  1737      2857  1133      2858  1121
2860 14002      2862    56      2863 10999      2866  4754      2867  6994
2868  5800      2869  9587      2871 14378      2873    54      2874  5405
2877 14544      2879 14851      2881  1456      2882  7201      2883 10951
2884   871      2886  5479      2892 14391      2896 10333      2897 14245
2902 14618      2903   814      2904  4356      2906  7050      2907  6902
2909 14979      2910    51      2911  5670      2912  9444      2916  7018
2917 10671      2918 13048      2919 16471      2920   189      2921 14081
2922  2481      2923  9569      2924 15243      2925  5486      2926  6613
2927  1841      2928  1801      2929  1537      2931 10028      2932  9076
2933  5357      2934   946      2940 16963      2949  7728      2951 10684
2952   416      2954 14846      2955  7626      2959  6479      2960  3977
2962  9808      2967  1291      2968 11234      2969  1032      2973 13302
2978  1472      2979 14838      2980 11030      2983  9293      2985  8385
2986 16212      2988   503      2990  7762      2992 14356      2995  2744
2996   149      2998  5238      3001 11024      3002 15799      3003  8424
3005  5190      3006  7361      3007  4330      3008 14469      3011  9091
3012 15518      3014  6192      3016   692      3018  6208      3019  5489
4786  8512      4787 16414      4789  6087      4790 12500      4793 14302
4794 10005      4797   302      4799  9597      4800 12427      4804  7706
4805 16210      4807 10159      4810  3238      4812 16494      4816  6421
4818  6271      4819  2997      4821  4723      4822 11962      4826 10342
4828   760      4829  9879      4830  3772      4833  2251      4834  7702
4837  2522      4838 12512      4839 15904      4840   333      4841 12695
```

Appendix CH4.5 Results of the message encryption/decryption program

```
4844 16382      4846 11389      4847  2836      4848  3745      4853  8864
4854 13821      4856  6449      4857  9537      4859  7997      4860  3255
4862  9580      4864 14148      4866 13951      4868 13093      4870 11935
4871   132      4872  7392      4874  5906      4878    56      4879 10685
4880  7589      4883  1232      4884 12340      4888  1532      4889  1159
4892  1154      4893  7776      4895  1714      4896 14657      4897 13107
4899 12677      4900  1031      4901  3170      4902 13833      4903  2326
4904  2328      4905  5016      4907  4888      4908  7231      4910  1781
4912 12114      4913  4663      4917  7772      4918  9893      4920  5746
4924  5001      4927 14090      4936  5611      4937  4041      4940  6716
4942 15500      4943  7206      4944 13017      4945  8972      4949  4554
4954 16376      4958  5795      4959  2424      4962 16245      4964 12229
4965 16791      4967 16597      4968 16414      4969 13862      4971 13428
4972  6574      4974  7573      4975 10173      4977 11768      4978  4657
4979 14775      4980 13144      4981  7708      4982   962      4983   201
4985 14404      4986 12146      4989  9535      4991  7215      4993 13090
4995  1470      4998  5828      4999  3030      5009  3529      5010 15445
5011   811      5012 10271      5014 14454      5015  5194      5016  7256
5017 13863      5021  1157      5022 15944      5024  6963      5026  7994
5030  7003      5033  5577      5034  8869      5037  5574      5038 13244
7504  8835      7505  5759      7507 12807      7508  4308      7511 15847
7512  8827      7513  3608      7514 15223      7516 14321      7517  5734
7518  6733      7522 14266      7525  7374      7528  2660      7529  4740
7531  8332      7532 14757      7535   484      7537  9849      7539  9325
7540  6214      7541 10464      7542   282      7543 16578      7544 13161
7549 15547      7550  9884      7551 16054      7553 12798      7555  1820
7559 16237      7566 11778      7569 11917      7570 15897      7571 12579
7573 10438      7574 15155      7576  9443      7578  7078      7579 16998
7582  6187      7584  4305      7585   757      7587 13104      7590 13149
7592  8351      7593 15646      7595 12593      7605  6924      7606  8520
7607  8003      7608 15470      7611  8905      7613  4514      7618  4477
7620 12085      7622  4330      7623 11964      7625  4905      7628  5534
7629  5304      7630  4729      7633  1221      7639 13877      7640  2209
7641 11548      7646 13589      7647  2164      7648 16210      7649  6330
7651 15231      7658  9185      7659  6255      7660 11039      7661  8319
7662  5600      7663  9005      7665  2586      7667 15706      7668  3900
7669  4134      7670  4177      7671 14755      7672  3197      7673  3059
7678 14758      7680 11054      7681  8839      7682 16026      7685 14562
7687 15951      7689   864      7693 16929      7694  6522      7698 10416
7699  6199      7700 15923      7701   930      7703 16202      7704  8400
7705  5506      7708  1904      7709  9243      7717  9125      7718 13477
7721 12246      7724  7763      7725  3119      7726  2427      7728  8108
7729  6401      7730 10629      7733  1679      7734 16278      7737 12019
7738  1952      7739  5806      7740  2562      7742 15246      7745 10813
7746 12599      7747 13140      7748 10138      7749  7592      7750   206
7751 14854      7752 13600      7753  6271      7755 13242      7759 12993
8987 12669      8988  5330      8991 13589      8992  3125      8994 16723
8995  3224      8997 15636      8998  3975      9000  8000      9007 12051
9013  2732      9014  9386      9015 16566      9016  4251      9017  1620
9019  3659      9020 12669      9021  4525      9025  2630      9027  1194
9029  1698      9030  1757      9031 13614      9033 11904      9036 15645
9038 15908      9039 16560      9041  4180      9042  6564      9045 13976
9047  4101      9049  2422      9050  6322      9051  2632      9053  3115
9055  5315      9057 12376      9058    64      9059 16350      9061 14510
9063  2694      9065  4624      9067  8173      9070  7042      9071 12606
9073  5983      9076  5106      9078  4345      9080  5175      9081  5751
```

Appendix CH4.5 Results of the message encryption/decryption program

9082 14880	9084 2889	9086 431	9088 792	9091 5418
9092 11481	9093 16488	9094 6087	9095 1726	9098 4997
9099 5549	9100 9743	9103 506	9104 16311	9106 14479
9107 1737	9111 14316	9112 13707	9113 12546	9116 12255
9117 16425	9118 16084	9119 5574	9122 1105	9123 10313
9125 2297	9127 5302	9129 5605	9131 8423	9132 2263
9133 13154	9134 3278	9135 12794	9136 6283	9137 7459
9139 6428	9141 16447	9144 15167	9145 5660	9146 16632
9148 6936	9149 3532	9150 3838	9152 16569	9153 2725
9154 10062	9155 13114	9156 16448	9157 2066	9158 10318
9162 9443	9164 2496	9166 15896	9167 5441	9171 4095
9172 12275	9175 11533	9177 10795	9179 5493	9180 15099
9183 10214	9184 16538	9185 14947	9186 13301	9189 16760
9190 15532	9191 15125	9192 101	9193 6231	9195 7595
9196 7543	9199 836	9200 10929	9201 11982	9202 6834
9203 11660	9204 5045	9205 4714	9206 14673	9207 15521
9209 12618	9210 6353	9217 4052	9218 2171	9219 10629
9220 13877	9222 732	9228 16652	9232 2290	9233 6187
9235 9623	9239 4922	9240 7111	9243 678	9244 5163
9247 6196	9248 6634	9250 9920	9251 4077	9252 381
11124 2936	11126 10578	11128 5242	11131 10852	11132 16501
11133 2822	11134 1595	11136 6429	11138 1476	11139 7922
11141 9630	11142 14948	11143 3517	11144 9225	11145 6726
11146 13930	11148 5880	11149 6836	11152 10265	11154 1949
11155 16190	11159 8753	11160 2983	11166 12817	11167 1600
11170 1382	11171 11471	11172 9705	11174 7166	11176 12004
11177 12984	11178 6723	11180 8330	11181 5431	11184 8912
11185 14952	11190 7123	11193 6416	11195 2205	11196 16311
11197 626	11198 5384	11199 181	11200 13202	11201 9409
11203 1465	11205 80	11207 8431	11208 4423	11211 6261
11212 13140	11214 26	11215 5608	11216 15187	11218 8111
11219 221	11220 1501	11221 10325	11224 14134	11227 12555
11228 2265	11229 740	11230 4	11233 6455	11235 1876
11236 6143	11239 11134	11240 10939	11242 15587	11243 2824
11245 484	11247 3492	11249 6285	11253 4056	11255 16604
11257 2755	11258 3567	11259 8018	11261 4378	11263 9316
11264 16949	11265 11490	11274 9276	11275 10158	11276 2455
11277 3101	11281 11150	11288 6779	11289 13334	11290 2806
11291 9514	11295 8111	11297 11301	11299 16708	11310 2251
11314 80	11315 4859	11316 15711	11317 13010	11320 7179
11323 13716	11327 14648	11328 2730	11330 5658	11331 10674
11333 16652	11338 15358	11341 1256	11342 16326	11343 9122
11345 12538	11346 13951	11347 4159	11349 1833	11353 12322
11357 124	11358 9783	11359 109	11360 14002	11362 3173
11363 8783	11365 16673	11366 10984	11367 4419	11369 964
11371 7739	11374 13650	11376 10578	11377 14337	11378 9414
11379 8323	11381 5943	11384 12820	11386 10076	11387 14066
11388 2459	11390 12590	11394 5024	11396 12737	11397 2988
11398 2600	11399 7422	11400 11511	11401 14209	11402 16373
11403 6042	11404 8662	11405 9612	11406 10530	11407 15939
11412 5604	11414 4824	11415 16408	11416 5714	11418 16131
11421 2618	11423 10517	11424 3004	11427 11523	11428 11510
11431 1489	11433 14258	11435 1011	11437 11417	11438 3334
11439 1120	11442 13853	11444 1350	11445 993	11447 14198
11448 8635	11450 13791	11451 3065	11454 11571	11461 16811
11462 7483	11463 9785	11464 5116	11466 15902	11469 3031
11470 16671	11471 15074	11472 8573	11474 9821	11475 16672

Appendix CH4.5 Results of the message encryption/decryption program

```
11479  8366     11482 10706     11483  5148     11488  4914     11494 11173
11496  1944     11498 12559     11499  4324     11501 16288     11503    80
11505  3729     11507  5857     11509  8111     11510  1194     11511  2898
11512  1236     11513  3312     11515 15865     11516 16593     11520 10578
11521 14761     11526 16103     11527  1537     11528 13346     11532  3749
11534 13365     11536  2323     11537 14153     11543 12672     11548  5323
11549  1821     11550  7059     11551  2507     11552  4464     11554 16371
11555  1284     11557  3749     11558 14734     11560  8023     11561 12676
11565 13420     11566 16269     11567  3906     11568  8490     11569  6353
11572 14550     11573  1251     11574  2864     11576 11272     11578 11583
11583 11932     11585 16382     11586  9445     11588  6883     11591 11278
11592 16534     11595 12019     11596  2186     11600  2995     11603 10297
11605  1118     11607 10628     11608  3756     11609 11291     11610  5579
11612  5066     11614  2539     11616  6244     11617  3024     11618  8933
11620 15563     11623  7540     11624   757     11625  1602     11628  5170
11631 11215     11632  6153     11634  8451     11635 15507     11637  2217
11638 11188     11640  1489     11641  4522     11643  1558     11646  9501
11648  4744     11650  7492     11651  7905     11652  2260     11653  7728
11655 16380     11657    96     11659 11392     11660   994     11662 10358
11665  2697     11666 16407     11668  1470     11669 13344     11670  6023
11672  5185     11673  7121     11675  4793     11676   714     11677  9310
11678  4744     11682   883     11684 10614     11686  2296     11687 16909
11691 13197     11693  3896     11695 12069     11699 14843     11705 16860
11709  7842     11710  1929     11712  5251     11713  8676     11714 14066
11715  8424     11716  6211     11718 15077     11719  2869     11720   381
11722  4136     11725  6733     11726  1009     11727  5115     11729  7874
11730  3040     11731 16283     11732   991     11733 13771     11736 13200
11739 13743     11741  7504     11744 10024     11745  3755     11746 15730
11748  4117     11749  2544     11750  9102     11751  2505     11753  7438
11756  9956     11757 12403     11758  2146     11762  7082     11764 13347
11765  2209     11766  2453     11769  8804     11771  5911     11773  6386
11775  4552     11782   185     11783   680     11786 10967     11790  7347
11791 12033     11793  3706     11799  3569     11800  6750     11802 15427
11803  6535     11806  9903     11807  9907     11808 10609     11809  1236
11810  3939     11812  4726     11815  6183     11817  8540     11818  7265
11821 15180     11822 13734     11823  1876     11824  2936     11825 10396
11828 13985     11829  1186     11830  5906     11831  3431     11833   874
11834 13998     11835 12356     11838 12957     11839  6435     11841  6033
11842  6878     11844  5885     11848 16352     11850  7967     11853  4777
11854 11178     11857 14756     11858  2443     11860 11883     11861 14429
11862 16203     11863 12128     11864 16344     11868 12048     11870  3296
11874  5759     11878  7657     11880 16851     11881  7569     11885 11990
11886  6351     11887  1254     11888  8170     11889  6059     11896   961
11901  6720     11903 16546     11904  3226     11906  3100     11908  7278
11911  4159     11914  7347     11916  2386     11919 14427     11920  3920
12515   902     12516 10702     12517 11488     12518 11719     12521  1884
12526   368     12531 10486     12533 10054     12535 11664     12536  5573
12537 13279     12538  2135     12539 12690     12544  4651     12545 16181
12546  4451     12550 16473     12551  8455     12553  5348     12555 15326
12556  4068     12558  7403     12559  8003     12560  4619     12561 16942
12562  4997     12566  8763     12568  3018     12569 12492     12570 10242
12572 13292     12575 16188     12576 12188     12577  3028     12579 15140
12581 13056     12584 15662     12585 13913     12586  6301     12589  1205
12590  3645     12591  6567     12592 14800     12593 11354     12598  9407
12599 16026     12601   974     12603  2161     12609  4507     12610 16918
```

Appendix CH4.5 Results of the message encryption/decryption program

```
12612 10277     12614  6311     12615  5119     12616  7926     12617  4037
12620  7082     12622  6117     12626  1530     12628  5550     12630  7005
12632  1492     12636  2995     12638 12901     12639 11451     12640  1289
12644  8405     12646 15978     12649  2175     12650 14479     12651 11846
12653  3459     12654  4708     12656  1054     12663  6296     12665 10419
12667  4711     12669  3385     12671  3976     12674   105     12676  2341
12677  5443     12680  6464     12681  4003     12688 14059     12690 14800
12691  5482     12693 13224     12694  6708     12695  1731     12697  5422
12698   828     12699  8657     12706  1172     12707 11192     12708  4871
12709  4501     12710 15742     12711  4952     12716 16028     12717  8794
12719   215     12720 13045     12724  5028     12726  2222     12729  9491
12732 14582     12736  6883     12738 11224     12739 14944     12741  7169
12742  8539     12743  2016     12744 15571     12745  7977     12747  8130
12748  7779     12749 14629     12750   901     12754 15759     12756  8780
12760  5028     12761 12535     12765  9778     12767 10676     12768  1740
12769  4113     12771  5911     12774  5632     12775 13797     12777  1370
12778  7573     12779 11503     12781  4968     12784 10892     12786 16366
12787  7888     12788   678     12791  7430     12792   782     12794  5357
12795 13314     12796 16400     12798  5924     12799  1664     12800 16685
12806 14950     12807 10588     12810 15687     12812 14375     12816 10151
12820   744     12826 15903     12827  2860     12829 11902     12830  1846
12831   181     12838 16655     12841 16480     12845  4068     12848  7374
12849  1752     12850 16632     12851 11955     12852  8363     12858 13144
12859  2726     12860  2186     12861  3218     12863  3343     12864  2473
12867  6679     12868 10267     12869 14553     12870 10848     12872  9111
12873  3879     12874 11215     12880  5658     12881 10917     12882 13859
12883 16578     12888  9729     12890  6408     12892  8015     12894  4069
12895   825     12897  8159     12899 12984     12900 12538     12902 12793
12905 11887     12907  6372     12911 15115     12912 15438     12916  5281
12917  7319     12919 11617     12920 12573     12921 11059     12922  5714
12924 12330     12927  5543     12929 16107     12930  3936     12931  6372
12932   104     12933 10753     12934  7287     12935  1333     12936 12904
12938  5096     12941  8773     12947  4290     12948  4088     12951 11519
12954 15751     12955 12690     12957  6428     12960  2679     12961 15207
12963  9639     12966  1311     12967 11939     12968  5784     12969  6665
12971  5729     12972  5104     12974 17000     12977  5386     12978 14833
12980  7572     12984 13383     12985  5614     12988 13864     12989  8990
12993 14437     12996  4945     12997  4507     13001  9501     13002 10900
13003  1410     13005 12114     13006 16490     13016 16641     13020 15909
13022  9139     13023 11080     13024  8446     13025  7706     13030 15750
13033  7618     13035  8264     13036  9848     13038  1369     13040  4969
13043  5161     13044  3579     13048 14398     13050   381     13052  9825
13053  2348     13054  3743     13056 10580     13057 12199     13060  3717
13062  1444     13063 12278     13064 13914     13065  6242     13066  7629
13067  9877     13068  6679     13069  8413     13074 13645     13075  7249
13079 16858     13081  7106     13083 11072     13085  8109     13086  3205
13087 14119     13089  5024     13090  1784     13096  3325     13099  1383
13100  2248     13101  6398     13102 10736     13106 12182     13107  6217
13109 12729     13110 13365     13111 13162     13117  4553     13120 14778
13122  4346     13124  6208     13125 15828     13127  2360     13134  6697
13138  8359     13139  8579     13140  2904     13141  3903     13143 10059
13144 12236     13146  2439     13150  9358     13151 15037     13154  1553
13157  4935     13159 16135     13161  7273     13162  7835     13164  2348
13165  8162     13166  6739     13168 15785     13169  1533     13171 16249
13175 14733     13176 14950     13179  3733     13182 11501     13185  6972
13186  7664     13190 12491     13191 16433     13192  2133     13196 13815
```

Appendix CH4.5 Results of the message encryption/decryption program

```
13199 10508     13201 14323     13202 12559     13204  4435     13205 10777
13206  7526     13207 14849     13211 15020     13214 14370     13215 14067
13218  1935     13227 14952     13228  5471     13240  5883     13242 14979
13243  6353     13244 13847     13245 13982     13246 11453     13248 16268
13249 15177     13250  9568     13253  6529     13254   130     13255  7947
13262  5688     13263 11708     13265  5888     13266  2097     13267  2082
13273 11410     13277  7887     13278  4366     13279  3157     13281  2856
13282 14964     13285  3621     13286 16918     13287 10356     13289 14791
13290  6994     13291  9403     13294  5486     13296 14108     13299  5563
13300   505     13304  6458     13305  2628     13307 13723     13313  5384
13321  3800     13327 15795     13328 11994     13329  8555     13330  1927
13338 15434     13339  2395     13340  1084     13343 15209     13344 10357
13345  5961     13346  2289     13347  1711     13349 12160     13350  3393
13351  4896     13355  8385     13356 16422     13357 14208     13359 15829
13362  6863     13363 11687     13364  8020     13366 10028     13368 11375
13369 14416     13371   453     13372 16817     13374  6782     13376 16364
13379  6832     13380 13768     13381 14884     13384  9702     13386  6311
13387 11391     13391  9851     13393 15946     13394 12986     13396  2128
13402  3835     13406  9022     13409 15594     13411 10878     13413  5731
13414  8480     13418 16091     13419 10053     13420  6905     13424  5845
13425  3356     13429 10693     13431   504     13434  7278     13436 14166
13438  4121     13439 12202     13440  6426     13441  6005     13442 16029
13444  5831     13445  7312     13446  2204     13448 12072     13450  5685
13451 15051     13454    23     13459  5350     13460  1708     13461 16652
13462  4493     13463 11402     13464  8254     13467  6563     13468 10995
13470 13947     13474   185     13475 13048     13476 11480     13479  9753
13484  9923     13485  1194     13486  6343     13488 14967     13489  1740
13490 10955     13491  3744     13492  9242     13495 11968     13496   400
13497 15031     13498  4261     13500 12178     13502 15234     13505 13985
13506 15766     13510  8557     13512 16418     13514  2647     13515   886
13519  7601     13520 14052     13521  5607     13523 12902     13524  9932
13525 16858     13528 12884     13530  7524     13531 16747     13532  4494
13533 12032     13534  2296     13535  3992     13541 13789     13543  9784
13549 11645     13550 11678     13551 14033     13552 14814     13553  5066
13554   777     13556  7012     13558  1237     13561 12811     13563 16921
13564  9782     13572  6812     13573  9242     13580  3256     13582 11408
13585 13258     13587   608     13588 11735     13589 14068     13591 14655
13592 13179     13594 14480     13596 16578     13597   860     13598   624
13599 10916     13602  1873     13603  7473     13610  1400     13612  1021
13618  7709     13619 16884     13621  9127     13622  3181     13623  4249
13625  2705     13626   866     13627  2785     13629 10060     13633  9629
13635  1027     13636  5350     13637 14869     13639  2072     13640  3929
13641  4766     13642  6821     13643  1306     13645 14989     13646 15161
13647 13315     13648  3031     13649  7374     13650 12185     13651 12836
13654 16739     13658   571     13659  8345     13663 16582     13664  6233
13665   887     13666  1395     13669 16933     13670  2167     13672  6697
13675  9233     13676 14469     13678 11206     13680 16967     13681  6334
13682  7409     13683  4598     13685  9953     13686  1522     13687  3412
14853   841     14854 13511     14855  5922     14857  9977     14859  5518
14860  5713     14861  7796     14862  9392     14864  8910     14872  1574
14877  5380     14878 10734     14879  6151     14880  6514     14881  9042
14882 15172     14884 14184     14885  2749     14886  9613     14887  4969
14889  3736     14890 13443     14891  5568     14900 14319     14903 14253
14905 11040     14907 16909     14908 15612     14910  6400     14911 13976
14912 11704     14914  1893     14915  9224     14916  6523     14917 13361
```

Appendix CH4.5 Results of the message encryption/decryption program

```
15069 16994     15072 12614     15073  3605     15079 14071     15080  2573
15081 16994     15082 13734     15083  4929     15086  1483     15088  3580
15090 13586     15091  1395     15092 15181     15093  8109     15100  5729
15101  4195     15102  4128     15106  3312     15107  3007     15109 16209
15111  1836     15112 14567     15115 10042     15121   270     15122   979
15125  7310     15126 11738     15128  9934     15131 15414     15132 11069
15134  4617     15135  7899     15139  4366     15140  6149     15141  8946
15142  1735     15144 11313     15145  4717     15146  7661     15148  6153
15149  9209     15154  2864     15160  3949     15164 14996     15167  4640
15168 12169     15170  1242     15171  3912     15173  5975     15175  9495
15176  8972     15177 10665     15178  6040     15179  5085     15180  4692
15181  5770     15182  9659     15186 11500     15190  3379     15197 12268
15199   760     15201 14769     15204 15283     15209 13844     15212 16685
15213  4787     15216  9580     15218 14320     15219 10237     15221  9431
15225 16101     15227 12446     15230    77     15231 13629     15235 16541
15236  1402     15241 16176     15242   484     15243  6942     15246 10518
15247 15177     15248 16932     15249  2760     15250 11575     15251  4239
15253  7757     15254  8630     15255  3744     15256 16747     15258 13314
15259  4281     15263  5379     15265 15599     15266 16653     15268  2657
15269 13017     15270  6541     15271  8118     15272  5447     15273 16149
15274  9222     15275  9861     15276  6796     15279  6542     15280 12521
15282  9587     15283  3457     15284 14195     15285  8537     15286  7457
15287 11024     15288 14414     15292 15680     15293  1077     15295 12679
15296 14378     15300  9951     15301 10399     15305 17000     15306  5865
15307  7626     15308  7784     15309  3120     15314  9299     15315    49
15318  1726     15320  4168     15324  6691     15325  8582     15326 10342
15327  8646     15328 13449     15329  5573     15331  6373     15336 13141
15337  9808     15339  3772     15343  6105     15345 12356     15346  2556
15347  1863     15348  6635     15349 12509     15350  3887     15352  9370
15354  7742     15356 16605     15366  9185     15367 14562     15369  4255
15372 13258     15374 10557     15376  3128     15377  5800     15378  3082
15380  6407     15381 11962     15382  4976     15383 14704     15384 10408
15385  7114     15386   721     15387  4222     15390   877     15392  3197
15393  9578     15394  8837     15398  3546     15399  2854     15401  6386
15402  6902     15404 16534     15405 15815     15406  9039     15409  8507
15410 10813     15411  9579     15412 13625     15414 11988     15415  5576
16000  9188     16002   988     16003  4336     16004  4312     16005   836
16010  5085     16015 12669     16016  8330     16017  9399     16018  5092
16021 12838     16022  8406     16026 13106     16027 11453     16028  7038
16032 13853     16037  7344     16041 12550     16042 15325     16043 10299
16048  9521     16049 11536     16051  4033     16053  2525     16054  9892
16055  2643     16056  7638     16057 12577     16060  5406     16061  6158
16063 12410     16064 15870     16066  5148     16067 13287     16068 13315
16071  1837     16075 13487     16077  7100     16081 10577     16083  6356
16086  2175     16087  1572     16090  7267     16092  3791     16093  5907
16094  8276     16095 12365     16097 14182     16100 14451     16105 12114
16107  7174     16109 10746     16111  6741     16112 10527     16114  8842
16115 14884     16119   163     16120  5576     16122  5214     16126 12211
16134  7698     16135 14041     16136 11930     16138 11666     16139 13984
16140  6696     16141 10319     16142 14628     16143 10226     16149 12439
16150  4769     16151  3331     16153  9209     16154 13376     16156  2719
16157 16059     16162 10127     16163 12151     16164 13497     16165  1633
16166 15155     16167 11061     16169   431     16170 11784     16172 16496
16175 13681     16176  8655     16183 16942     16184 13144     16187 13491
```

Appendix CH4.5 Results of the message encryption/decryption program

```
16188 11173     16189 16447     16190  4304     16192  6143     16193  7706
16194 16383     16195  9998     16197  6644     16198 11008     16199  6232
16200  3112     16201  6355     16203  8048     16206 14527     16207  3203
16209 12724     16211  9032     16212  5882     16213 16743     16215  4549
16225  5482     16227  2653     16230  4195     16233 14880     16236  3648
16240 16499     16241  1927     16242 14621     16243   740     16246  1779
16249  6482     16253 13347     16254 11267     16256 14593     16257 16343
16262  8299     16263 11102     16265  5850     16268 14771     16269   624
16270  7573     16272  3790     16274  3547     16275  7101     16276  9912
16278 15709     16279  9514     16281  2549     16282  4705     16283 14398
16284  6185     16288  7970     16289 13296     16290  1881     16294    37
16295  8294     16296  6737     16297  4753     16298  5963     16302  3579
16304  4020     16305 13937     16307  7746     16308 16214     16312  2226
16313 16921     16316   383     16317   306     16319 12022     16320  1763
16322  9420     16323 14026     16330  6247     16332  6355     16333  1077
16334  9694     16336  2091     16337 16119     16338  9138     16341 11942
16343  3683     16347  1296     16352  2938     16354 15621     16356  5579
16358 15699     16360 12609     16361 11800     16363 11022     16366  4385
16368  1229     16369  9179     16370  3393     16372  3536     16374  5004
16376 12131     16377  4628     16378 16581     16379  4096     16382 12950
16384  8907     16387 10112     16388 12864     16391 13650     16392 16497
16393 14192     16398 10313     16399 11149     16400  1698     16401 11009
16403   121     16407  7939     16408 11834     16414  5702     16417  3629
16419 13616     16422  5224     16425 13683     16427 11002     16428 13657
16430  2954     16433 14520     16434 11008     16437 13916     16439  1230
16440 11278     16447  9773     16450  1007     16452  3669     16455  9811
16456  1230     16459 14311     16460  3588     16461 16062     16462 12012
16463  6591     16464  3929     16466 13797     16476 11722     16481  5681
16485 11350     16487  1021     16488 13788     16492  8030     16493  2616
16496  3657     16499   138     16500 11548     16503 16301     16504 15532
16505  7230     16506 15860     16508  2754     16509  3986     16510  6378
16511 12741     16513  8743     16515  1510     16517  8943     16521  8601
16523   842     16524 13509     16525  7622     16526 10968     16527  4384
16531 13369     16532 11616     16534  8701     16535 16397     16536 15633
16541  3270     16542 12373     16547  8677     16549 14632     16550  3076
16552  8002     16553  2809     16558  9924     16559 16501     16561  6919
16562  7232     16563 14275     16564 13609     16566 11059     16567  6724
16569  8008     16570  1911     16571 11775     16576  2530     16578  9630
16580  2033     16581 13714     16584 14417     16585 16418     16587  3279
16588 10552     16590 16308     16592  3248     16594  8636     16596  3938
16598  2971     16600   620     16601  3082     16602  6555     16603 13409
16604  3128     16609  9366     16610 11665     16612 15497     16613  8067
16614  7346     16616 16536     16617  9536     16620 10694     16622 10474
16624 10131     16625  4177     16626  6196     16627 12794     16628  2939
16629  6602     16631 13960     16637 13531     16642 12874     16643  1679
16644  4088     16645  8034     16646  2102     16648  6228     16651 14495
16652   471     16653  4345     16657 12809     16658  7957     16659  9111
16660  6243     16661  8636     16662  3681     16664 12005     16665 14252
16666  3442     16667 11306     16668 14240     16669  2366     16670  2963
16674  2724     16675  2732     16676  5659     16678  7430     16680 13020
16682  8868     16686 12642     16688  1823     16690 14926     16691  8586
16693  1165     16695 15074     16698  6827     16699  4691     16700  5810
16701  6610     16702  6112     16705  3003     16708  3120     16711  5344
16712  8807     16713 14949     16714 13506     16715 15305     16716 14655
16718   574     16719  7685     16723 16225     16726 16733     16727  9877
16728  9243     16730 14567     16731  5189     16733 10277     16735 13998
16738  2948     16740  2983     16744  1023     16746 14813     16747  8771
```

Appendix CH4.5 Results of the message encryption/decryption program

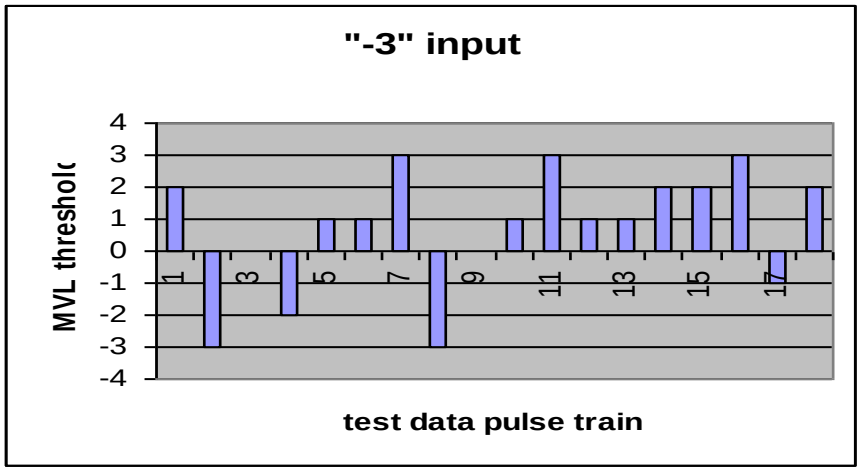
```
16748 16143     16750  1133     16751  9181     16752 10237     16753 13336
16754 16564     16758  5669     16760 13905     16763  5975     16764  7017
16767  6030     16768  2933     16769 13174     16770  4227     16772  1410
16773  2216     16775  3192     16776 14916     16778  8386     16779 15431
16784  7131     16785  3269     16787  6268     16791 15911     16792  3380
16794  6233     16795   573     16798 12868     16799 11644     16800  2866
16803  1816     16804 14888     16808  6447     16809 14658     16810  7429
16811  9945     16812  8728     16813 14786     16815 10146     16819  1061
16821  4859     16823 13649     16824 11134     16825 10982     16828 10430
16830 13760     16832  7348     16833 12454     16835  6516     16837 13506
16841  6093     16842  2737     16843 11678     16846 12822     16853 15696
16854  5970     16856  3349     16859 15443     16860  4651     16861  8763
16863    23     16865 13960     16868  1887     16869 15239     16871  4356
16873 14510     16874  1727     16875  5806     16876  3373     16879  4417
16880  1089     16881  2434     16884 14295     16885  2673     16893  1459
16897  8298     16898  9848     16899 13154     16901 16112     16902  3645
16904  8816     16905 12573     16908   371     16911  9937     16912 12809
16913  5161     16918 10323     16919 13887     16920    43     16922  9693
16930  8034     16933  8561     16935  3329     16936  7524     16938  5608
16939 10327     16943 14764     16945 14621     16946   126     16948  6523
16953  4931     16954  3916     16955   816     16958  5004     16960 10592
16961  1553     16967  3442     16976  2660     16979  9335     16980 11794
16982   772     16983 11642     16985  2714     16987 16091     16989 12166
16990   792     16996  3300     17001  7136     17003 16350     17004 14102
17005 15111     17008   620     17010 14564

   TOTAL NUMBER OF HITS= 22 AND THE ORDER OF THE GROUP IS = 0
   1    1507 16417       1507 16417A HIT
   2    2704   393       2704   393A HIT
   3    2628  2950       2628  2950A HIT
   4    1204 15324       1204 15324A HIT
   5     528 12647        528 12647A HIT
   6    2006  9629       2006  9629A HIT
   7    1828  1936       1828  1936A HIT
   8    2205 14215       2205 14215A HIT
   9    2715 13753       2715 13753A HIT
  10    1619   915       1619   915A HIT
  11    2113 10358       2113 10358A HIT
  12    2710   401       2710   401A HIT
  13    1525  4099       1525  4099A HIT
  14    2708 11651       2708 11651A HIT
  15    2314 12327       2314 12327A HIT
  16    1816  6343       1816  6343A HIT
  17    2702 11279       2702 11279A HIT
  18     311  4188        311  4188A HIT
  19    2109 15967       2109 15967A HIT
  20    2718  3716       2718  3716A HIT
  21     610  3309        610  3309A HIT
  22      21  8173         21  8173A HIT
   CONFIRMING THE DECODED TEXT TO BE THE ORIGNAL MESSAGE
```

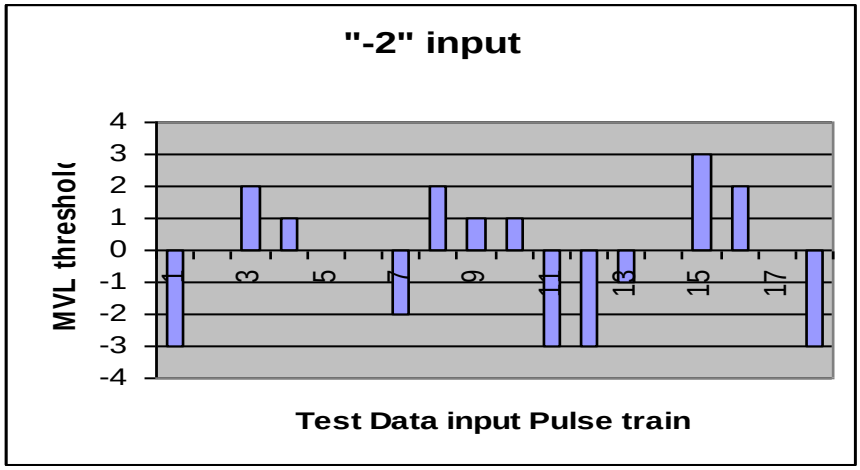
Appendix CH4.5 Results of the message encryption/decryption program

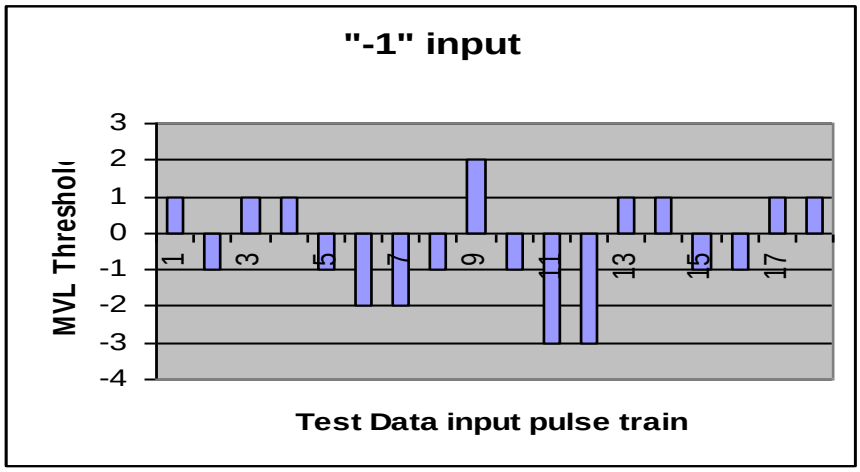
```
BLK/N0 ENCODED MESSAGE  EMBEDDED MESSAGE  ENCRYPTED MESSAGE  DECRYPTED
MESSAGE

   1       1507            1507 16417        7842  2346       1507 16417
   2       2704            2704   393       11750    16       2704   393
   3       2625            2628  2950        1608 12565       2628  2950
   4       1201            1204 15324       12555  9185       1204 15324
   5        527             528 12647        6926 12655        528 12647
   6       2006            2006  9629       13202  9812       2006  9629
   7       1827            1828  1936        8918  7380       1828  1936
   8       2205            2205 14215        1253  4343       2205 14215
   9       2715            2715 13753       13617  9536       2715 13753
  10       1619            1619   915       14606  2515       1619   915
  11       2113            2113 10358         954  8446       2113 10358
  12       2710            2710   401        9041 12792       2710   401
  13       1524            1525  4099       12250   259       1525  4099
  14       2706            2708 11651       10761    42       2708 11651
  15       2314            2314 12327        8380 10376       2314 12327
  16       1815            1816  6343        3664 11782       1816  6343
  17       2702            2702 11279       11786 16088       2702 11279
  18        311             311  4188        4520  7286        311  4188
  19       2109            2109 15967       12668 11909       2109 15967
  20       2717            2718  3716        7545  8423       2718  3716
  21        605             610  3309        1926  8348        610  3309
  22         20              21  8173       13933  2298         21  8173

              FINAL  COMPARATIVE ANALYSIS IN SUMMARY

                MISSON DURATION  = 25.53906 seconds

                MISSION ACCOMPLISHED THANK YOU LORD
```

QB 4.5 CODE

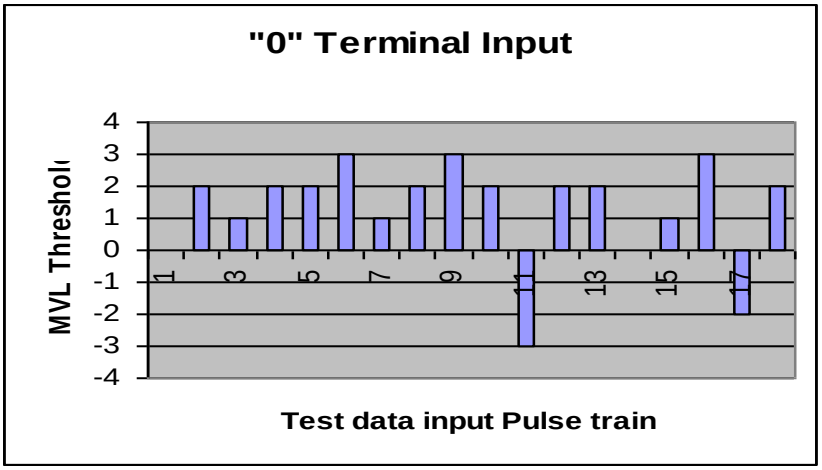
Appendix CH4.6 Multiplexing property test of the RR7SQSD T-gate Column Charts

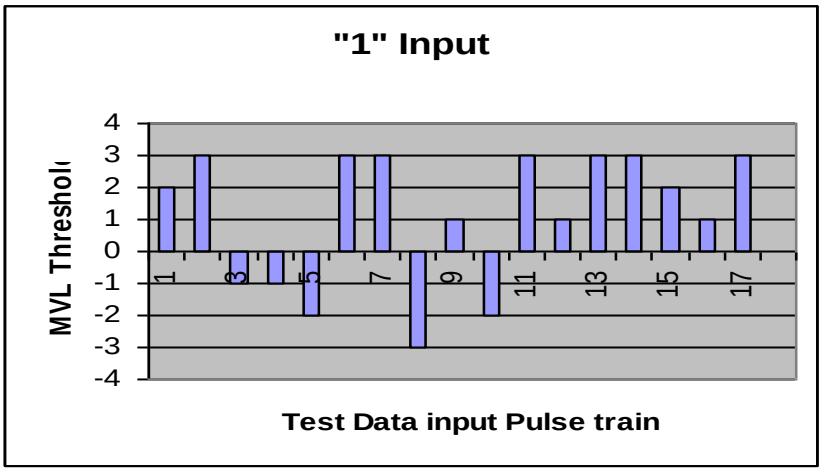


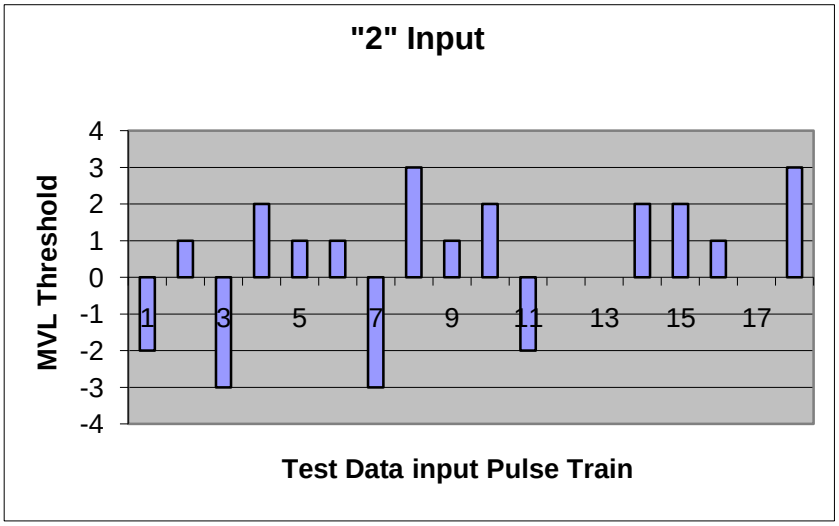




Appendix CH4.6 Multiplexing property test of the RR7SQSD T-gate Column Charts







Appendix CH4.6 Multiplexing property test of the RR7SQSD T-gate Column Charts