VLSI implementation of a high speed systolic finite field constant multiplier
by Sanjay Nirenda Mitra

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering
Montana State University

Abstract:
An architecture for a two-dimensional systolic finite field multiplier is proposed, where the present state feedback is organized as a set of n row pipelines and the next state equations organized as n column pipelines, where n is the width of the state variable vector. The intent here is to maximize clock speed.

The modules needed for the systolic finite field multiplier have been designed using a single-phase non-complemented clocking scheme, for dynamic precharge logic technique. The process involved in the full custom design of the systolic multiplier is described. Standard cell implementation of the systolic multiplier is also discussed.

Simulation and timing analysis suggests that the systolic finite field multiplier will have an advantage over the conventional multipliers, in applications where the volume of data is large.

VLSI IMPLEMENTATION OF A HIGH SPEED SYSTOLIC FINITE FIELD CONSTANT MULTIPLIER

by

Sanjay Nirendra Mitra

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

November 1991

APPROVAL

of a thesis submitted by

Sanjay Nirendra Mitra

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

11-19-91
Date

Chairperson, Graduate Committee

Approved for the Major Department

11-21-91
Date

Head, Major Department

Approved for the College of Graduate Studies

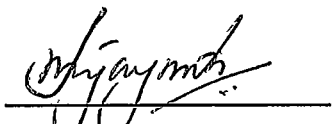
11/27/91
Date

Graduate Dean

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his/her absence, by the Dean of Libraries when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature 

Date 11/23/91

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank Professor Kel Winters for the guidance, suggestions, and inspiration he has provided. I also wish to thank Sandor Szego for his invaluable help during the software development for this project. Thanks also to Jaye Mathisen, for his help with the OCT tools and UNIX problems, and to Bob Wall for his helpful suggestions.

This work was supported by grants from the NASA Space Engineering Research Center (SERC) at the University of Idaho and the National Science Foundation MOSIS program.

TABLE OF CONTENTS

Page

1. INTRODUCTION 1
The Galois Field 1
Finite Field Multiplication 3
The Primitive Element 4
Finite-Field Constant Multiplier 4
Circuit Design 8
Design Methodology 8
Scope and Organization of Remaining Chapters 9

2. THE DESIGN PROCEDURE USING STANDARD CELLS 10
CAD Tools 10
Bdnet 11
Octflatten 12
MUSA 12
Wolfe 13
Mosaico 14
Padplace 14
Fabprep 14
Oct2ps 15
Creation of the Different Standard Cell Modules 15
The Crossbuff Cell 16
Ourxor Cell 17
Flipcross Cell 20
Ourxorsys Cell 22
FlipcrossII Cell 24
OurxorsysII Cell 27
Bdnet Code Generator Software For Constant Multipliers 30
Program Usage 30
Syntax 31
Generating the Multiplier Chip using Standard Cells 31
Generating the Conventional Multiplier Chip "Cgfmult1" 34
The "Cgfmult" Chip 39
Generation of the Systolic Multiplier Chip 41

3. CLOCKING AND DYNAMIC CIRCUITS 46

TABLE OF CONTENTS-- Continued

Static And Dynamic Flip-Flops 46
CMOS Dynamic Flip-Flop 48
Dynamic Precharge Logic 50
CMOS Domino Logic 53
No Race (NORA) Logic 54

4. FULL CUSTOM DESIGN FOR THE SYSTOLIC MULTIPLIER 58
Design Process 58
Resistance Estimation 59
Capacitance Estimation 59
Device Sizing 60
The Layout Procedure 62
Custom Version of the Flipcross Cell 62
Custom Version of the Ourxorsys Cell 69
Custom Version of the FlipcrossII Cell 76
Custom Version of the OurxorsysII Cell 83
The Custom Systolic Multiplier Chip "gfmult" 89
Generating the "gfmult" Chip 89

5. THE CHIP RELEASE PROCEDURE 93
The Release Process 93

6. CONCLUSIONS 97
Finite-Field Constant Multiplier Time-Area Considerations 97
Time Complexity for Scenario I 98
Area Complexity for Scenario I 98
Time Complexity for Scenario II 100
Area Complexity for Scenario II 101
Future Work 103

REFERENCES CITED 104

APPENDICES 107
Appendix A
Documentation on Finite Fields 108
Appendix B
Documentation for the Design Procedure Using Standard Cells 110
Appendix C
Documentation for the Design Procedure Using Custom Cells 164

LIST OF TABLES

Table Page

1. Logarithmic Table for 3-bit Galois Field 4
2. Output State Equations for Constant Multiplier 5
3. CMOS Physical Properties: Resistance and Capacitance 61
4. Time-Area Considerations for Scenario I 103
5. Time-Area Considerations for Scenario II 103

LIST OF FIGURES

Figure — Page

1. Block Diagram for Multiplier Circuit 5
2. Finite Field Constant Multiplier 6
3. Systolic Finite Field Multiplier 7
4. The Crossbuff Cell 16
5. Bdnet Description for the Crossbuff Cell 17
6. The Ourxor Cell 18
7. Bdnet Description for the Ourxor Cell 19
8. The Flipcross Cell 20
9. Bdnet Description for the Flipcross Cell 21
10. The Ourxorsys Cell 22
11. Bdnet Description for the Ourxorsys Cell 23
12. The FlipcrossII Cell 25
13. Bdnet Description for the FlipcrossII Cell 26
14. The OurxorsysII Cell 28
15. Bdnet Description for the OurxorsysII Cell 28
16. The Conventional Finite-Field Constant Multiplier 33
17. Cgfmult1.core 35
18. The Conv. Finite-Field Multiplier Chip "Cgfmult1" 38

LIST OF FIGURES--Continued

19. The Conv. Finite-Field Multiplier Chip "Cgfmult" 40

20. The Systolic Finite-Field Constant Multiplier 42

21. The Systolic Finite-Field Multiplier Chip "Sgfmult1" 44

22. The Systolic Finite-field Multiplier Chip "Sgfmult" 45

23. Static Flip-Flop 46

24. Dynamic Flip-Flop 47

25. CMOS Dynamic Flip-Flop 49

26. Dynamic Precharge Logic 50

27. Cascaded Dynamic Gates 51

28. Charge Distribution Problem 52

29. CMOS Domino Logic 53

30. NORA Logic 55

31. Ji-Ren's NORA Logic 56

32. Circuit Diagram for the Flipcross Cell 64

33. Circuit Layout for the Flipcross Cell 65

34. Tekspice Simulation for the Flipcross Cell 66

35. Tekspice Simulation for the Flipcross Cell 67

36. Circuit Diagram for the Ourxorsys Cell 70

37. Circuit Layout for the Ourxorsys Cell 72

38. Tekspice Simulation for the Ourxorsys Cell 73

39. Tekspice Simulation for the Ourxorsys Cell 74

LIST OF FIGURES--Continued

40. Tekspice Simulation for the Ourxorsys Cell 75

41. Circuit diagram for the FlipcrossII Cell 77

42. Circuit Layout for the FlipcrossII Cell 79

43. Tekspice Simulation for the FlipcrossII Cell 80

44. Tekspice Simulation for the FlipcrossII Cell 81

45. Tekspice Simulation for the FlipcrossII Cell 82

46. Circuit Diagram for the OurxorsysII Cell 84

47. Circuit Layout for the OurxorsysII Cell 85

48. Tekspice Simulation for the OurxorsysII Cell 86

49. Tekspice Simulation for the OurxorsysII Cell 87

50. Tekspice Simulation for the OurxorsysII Cell 88

51. The Custom Systolic Multiplier Chip "gfmult" 92

52. NEW-PROJECT Template 94

53. SUBMIT Template 95

54. FABRICATE/INFORMATION Template 95

55. Irreducible Primitive Polynomials of Minimum Weight 109

56. Standard Cell Non-Inverting Buffer 111

57. Standard Cell 2-Input-EX-OR 113

58. Standard Cell Delay Flip-Flop 115

59. Standard Cell Multiplexer 117

60. Musa Simulation File for the Crossbuff Cell 119

LIST OF FIGURES--Continued

61. Musa Simulation File for the Ourxor Cell 120

62. Musa Simulation File for the Flipcross Cell 121

63. Musa Simulation File for the Ourxorsys Cell 123

64. Musa Simulation File for the FlipcrossII Cell 125

65. Musa Simulation File for the OurxorsysII Cell 127

66. Program Listing : "stdconv.c" 129

67. Program Listing : "stdsys.c" 135

68. Bdnet Input File for the Conventional Multiplier "Cgfmult1" 141

69. Bdnet File for the Conventional Multiplier " Cgfmult1" placed within the pad ring 144

70. Statistics for the "Cgfmult1" Chip 153

71. Musa Simulation File for the "Cgfmult1 Chip" 154

72. Statistics for the "Cgfmult" Chip 156

73. Musa Simulation File for the "Sgfmult1" Multiplier 157

74. Statistics for the "Sgfmult1" Chip 159

75. Musa Simulation for the "Sgfmult" Multiplier 160

76. Statistics for the "Sgfmult" Chip 163

77. MOSIS Scalable Design Rules (Rev. 6) 165

78. Tekspice Deck for the Flipcross Cell 166

79. Tekspice Deck for the Ourxorsys Cell 169

LIST OF FIGURES--Continued

80. Tekspice Deck for the FlipcrossII Cell 173

81. Tekspice Deck for the OurxorsysII Cell 177

82. Program Listing: "cussys.c" 181

83. Bdnet Input File for the Custom Systolic Multiplier "gfmult" 187

84. Statistics for the "gfmult" Chip (ringed view) 203

ABSTRACT

An architecture for a two-dimensional systolic finite field multiplier is proposed, where the present state feedback is organized as a set of n row pipelines and the next state equations organized as n column pipelines, where n is the width of the state variable vector. The intent here is to maximize clock speed.

The modules needed for the systolic finite field multiplier have been designed using a single-phase non-complemented clocking scheme, for dynamic precharge logic technique. The process involved in the full custom design of the systolic multiplier is described. Standard cell implementation of the systolic multiplier is also discussed.

Simulation and timing analysis suggests that the systolic finite field multiplier will have an advantage over the conventional multipliers, in applications where the volume of data is large.

CHAPTER 1

INTRODUCTION

Galois fields, also called prime fields or finite fields, are the basis of most error detection and correction coding schemes in use today. Efficient error detection and correction schemes are a necessity in today's age of information technology. Finite field constant multiplier circuits constitute a critical component of error detecting and correcting algorithms and procedures [Chien64], [Winters84]. The proposed systolic finite field multiplier circuit is a custom **Very Large Scale Integrated (VLSI) Circuit**, designed to maximize performance, where the volume of data involved is large. The systolic multiplier is implemented using the **MOSIS** 2-micron (drawn gate length) SCMOS, N-well process. The design procedure using standard cells as well as custom cells is described.

The Galois Field

A very brief introduction to finite field theory is given in this section. For more detailed information on Galois fields refer to [Clark81],[Sweeny91].

A finite field, also called a Galois field and designated GF(q), is a finite set of q elements for which we have defined some special rules for arithmetic [Clark81]. These rules are not very different from those which are used to do

arithmetic with ordinary numbers. The principal difference is that there is only a finite set of elements involved. Finite fields can be created where the number of elements is an integer power of any prime number p. Finite fields whose size is an integer power of 2 will be of main interest here. A n-bit Galois field consists of 2^n values, each n bits wide, for example, a 3-bit galois field has 2^3 values, each 3 bits wide. The properties of a finite field are as follows [Sweeny91]:

1. There are two defined operations, namely addition and multiplication.
2. The result of adding or multiplying two elements from the field is always an element of the field.
3. The field contains a multiplicative identity element, 1, and a additive identity element, 0.
4. There exists an additive inverse and a multiplicative inverse element for each element of the field.
5. Associative, commutative, and distributive laws apply in the usual manner.

These properties cannot be satisfied for all possible field sizes, which means that only certain sizes of finite field can be constructed. The properties can, however, be satisfied if the field size is any prime number or any integer power of a prime [Sweeny91]. For an n-bit field, these bits may be thought of as binary coefficients to a polynomial of degree n-1. For addition, corresponding coefficients are exclusive-ored together. For multiplication, the resulting polynomial product must be reduced modulo a special irreducible polynomial p(x) of degree n, called

a primitive polynomial. This polynomial must have the property that it cannot be factored using only polynomials with coefficients from GF(q).Tables of primitive polynomials are widely available in the literature, and a selection is shown in Appendix A.

Finite Field Multiplication

An example of how finite field multiplication is performed is shown. Consider a 3-bit field, having a primitive polynomial $p(x) = x^3 + x + 1$. Multiplication of the elements 100 and 011 in polynomial form appear as:

$$\begin{array}{lll} & x^2 & \text{(element 100)} \\ \text{[mult]} & x + 1 & \text{(element 101)} \\ & \text{--------} & \\ & x^3 + x^2 & \text{(1)} \end{array}$$

The result must be transformed to the 3-bit field by reducing it modulo p(x). This is done using the identity p(x) = 0:

$$p(x) = x^3 + x + 1 = 0$$

$$x^3 = x + 1 \qquad \text{(2)}$$

Substituting x^3 in polynomial (1):

$$x^3 + x^2 = x + 1 + x^2 \qquad \text{(3)}$$

This corresponds to the binary value 110 and is an element of the 3 bit field.

The Primitive Element

In finite fields, the concept of logarithms apply just as they do with ordinary numbers. It is a property of all finite fields that there exists at least one element, called the **generator** or **primitive** element (which is a root of p(x)), such that every non-zero element of a field may be expressed as a power of the primitive element. For example, if α represents the root of p(x) where $p(x) = x^3 + x + 1$, then all 7 non-zero elements of the 3-bit field may be expressed as powers of α as shown in table 1.

Table 1. Logarithmic Table for 3-bit Galois Field

Power of α (log form)	Polynomial Element	Binary Form (anti-log form)
0	1	001
1	x	010
2	x^2	100
3	x + 1	011
4	$x^2 + x$	110
5	$x^2 + x + 1$	111
6	$x^2 + 1$	101
7	1	001

Finite-Field Constant Multiplier

A finite field constant multiplier having a polynomial $p(x) = x^3 + x + 1$, with an exponent $\alpha = 1$, multiplies the input vector by a constant $\alpha^1 = 010$ (refer table 1). Similarly for $\alpha = 2$, the input vector is multiplied by $\alpha^2 = 100$.

Figure 1, shows a block diagram for a multiplier circuit with $p(x) = x^3 + x + 1$. Table 2 shows the output state equations when the input vector is multiplied by different powers of α. Multiplication of an input vector by α^m corresponds to loading the shift register, shown in figure 1 with the vector, then clocking the register m times.

Table 2. Output State Equations for Constant Multiplier

Exponent	S_2	S_1	S_0
$\alpha = 0$	S_2	S_1	S_0
$\alpha = 1$	S_1	$S_2 + S_0$	S_2
$\alpha = 2$	$S_2 + S_0$	$S_1 + S_2$	S_1

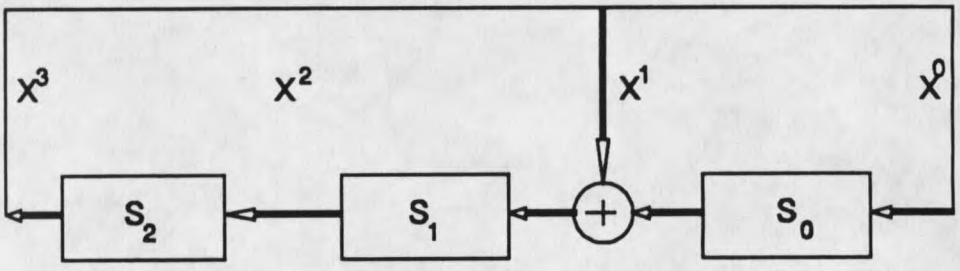


Figure 1 - Block Diagram for Multiplier Circuit

A finite-field constant multiplier consists of a synchronous register to store the present-state and an array of XOR gates representing modulo-2 adders