FOOTSTEPS IN AN EMPTY VALLEY

NC4000 SINGLE CHIP FORTH ENGINE

Dr. C. H. Ting

Third Expanded Edition

Offete Enterprises, Inc.

1988

First Edition: November 1985

Second Edition: April 1986

Third Edition: March 1988

PUBLISHED BY

OFFETE ENTERPRISES, INC.
1306 South B Street
San Mateo, CA 94402
Tel. (415) 574-8250

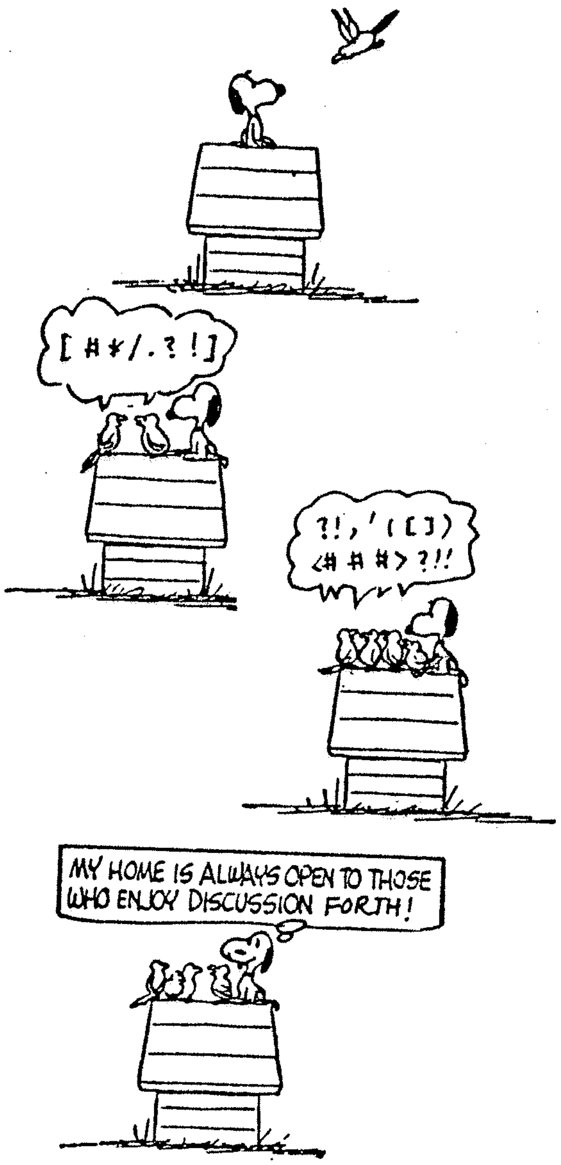
Printed in the United States of America

Dedication

To my parents

for their love, support, and appreciation.

FORTH INTEREST GROUP



PREFACE TO THE THIRD EDITION

Dr. Glenn Haydon, one of the developers of the WISC (Writable Instruction Set Computer) Forth engine, often made the remark that it was unfortunate that Chuck Moore built the NC4000 chip and cast Forth into silicon. He lost the freedom to change his mind, as he often did. Indeed, it was very difficult to make any change in NC4000. Novix had not been able to fix the bugs in the NC4000 prototype chip, or to bring out the NC6000/5000 with enhanced features, even with the infusion of large amount of venture capital after its incorporation. After two years since NC4000P was released, it is still the only chip available from Novix.

Although Chuck Moore could not do much on NC4000, he is not standing still either. He talked about the 32 bit 'Buffalo Chip' on several occasions, although it is not clear how much has been committed to its design and construction. He also kept on revising the FORTHkit and the cmFORTH system. The latest version was released in December 1987. He sent me a copy of the new cmFORTH to be published in 'More on NC4000' and allowed me to use it in the third edition of this book.

It is a pleasure to go through the new cmFORTH and compare it line by line with the old version. Chuck made many changes, polishing the code in more than one way. Many definitions are refined. Many names are changed. The style and code layout are also much improved for readability. Shadow screens are included to provide functional description for every word defined. He managed to shave off 15 lines of code and save a whole screen. He also put shades at the right and bottom edges of the text blocks, making them rise above the page. It is rather pleasing to the eyes.

His persistent obsession in pursuing the simplest expression to be shared by NC4000 and its users is always fascinating to me. He must have felt the enormous responsibility of a system programmer and a language designer, in that every instruction he deleted and every cycle he saved will be multiplied by the billions in memory and times saved by the users. The old Chinese master ought to have made this observation when he declared: "For knowledge, add a little everyday. For wisdom (Tao), erase a little everyday". The Tao of Forth is not something vague and ethereal. It is embodied in cmFORTH for us to see, to feel, to study, and to meditate on, if we cared.

In this edition I tried to include information and developments on NC4000 over the last two years. Many new sections are added based on my papers appeared in 'More on NC4000'. However, the major focus of this book is still Chuck Moore's cmFORTH.

Observing Chuck's programming style and how he constructs large structures from the components is the best way to gain maturity in Forth programming. Simplicity manifests itself in correctness, flexibility, capability, and productivity.

In last year (1987), we saw several Forth engines: the 32 bit Forth chip from the Applied Physics Laboratory in John Hopkins University, both a 16 bit and a 32 bit WISC engine from WISC Technology, and the 16 bit FORCE chip set from Harris Semiconductor. We can expect more entries into this field which will generate more interests and excitement in the coming years. Finally, we start to see blossoms from the seeds Chuck Moore planted two decades ago.

A NC4000 Users Group was formed in the San Francisco Bay Area in 1986. The Silicon Valley Chapter of the Forth Interest Group hosts the NC4000 Users Group Meeting once every three months, on the fouth Saturdays in January, April, July, and October. The discussions in these meetings are always lively and sometimes provocative. For members outside of the Bay Area, a newsletter 'More on NC4000' helps the circulation of technical information about NC4000 and related products. Volume 7 of 'More on NC4000' was just released. We hope that it will provide timely and useful exchange of ideas and techniques among the users.

Chen-hanson Ting

March 1988
San Mateo, California

FORWARD TO THE FIRST EDITION

空 KUNG 谷 KU 足 TZU 音 YIN

KUNG KU TZU YIN --- FOOTSTEPS IN AN EMPTY VALLEY

"Footsteps in an empty valley" is a Chinese ideographic phrase with very deep poetic connotation. It is used to describe the emotional feeling when one is about to meet a long missed friend as his footsteps are nearing. The picturesque setting is an empty, desolated valley this person had chosen to retire. He has as his companions the trees and the flowers, a small flowing brook perhaps, and wildlifes of the woods--the best mother nature has to offer--except for trusted friends whose friendships he had to sever as the price of his retirement.

One day he was awakened amidst all the familiar sounds of his environment--whispering of the trees, splashing of water, and soft songs of the birds--by the footsteps of someone he dearly missed all these years. Who would travel this far to this distant valley to visit, but the most intimate and the most trusted of friends?

Living in this Silicon Valley full of people, money, energy, activity, and ideas; we've seen new things invented and new products introduced with rapid pace. Some are successful. Most do not see the light of day. Even the most successful fade in a couple of years. Amongst the high pitched, loud sounding hype, there is always this silent loneliness deep down inside. Which voice shall we heed? What direction are we heading? Where is the best and the truest to be found?

We have witnessed hosts of microprocessors and microcomputers marching from cradle to grave, right before our eyes. Languages and operating systems come and go. Even in Forth, which I use to code for a living and write about to entertain, we've seen good work done and disappear, come and go. Have we seen the best yet?

The NC4000 chip made by Novix is very refreshing. Seeing what these 4000 gates can accomplish that took 100,000 gates to do in other microprocessors makes you wonder if the computer industry is driving into a blind alley at 100 mph. It was worth the long wait to see this chip become a working reality. We can now dream about what we can do with this chip as it becomes available commercially.

What seemed to be a very distant interest suddenly became a personal dedication when we designed and built a single board computer using the NC4000 chip as its CPU. Chuck Moore graciously provided us with his cmFORTH to run on our board, and helped us to bring this system up. He also released cmFORTH to the public domain to help us learn about this chip.

Chuck stated that the primary goal of the cmFORTH system was to build a Forth computer small enough to fit in 2K words of memory, yet powerful enough to recompile itself. With this system, its user can at once do development work and change the system as his application demands. He has achieved his goal, with only 30 screens of source code! This system has life because it can reproduce. Most of the other languages and operating systems are sterile. They can grow bulky and fat, devouring whatever resources the hardware can afford, but they are sterile. The user cannot use them to build his own system.

I took the cmFORTH source code back home and studied it carefully line by line. I was amazed at the cleanliness of its design and the conciseness of its expression. It was like reading a poem. Every word served a useful purpose, and served it well. No word was included thoughtlessly and any superfluous material has been purged. Thus he was able to bundle the kernel, the interpreter, an optimizing compiler, and a target compiler together in this small package.

I have written code and I have seen other people's code. Most of the code were not impressive because they were unpolished products thrown together for a quick sell. The craftsmanship might be adequate but the product was not art. In contrast, cmFORTH stands out like a gem--a piece of art. It is the result not only of ingenious design and elegant implementation, but also of patient honing and polishing. It is the footstep in this empty valley.

Beside a few words with long nested structures, I did not have much difficulty in reading cmFORTH. Chuck's style was familiar to me because I had documented his early polyFORTH some years ago when Forth documentation was the exception rather than the rule. However, I thought it would be a useful service to many actual and potential NC4000 users if the cmFORTH source code were fully documented and its fundamental logic laid out in a more structured fashion. The heart of this book is a line by line, screen by screen explanation of the source code. It would be especially helpful to the user when the stack picture got fuzzy and the logic seemed tied in knots. At the very least, the user will have another point of view to explore Chuck's ideas besides reading the source code itself.

The NC4000 is truly a milestone in computer technology-- more so than the much touted RISC computer. The Berkeley RISC machine was essentially a rediscovery of the original Von Neuman's ENIAC design. The only addition was the overlapping registers used to facilitate parameter passing between procedures. NC4000 is much more sophisticated than the RISC machine in its dual stack architecture, single cycle subroutine call and return, and the externally microcoded instruction set. To take advantage of the unique architecture of NC4000 and to make the best use of its powerful instruction set, the user needs a firm grasp on the inner mechanism of this chip. A systematic exposition of this chip is therefore necessary to bring all the information about this chip into sharp focus. Some knowledge about the hardware structure in this chip is mandatory in order to understand the software system embedded in cmFORTH; although lacking this knowledge would not prevent the user from programming this chip in the normal Forth style. A long section in this book is devoted to the chip itself to provide background information on the chip.

The NC4000 chip does not work by itself. You have to connect it to some RAM and ROM memory to build a computer. Circuit schematics are provided for such a computer so that the user can build it with minimal parts and labor, and be productive in a short time. Many programming tips and tutorial examples are also included. Nevertheless, the richest source of coding examples are in the cmFORTH source code itself, where the user can find practical solutions to a broad spectrum of problems an operating system has to solve. There is much we can learn from Chuck Moore both in terms of programming techniques and programming style.

I didn't really have much time to explore all aspects this computer and the NC4000 chip. This manual represents the scope of my understanding at this moment. As time passes, I will make additions and updates and hope that you will keep me informed of your opinions and suggestions. If the chip is to be successful, it needs the entire Forth community to support it by providing viable applications and services to users not fluent in the language. If it is successful, we will ride on its coattails for a long, long while.

It is superfluous to acknowledge Chuck Moore, because the acknowledgement is implicit every time I utter 'Forth". However, his personal help in bringing up our first NC4000 system and providing it with cmFORTH greatly accelerated our pace in making this information available to Forth users. Dr. George A. Nicol and Mr. Scott Reinhart of the Software Composers were very helpful in providing information on their SC1000 computer which also uses NC4000 as its CPU. Mr. John

Peters and Dr. and Mrs. Albert Ting read the manuscript and made numerous corrections and suggestions.

My best wishes to you and to your NC4000 computer.

Chen-hanson Ting

March 1986
San Mateo, California

CONTENTS

Dedication		i
Preface for the Third Edition		iii
Forward fo th First Edition: Kung Ku Tzu Ying		v
Contents		ix
Figures		xiii
Tables		xiv
I.	Introduction	1
1.1	Historical background	1
1.2	RISC Panacea	3
II.	The Chip	9
2.1	Features of NC4000 chip	9
2.2	External data paths	11
2.2.1	Main memory	11
2.2.2	The data stack and the return stack	13
2.2.3	B-port and X-port	15
2.2.4	System timing and control	16
2.3	NC4000 architecture	19
2.3.1	The internal registers	19
2.3.2	Program sequencer	22
2.3.3	Data stack and return Stack	23
2.3.4	The ALU	25
2.3.5	The I/O ports	25
III.	The Instruction Set of NC4000	29
3.1	Classification of NC4000 instructions	31
3.2	The ALU instructions	36
3.3	I/O and memory instructions	41
3.4	Graphic models of some NC4000 instructions	45
3.4.1	The model of NC4000 ALU	45
3.4.2	The SWAP group	47
3.4.3	The DUP group	49
3.4.4	The binary ALU group	51
3.4.5	The multiply/divide group	53
3.4.6	Miscellaneous instructions	56
IV.	NC4000 Computers	59
4.1	Commercial products using NC4000 chip	59
4.1.1	The early alphabetic boards	59
4.1.2	The FORTHkits from Computer Cowboys	60
4.1.3	Products from Novix, Inc.	61
4.1.4	Products from Silicon Composers	61
4.1.5	Other companies and products	62
4.1.6	List of manufacturers	63
4.2	Build your own NC4000 computer	65
4.2.1	The CPU section	65
4.2.2	I/O ports	66

4.2.3 Main memory 67
4.2.4 The data stack and the return stack 71
4.3 Circuit boards for NC4000 computer 72
4.4 Hardware enhancements 73
4.4.1 PAL memory decoder OF5138 73
4.4.2 Stack expansion counter OF5493 76
4.4.3 Another novel memory decoding technique 79
4.4.4 The interrupt fixer NIC-4 80

V. The cmFORTH Operating System 83
5.1 The kernel 83
5.1.1 The primitive Forth words 84
5.1.2 Memory accessing words 86
5.1.3 Multiply and divide 88
5.2 System variables 90
5.3 Terminal input and output 92
5.3.1 Primitive input and output words 92
5.3.2 The line input and output words 93
5.3.3 Other terminal I/O words 94
5.4 The number conversion words 95
5.4.1 Convert digits to binary number 95
5.4.2 Convert binary number to ASCII string 96
5.4.3 Memory dump 98
5.4.4 Message output 99
5.5 Serial disk 101
5.5.1 Disk buffer manager 101
5.5.2 Disk read and write 103
5.6 The text interpreter 106
5.6.1 Parsing of words 106
5.6.2 Dictionary search 108
5.6.3 The interpreter 111
5.6.4 Power up and reset 113
5.7 The compiler 117
5.7.1 Compiler loop 117
5.7.2 Defining words 121
5.7.3 Control structures 124
5.7.4 The NC4000 assembler 127
5.7.5 The compiler vocabulary 131
5.8 Optimizing compiler 132
5.8.1 The smart ; compiler 133
5.8.2 The smart ALU function compiler 134
5.8.3 The shift compiler 136
5.8.4 Merging of DUP 137
5.9 The target compiler 138
5.9.1 Utility 138
5.9.2 Target dictionary 139
5.9.3 Variables in the target system 142
5.9.4 Separate Target dictionary from host dictionary 142
5.9.5 Target compiler in action 146

VI. Programming Tips 147
6.1 Benchmarks 147
6.2 WORDS--listing the vocabulary 149
6.3 Memory dump 151
6.4 Line editor 151
6.5 Stack pictures 153
6.6 Display the internal registers 155

6.7 Input and output 155
6.8 PICK and ROLL 159
6.9 Square-root 159
6.10 Terminal and disk server on IBM-PC 160
6.11 Arcsine by intrapolation 163
6.12 High speed pattern generator 163
6.13 A/D conversion with NC4000 168
6.14 Fast byte flip 173
6.15 More vocabularies 174

Appendix A. cmFORTH Source Listing 177

Appendix B. NC4016C preliminary Electrical Specification 203

Appendix C. Novix NC4000 Microprocessor 205

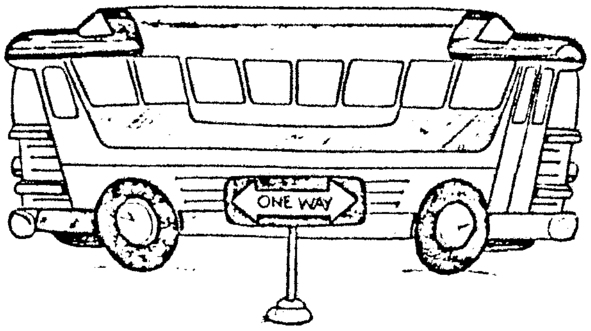
Appendix D. cmFORTH Glossary 211

Appendix E. Glossary Sorted by Vocabulary and Type 217

Appendix F. Comsol NIC-4 Interrupt Controller 223

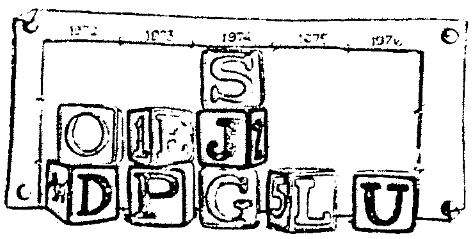
Index 227

-TRAILING



ASSEMBLER

BLOCK



FIGURES

2.1. NC4000 pin layout 10
2.2. External data paths of NC4000 12
2.3. NC4000 memory map 14
2.4. Timing diagrams of NC4000 17
2.5. The architecture of NC4000 20
2.6. The arithmetic Logic Unit 26

3.1. Encoding of NC4000 instructions 30
3.2. NC4000 instruction formats 32
3.3. Data paths and registers in ALU section 35
3.4. Another view of ALU 46
3.5. The SWAP group 48
3.6. The DUP group 50
3.7. The binary ALU group 52
3.8. The multiply/divide group 54
3.9. Miscellaneous instructions 58

4.1. CPU section of a NC4000 computer 64
4.2. Memory decoding of a small NC4000 computer 68
4.3. Memory decoding of a large NC4000 computer 69
4.4. Data and return stack of a NC4000 computer 70
4.5. Pinout of OF5138 decoder chip 75
4.6. Memory map of a 512K byte system 75
4.7. pinout of OF5493 counter chip 77
4.8. Expansion of data stack for NC4000 78
4.9. Decoding memory with a 74HC74 81

6.1. Simple benchmark programs 148
6.2. Vocabulary definitions and WORDS 150
6.3. Regular DUMP routine 150
6.4. Line editor 152
6.5. .S and .RS to show stack pictures 154
6.6. Internal registers 154
6.7. Input and output demonstration 156
6.8. PICK and ROLL 158
6.9. Square-root 158
6.10. Terminal and disk server 161
6.11. Source code of intrapolation 164
6.12. Schematics of the pattern generator 166
6.13. Program to control the pattern generator 164
6.14. A/D conversion with Datel ADC815 170
6.15. A/D conversion with National ADC0820 171
6.16. NC4000 code for A/D conversion 172
6.17. Byte flipping 172

TABLES

1.1.	The von Neumann machine instruction set	5
1.2.	The RISC instruction set	6
1.3.	The NC4000 instruction set	8
2.1.	NC4000 pin names and functions	11
2.2.	Internal registers in NC4000	21
3.1.	ALU code and function	37
3.2.	Y-port selector	37
3.3.	Data stack code and functions	39
3.4.	Shift code and functions	39
3.5.	Valid ALU instructions	40
3.6.	Valid I/O and memory instructions	42
3.7.	NC4000 internal registers	44
3.8.	Function of bits in the I/O registers	44
4.1.	Pins of OF5138	74
6.1.	Machine cycles for 16 bit integer operations	148