

# FORTH NOTEBOOK

by C. H. Ting . PHD



**OFFETE ENTERPRISES, INC.**

**1986**

# **FORTH NOTEBOOK**

**by C. H. Ting . PHD**

**OFFETE ENTERPRISES, INC.**

***1986***

(c) Copyright, 1983 by C. H. Ting

FIRST EDITION, SEPTEMBER 1983 (FIRST PRINTING)

MAY 1985 (SECOND PRINTING)

JULY 1986 (THIRD PRINTING)

\*\*\*\*\*

All rights reserved. This book, or any part thereof, may not be reproduced for commercial usages without written permission from the Author.

\*\*\*\*\*

Printed in the United States of America

by

*Offete Enterprises, Inc.*

1306 SOUTH 'B' STREET  
SAN MATEO, CALIFORNIA 94402  
TEL: (415) 574-8250

## CONTENTS

|                               |     |
|-------------------------------|-----|
| Preface                       | i   |
| Dialectic Variations          | 1   |
| Games in FORTH                | 3   |
| 1. Animal                     | 4   |
| 2. Calendar                   | 8   |
| 3. Depth Charge               | 12  |
| 4. Game of Life               | 14  |
| 5. Guess a Number             | 18  |
| 6. Gunner                     | 20  |
| 7. Hello                      | 24  |
| The GO Game                   | 32  |
| 1. Coding GO Games            | 33  |
| 2. GO in FORTH                | 36  |
| 3. Joseki                     | 46  |
| 4. Gomuko                     | 50  |
| Tools                         | 56  |
| Controller Programs           | 61  |
| 1. PROM Programmer            | 62  |
| 2. A/D Converter              | 66  |
| 3. DMA Interface              | 68  |
| 4. Robot Control              | 71  |
| The Image Processor           | 79  |
| 1. Image processing           | 82  |
| 2. Run Length Coding          | 108 |
| 3. Connectivity Anylysis      | 115 |
| 4. A Simple Graphics System   | 130 |
| Microassembler                | 136 |
| Continuous Fourier Transform  | 170 |
| Bacteria Identification       | 186 |
| The Virtual FORTH Computer    | 194 |
| FORTH Seminar Viewgraphs      | 211 |
| 1. Advantages of FORTH        | 213 |
| 2. Introduction to FORTH      | 218 |
| 3. Programming in FORTH       | 222 |
| 4. Structured Programming     | 227 |
| 5. Utilities in FORTH         | 233 |
| 6. FORTH Virtual Computer     | 240 |
| 7. FORTH Operating System     | 248 |
| 8. FORTH Programming Language | 252 |
| Formal Definition of FORTH    | 268 |
| Basic-FORTH                   | 276 |
| Index                         | 283 |

## FIGURES

|     |   |     |
|-----|---|-----|
| 1.  | Image Processor Architecture                  | 78  |
| 2.  | Image Array Processor in IP5500               | 80  |
| 3.  | Memory Mapped Registers in IP5500             | 81  |
| 4.  | Microcodes of Simple Examples                 | 144 |
| 5.  | Partial List of Microcodes of Supersixteen    | 158 |
| 6.  | Partial List of Microcodes of Disk Controller | 169 |
| 7.  | Comparison of Computer Languages              | 212 |
| 8.  | Extensions of FORTH Computer                  | 214 |
| 9.  | Program Development Cycle                     | 216 |
| 10. | Layered Structure of FORTH Computer           | 217 |
| 11. | Structures in FORTH Language                  | 228 |
| 12. | The Virtual FORTH Computer                    | 241 |
| 13. | Memory Map and Pointers                       | 243 |
| 14. | Instruction Format                            | 245 |
| 15. | High Level and Low Level Instructions         | 247 |
| 16. | Flow Chart of Text Interpreter                | 250 |
| 17. | Syntax of FORTH                               | 253 |
| 18. | Software Costs                                | 264 |

## PREFACE

My experience in the last few year in teaching FORTH to people of different backgrounds was that Starting FORTH was quite sufficient as a textbook, introducing people to the basics of FORTH. However, to more experienced programmers, the materials in it are not enough. Though there are numerous examples and exercises, most of them are brief and the scope of the examples is also limited. I was constantly asked to provide real applications besides teaching examples. Another shortcoming is that it does not deal with the internals of the FORTH operations at the machine code level. There is a wide gap between Starting FORTH and the source code as shown in the fig-FORTH Model and Installation Manual.

We've seen a number of books on FORTH appearing in the last year. They are welcome news. However, most of them did not exceed Starting FORTH either in the scope of their treatments or in the amount of examples towards more realistic applications. Starting FORTH probably will remain the bible of FORTH to beginners for many years to come.

There are two areas where FORTH books are of urgent need: one is in presenting program design with examples of moderate complexity, and the other is to explore the FORTH computer in a deeper level, dealing with real CPU's and instruction sets. What I wanted to present in this Notebook are some of my personal efforts in these two directions.

In this Notebook, I collected many programs which I used for teaching purposes. Many interesting games were translated from BASIC into FORTH. These translations offer some insight into the natures of these two different languages. I am very interested in the computerization of the ancient oriental GO game. A few programs in this field are included. My profession requires the utilization of digital image processors. A number of programs dealing different aspects of image processing were used to demonstrate that FORTH is a more natural language for this type of applications. I included them here as working examples of practical applications. They are of very limited use to people who do not have the same image processor as mine. Nevertheless, they do illustrate many commonly used techniques in handling digital images.

Recently, I was involved in a bit slice microprocessor project and developed a microassembler to assemble microcodes. It is a very simple solution to a rather complicated problem. I simply cannot resist the temptation to publish it here. Two other programs of similar nature were included for the same reason. One was developed when I was with Yang Ming Medical College in Taipei, to identify bacteria according to the results of a set of tests. Another one was to explore the technique of continuous Fourier transform I proposed a number of years ago.

I taught a number of short courses on FORTH and used FORTH screens to prepare viewgraphs. These courses were generally organized into eight sessions, treating subjects ranging from data and return stacks to text and inner interpreters. These viewgraphs are also collected in this Notebook for those who might want to use them as teaching aids.

In my book Systems Guide to fig-FORTH, I avoided the very unpleasant task of explaining the nucleus words which were in 6502 machine codes as published in the fig-FORTH Model and Installation Manual. I worked out these words using PDP-11 codes with some commentary. This section, hopefully, will be a worthy appendix to the Systems Guide.

I am greatly indebted to the members of the Taiwan FIG Chapter who encouraged me to put together this Notebook. The very informal format adopted here was decided when I was participating in their Summer Workshop, July 21 to 25, 1983. It is a response to their immediate needs which I believe are shared by many young FORTH communities. Applications programs dealing with real world problems sometimes are more interesting than textbook examples which were designed to sell a viewpoint or a language.

Thanks are due to Mr. Anson Averell who carefully read through the manuscript and made numerous suggestions and corrections.

All the programs presented in this Notebook were fully debugged as far as I could test them. However, I cannot guarantee that they are bug-free. Users beware! Like all responsible software manufacturers, this is the proper place to state that I shall not be liable for errors contained herein or for the furnishing, performance, or use of the material contained herein.

San Mateo, California  
September, 1983

Chen-hanson Ting

## DIALECTIC VARIANCES

Almost all the programs presented in this notebook were written in the earlier version of poly-FORTH, developed by FORTH, Inc., released in Nov. 1979. Names of many instructions in this version of FORTH are different from the other dialects such as fig-FORTH, FORTH-79 Standard, and the latest version of poly-FORTH, i.e., poly-FORTH II.

For the convenience of readers who are more familiar with fig-FORTH or FORTH-79, I had prepared a short list on the variance between these three dialects. This list is by no means complete nor even exhaustive. It contains only the instructions I used very often in this Notebook. I felt obliged to warn the reader of these variances, so that proper modification can be made upon transporting programs to other FORTH systems.

## DIALECTIC VARIANCES

| poly-FORTH     | fig-FORTH          | FORTH-79           |
|----------------|--------------------|--------------------|
| >IN            | IN                 | >IN                |
| ABORT" xxx"    | ABORT              | ABORT              |
| AGAIN          | REPEAT             | REPEAT             |
| ---            | AGAIN              | AGAIN              |
| BLANK          | BLANKS             | ---                |
| EMPTY          | COLD               | COLD               |
| END            | UNTIL              | UNTIL              |
| EXIT           | ;S                 | EXIT               |
| BEGIN-IF-AGAIN | BEGIN-WHILE-REPEAT | BEGIN-WHILE-REPEAT |
| FLUSH          | FLUSH              | SAVE-BUFFERS       |
| MINUS          | MINUS              | NEGATE             |

## DIALECTIC VARIANCES

| poly-FORTH        | fig-FORTH     | FORTH-79      |
|-------------------|---------------|---------------|
| MOVE              | CMOVE         | CMOVE         |
| ---               | STATE         | STATE         |
| THEN              | ENDIF         | THEN          |
| VARIABLE          | n VARIABLE    | VARIABLE      |
| WORD ( c -- addr) | WORD ( c ---) | WORD ( c ---) |
| [']               | '             | '             |



## PROGRAM TRANSPORTABILITY

Most of the words are common to all FORTH systems and the FORTH programs are transportable between different systems. However, care must be exercised because of the differences in a few words. I would not expect readers to take the programs and blindly type them in their FORTH computer without looking at the contents carefully. It is probably better if one would question the way a word was defined and experiment with improvements and enhancement. It is always a pleasure to shave off a few words in a definition, or to use a different approach to achieve the same goal while speeding up the execution.

## THREE CHARACTER NAMES

Poly-FORTH had been criticized vehemently for retaining only the first three characters and the character length of word names. In using this naming convention for a number of years, I don't feel it is a major limitation as long as I have the freedom in choosing names. It is not difficult to find unique names for all the words in an application package. However, there were occasions that my tongue slipped and seemingly different names were not identified by the system as such. Strange things happen depending on the loading sequence of screens and the weather. Many hours were spent before realizing that there was a conflict in names. These things were part of the problems one would have to solve in the debugging processes.

The only instances that the three character names became a limitation were when I had to write programs conforming to other people's naming schemes. Two examples in this Notebook fell into this category. In the programs on image processing, I tried to stick to the naming conventions used by De Anza, the IP manufacturer. Most register mnemonics were 6 characters long and the distinguishable characters were often in the fifth or the sixth character. In the microassembler program, many of the microcode fields definitions and operator mnemonics were indistinguishable in the first three characters. I had to doctor these names so that significant characters were positioned in the first three characters, making names not quite natural to those familiar with the original AMD literature. Only in these agonizing hours, I missed the 31 character names in fig-FORTH.