

FORTH-NODAL PREPARATIVE: IMPLEMENTATION OF A FORTH MONITOR NUCLEUS INTO A 16-BIT PC

Yuuki KAWARASAKI
Department of Physics, JAERI
Tokai, Ibaraki 319-11

A Forth monitor nucleus has been transported into a 16-bit personal computer (PC) from an original 8-bit PC's by rewriting primitive words only, for aiming at the construction of an interactive and distributed multi-computer control system.

A typical test example of execution speed is demonstrated, yielding in an encouraging result.

Further expansion, including real-number arithmetics and multi-segment programming, is now in progress.

Introduction

This paper stems up from the previous works/1,2/.

Yet another area of accelerator science and technology is now more closely coming interwoven with computer science: typically in computer-aided-design of accelerating structures with beam behavior simulations and in application for their control.

In the latter case, there are many ways how to construct computer-based control and/or data-logging systems both in hardware configurations and in sorts of software implemented herein. In these applications, a somewhat different feature may reasonably be adequate from ordinary environments, from the viewpoints of high-speed execution and programming flexibility which includes the system expandability.

A distributed multi-computer system, typically like NODAL systems/3/, has already proven the effectiveness. A PC-and-intelligent CAMAC system/4/ or a linked single-board-computer (SBCs) system/5/ eventually formed themselves a simplest distributed system.

An interactive programming environment can be provided by use of an interpreter language instead of compiler languages: in NODAL, the BASIC-like interpreter is used. However, the execution speed of the task written in the above is not sufficient. FORTH /6,1/ is then expected to overcome the deficiency of the speed, keeping other kinds of distinctive characteristics; self-reproducing, as a Forth system can be described in Forth itself (high expandability) and both in low level like an assembler and in high level (high productivity).

In addition, a Forth system is simple and then it can rather easily be assembled. This situation allows births of many dialects of Forth, just like LISP. NEON used in well known APPLE's MacIntoshs and STOIC in rumor in EPSON's are also one of the dialects.

Forth Monitor

The structure of the monitor is basically the same as the 8-bit PC's, whose CPU is Z-80/4,5/. The transportation into the 16-bit PC's, whose CPU is 8086 or equivalent, has been performed by rewriting only the kernel, in which several tens primitive words have been assembled. A 8086 has been naturally developed from a 8085, similar to a Z-80. A 32-bit 80386 is then similarly developed from a 8068. Further transportation can then be in this direction. Some words in the 8086's can be assembled more compact than in the Z-80's. Newly prepared words are due to the differences between the hardwares.

Speed Test Example

At present, only integer arithmetics is fully available in the developing Forth system. A heart of a sample program written in BASIC is then chosen as/7/:

```
250 FOR N=1 TO 399
260 Y(0)=(X(0)+X(399)) MOD 2
270 PSET(0,N),Y(0)
280 FOR I=1 TO 399
290 Y(I)=(X(I)+X(I-1)) MOD 2
300 PSET(I,N),Y(I)
310 NEXT I
320 FOR I=0 TO 399
330 X(I)=Y(I)
340 NEXT I
350 NEXT N
```

The program cited above is a kind of cellular automaton; a model of self-reproducing machines.

The above BASIC program can be converted in Forth program:

```
( Y-X      B400H B000H 400H MOVE )
( AUTOS    400 1 do 1 B* B000H + DUP @ SWAP B- @ + 2 /
          DROP DUP 1 B* B400H + ! B800H @ 1 PSET loop )
( AUTO     B002H SET1 , 400 1 do B000H @ B31EH @ + 2 /
          DROP DUP B400H ! 1 DUP B800H ! 0 PSET AUTOS Y-X loop )
```

where Y-X corresponds to lines 320-340 in BASIC'S, AUTOS to lines 280-310 and AUTO to 250-350, including AUTOS and Y-X, respectively.

In Forth, each sentence, newly compiled task, is immediately executable and able to be, from its characteristic, subroutinized in proceeding sentence compilation; X-Y, AUTOS and AUTO can run quite independently. B400H and B000H are variable areas for X(I) and Y(I) in BASIC's, respectively, being represented directly in the absolute addresses in a hexadecimal expression.

The following lists the execution times in three kinds of programming language on NEC PC-9801 vm (10 MHz).

1.	BASIC	N88-original	17:21. (min:sec.XX)
2.	BASIC	on MS-DOS	18:38.
3.	BASIC	MS'-compiled	6:22.
4.	MS-Fortran		1:22.
5.	FORTH	listed	1:05.
6.	FORTH	tuned-up (1)	0:48.3
7.	FORTH	tuned-up (2)	0:28.6

In the tuned-up(1), separate primitive words are concatenated together, compound primitive words; "1 B*" to "1B*", " + DUP @ SWAP B- @ + " to " +D+ " and " 2 / DROP DUP " to "MOD2D", respectively. In the tuned-up(2), PSET in the BASIC's original subroutines is replaced by newly prepared one.

For real-number arithmetic, a numeric data processor, NDP, e.g., 8087/8/ for 8086, can effectively be incorporated in the system. The NDP has the architecture of 8 80-bit stack registers, which is thus the most suited to Forth's function, because Forth functionally operates through the stacks; pop-

ping data up and pushing data down. New primitive words for NDP's operation can, therefore, be prepared more directly than for CPU's. The intentional separation of function and representation among CPU's and NDP's may later facilitate programming, because the NDP's can be considered as one of I/O's like a graphics.

The following is an example, in which a circle area is calculated, getting a value of its radius in a floating-point number as an input. Then the subroutine-like usage follows for the sequential calculation.

```
R$ ( Circle_area FDUP R. SP SP FDUP; F* FLDPI$ F* D. CR )
R$ 1.24 Circle_area 3.456 Circle_area
1.2400 4.83051286
3.4560 37.52298159
R$RENAME Circle_area CRCL-A
R$ ( CRCL-SR do' i ITOR 0.2 F* 5.0 F+ CRCL-A loop )
R$ 10 0 CRCL-SR
5.0000 78.53981634
5.2000 84.94866545
5.4000 91.60884198
5.6000 98.52034593
```

The direct addressing range of 8086's is intrinsically limited to 64 kilo-bytes; this leads to the introduction of a segment. Thus the multi-segment programming will in its structure become resemble to that of multi-computer's, as in NODAL.

A PC-9801 vm provides 10 64-Kbyte segment blocks of RAMs with the expandability of an additional RAMs of upto 4 Mbytes in a form of bank selection. One block area seems ample enough for a resident monitor with a sufficient number of software tools, while remaining blocks can be appropriately allotted to application program area accompanied with data's. Each block assigned for application is of the same structure as the monitor's: except those of compilation and tools. Preliminary test of intersegment programming has yielded in a good result.

Discussion

The test of execution speed has been satisfactory. A tuning-up technique has also been proved to be effective as a kind of prototyping/9/.

For a distributed control system, a star-like network may be better than a loop's, because the system has, a priori, to be in a definite hierarchy.

References

- 1) Y. Kawarasaki: Proc. 1-st Workshop on Control Systems, KEK Report 87-25, p.106
- 2) Y. Kawarasaki: to be published in Annual Report, Dept. of Physics, JAERI
- 3) M. C. Crowley-Milling and G. C. Shering: CERN 78-07
- 4) Y. Kawarasaki: JAERI-M 86-112, p25
- 5) Y. Kawarasaki: Proc. 10-th LINAC MEETING, p.19
- 6) A trademark of FORTH Inc.
- 7) H. Takayasu: "Fractal", p.110, Asakura Publishing Co.
- 8) J. F. Palmer and S. P. Morse: "The 8087 Primer", John Wiley & Sons, Inc., New York
- 9) M. Arisawa: "Software Prototyping", Kindai-Kagaku Sha