SYMBOLIC PROCESSING POTENTIAL OF FORTH-BASED MICROCOMPUTERS*

H. G. Arnold
Energy Division
Oak Ridge National Labortory
Oak Ridge, TN 37831

ABSTRACT

The potential of microcomputers using Forth to perform Symbolic processing is evaluated using integers, strings, and rules in timing benchmarks. In addition to comparing 8086, 68000, and NC-4000 performance, some order-of-magnitude comparisons with popular mainframes are made. The conclusions reached are that rule processing speed in the neighborhood of 5000 rules per second can be achieved at the sacrifice of program complexity using a Forth engine microprocessor and that programming complexity can be preserved with the same processing speed now obtainable with mainframes (10 to 200 rules per second) by using Forth on a microcomputer.

INTRODUCTION

To define symbolic processing is roughly to say that it is not number crunching. Essentially it is the use of symbols, strings, or typographical notation to accomplish one's data processing needs. The more common examples of symbolic processing are in the arena of expert systems, and include the object-oriented programming, list processing and rule processing techniques that are the underlying theme of this conference. Although the subject of expert systems is important unto itself, it is also important to recognize that symbolic processing may require a different reference point... a different architecture, different goals, and consideration of a different end-user community. It is not necessarily true that the users of symbolic processing (expert systems?) are the same accountants and scientists who specified and developed the current numerical data processing architectures and models. It is more likely that the new audience for these types of programs are in the front office and are looking for what we call Decision Support Systems -- help in accumulating data and analyzing it for the purpose of more effectively making decisions.

PROCESSING GOALS

There have been several groups of researchers looking at Artificial Intelligence in computer systems. If one accepts that symbolic processing is the basis of writing AI programs, and that expert systems are the forerunners of AI programming, these experts have identified two needs that AI development tools must meet. First there must be a Very High Level Language (VHLL) to simplify the communication between

* Research performed at Oak Ridge National Laboratory, operated by Martin Marietta Energy Systems, Inc., for the U. S. Department of Energy under Contract No. DE-AC05-84OR21400.

computer science experts and experts in other areas of subject matter. Loffman[2] has described the characteristics of VHLL and shows that in addition to having complexity, it must also be understandable to humans. Second, expert systems must have high processing speeds. DARPA[3] has concluded that in order to meet the likely needs of expert systems a computer program must be capable of handling a rule base of 30,000 rules at a rate of 10,000 rules per second. This expert system will not be able to handle all types of decisions, but may be able to identify friend or foe in a real time situation. Present expert systems on existing mainframes can process about 200 rules per second (depending upon the definition of a rule). Therefore the estimates of these researchers is that the goal of 10,000 rules per second may be achievable by the early 1990's.

Research at Oak Ridge National Laboratory has shown that the potential for meeting this type of goal may not require the mainframe route. A version of OPS5 on a microcomputer[4] has run as fast as the same widely accepted language will run on popular minicomputers. This microcomputer version of OPS was written in Forth for a 68000 processor desk top computer. Extrapolations of this initial OPS performance to other environments indicates that the DARPA goals may not only be achievable, but may even be possible today in shoebox size computer systems. Benchmarks of a Forth Chip (a microprocessor using Forth as its machine language) in October of 1985 indicated that speeds as much as twenty times as fast as the 68000 were possible if the Forth Chip could support symbolic processing as well as it supports integer arithmetic. This set of early benchmarks led to the investigation of the potential for symbolic processing speed through other benchmarks which are summarized here.

BENCHMARKS

One rough measure of symbolic processing speed is the rate at which integer operations can be performed, i.e. simple DO-LOOPS. Figure 1 shows the results of running a million iterations on several types of computers in empty loops for which only the loop instructions were executed and in loops for which a 16-bit or 32-bit number was stored into memory. While the results are not to be interpreted as the actual speed of symbolic processing, they do offer some feel for the relative speed at which a rule represented by some type of pointer might be processed (as opposed to a rule represented by a string compared to another string). The overall conclusion is that 32-bit micros running Forth are more than twice as fast as 16-bit micros, maybe even as much as 3- to 4-times as fast.

Figure 2 shows the same benchmarks with the fastest 32-bit micro compared to the Forth Chip and to three large computers. Based on the results shown the Forth Chip should be expected to perform integer operations about 15 times as fast as a fast

---

2 Loffman, R. S. "A Survey of the Characteristics of Very High Level Languages", 1986 Rochester Forth Conference, University of Rochester, June 1986.

3 Defense Advanced Research Projects Agency, "STRATEGIC COMPUTING - New Generation Computing Technology: A Strategic Plan for its Development and Application to Critical Problems in Defense", Oct. 28, 1983.

4 Dress, W. B. "REAL-OPS - A Real-Time Engineering Applications Language for Writing Expert Systems", 1986 Rochester Forth Conference, University of Rochester, June 1986.

microcomputer. What is interesting to note is that the Chip can keep up with a VAX-780 when running empty loops and there are conditions using a special FOR-NEXT feature of the Chip under which it can almost keep up with one of the fastest mainframes made (a vectored machine running under conditions where vectoring does not come into play). In fact it is this inability to utilize the fast machine's architecture that illustrates the point that symbolic processing requires a different approach than floating point parallelism.

In an attempt to add the complexity of VHLL to the execution burden, a simple set of LISP instructions was written for the Forth Chip and compared to the same LISP instructions running on a micro-VAX and LMI LISP machine. Figure 3 shows these results for 100,000 iterations of a LISP do-loop that performed list processing within the loop. While the comparisons are not actually on the same basis since the Forth Chip automatically did "garbage collection" and the LISP machine takes "forever" to do it, the conclusion is that the Forth Chip can run LISP just about as fast as a LISP machine or a micro computer. This does not address the potential for optimization of the code for the Chip nor the problem of the tested prototype Chip in handling strings of bytes 40 times as slow as it should because of cell addressing.

A closer case to optimization was done with an inference engine for Forth called FORPS[5]. This expert system lacks the VHLL features by requiring Forth words in its rules, but clearly takes advantage of the Forth language in writing an expert system. The results of this system running on the chip are compared to other computers in Figure 4. The comparison is only incidental, however, since the results present an opportunity to calculate the speed of rule processing in an environment that may be recognizeable, the classic Towers of Hanoi problem solution. In a goal directed inference situation, the 68000 processor running Forth achieved speeds close to the 200 rules per second of mainframe machines (while minicomputers running a VHLL in the form of OPS could only do about 10 rules per second.) The Forth Chip however achieved processing times in the range of 4000 to 6000 rules per second in solving the Towers problem. It must be noted that there were only four rules and that OPS would handle large rule bases more efficiently, but the fact remains that such processing speeds are possible in small computers when the architecture of the machine, the design of the solution, and the type of the problem are compatible.

CONCLUSIONS

While it is not evident from these results that the problems of expert systems in particular and AI in general can be solved with Forth-based microcomputers the evidence exists that the potential is there. The inference engine that solved the Towers problem with the Forth Chip may appear to be irrelevant because of the small number of rules involved, but the rules were fired in a goal directed process. The Forth based microcomputer OPS solution equaled the OPS performance on high powered computers even though they could only achieve 10 or 12 rules per second with the equivalent four rules while the Forth Chip achieved 5000. The problem is that the Forth inference engine is unlikely to handle a base of 30,000 rules whereas OPS can handle the large rule bases very efficiently. The potential of OPS power at mainframe OPS speeds on microcomputers is therefore of equal interest to the amazing speed potential

---

[5] Matheus, C. J. "The Internals of FORPS (A FORth-based Production System)", In Publication, The Journal of Forth, 1986.

of a Forth inference engine. Future work at ORNL involves the application of the Forth Chip to real time expert systems, the translation of OPS to the Forth Chip to explore the capabilities of such a small microprocesser using a mature expert system shell, and building an application on a 68000 microcomputer using the OPS currently being developed in Forth. It is expected that these activities will demonstrate whether or not the symbolic processing requirements for expert systems can be achieved through the use of Forth on microcomputers.

CONTENTS OF LOOP / FORTH VERSION	Empty Loop	16-Bit Integer Store	32-Bit Integer Store
val Forth Atari 800	1:47.30	5:25.19	22:45
MVP-Forth IBM-XT	1:35.10	3:01.50	4:29.80
Forth-32 IBM-XT	1:08.55	4:04.55	4:13.25
MVP-Forth IBM-AT	0:35.85	1:07.25	1:46.10
Forth-32 IBM-AT	0:24.59	1:23.96	1:25.25
MacForth MacIntosh	0:19.10	1:06.46	1:06.31

```
* VARIABLE 1TEST  ( 16-Bit Integer Store)
  : THOUSAND 999 0 DO 1 1TEST ! ( W! ) LOOP ;
  : MILLION 999 0 DO THOUSAND LOOP ;
```

Figure 1. Comparison of Times to Perform One Million Iterations for Selected Microcomputers (0:00.0 Min:Sec)

CONTENTS OF LOOP / COMPUTER	Empty Loop	16-Bit Integer Store *	32-Bit Integer Store
MacForth Macintosh	0.19.10	1:06.46	1:06.31
Novix Forth	0:02.50 0:00.17 **	0:03.22 0:01.20	0:05.81 0:03.69
Cray-XMP (FORTRAN)	0:00.12		
IBM-3033 (FORTRAN)	0:00.41		
VAX-780 (FORTRAN)	0:02.112		

** FORTH algortihm to compare with mainframe times (based on special customized loop similar to optimized compilers)

Figure 2. Comparison of Times to Perform One Million Iterations for Selected Mainframes and Microcomputers (0:00.0 Min:Sec)

Garbage Collection	LISP MACHINE LM1	MINI COMPUTER Micro-VAX	FORTH CHIP Novix
(NO)	22	60	NA
(YES)	(15 Min?)	95-167	179

Figure 3. Time for 100,000 LISP Iterations (Seconds)

No. of Disks	OPS-5			FORTH INFERENCE	
	LISP MACHINE	VAX 780	68000 8 MHz	68000 10 MHz	NOVIX 6 MHz
5				0.29	0.01
7	22	60	26	1.15	0.06
10				9.25	0.42

(Seven disks required 256 rule firings for inference engine.)

Figure 4. Times for Towers of Hanoi Solution in Seconds for Selected Mini- and Microcomputers Running OPS and a Forth Inference Engine.