

THE E-MACHINE: SUPPORTING THE TEACHING OF
PROGRAM EXECUTION DYNAMICS

by

Samuel Dellette Patton

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

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

June 1989

APPROVAL

of a thesis submitted by

Samuel Dellette Patton

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.

Date

Chairperson, G

Approved for the Major Department

Date

Head, Major D

Approved for the College of Graduate Studies

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 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 _____

TABLE OF CONTENTS

	Page
LIST OF FIGURES	v
ABSTRACT	vii
1. INTRODUCTION	1
2. DYNAMOD	5
3. PROPOSED SYSTEM	13
4. REVIEW OF LITERATURE AND EXISTING SYSTEMS	18
5. THE E-MACHINE	25
Design Considerations	25
E-machine System Overview	30
E-machine Instruction Set	34
Addressing Modes	35
Instruction Set	36
Source Program Variable Representation in E-machine Code	42
The Save Stack	44
The Label Registers	51
Critical vs. Noncritical Instructions	59
6. COMPILING TO E-CODE	

61

7. CONCLUSIONS AND NEW DIRECTIONS
74

REFERENCES CITED
76

LIST OF FIGURES

Figure	Page
1. DYNAMOD Welcome Screen	6
2. DYNAMOD Library Screen	7
3. DYNAMOD Program <i>fact</i> Screen	8
4. DYNAMOD Program <i>fact</i> Screen After Partial Execution	9
5. Program <i>fact</i> Screen After Four Calls to Function <i>Factorial</i>	10
6. <i>for</i> Statement Display Window in a Syntax Directed Editor	15
7. Help Window for a Syntax Directed Editor	16
8. The E-machine	31
9. E-machine Global Variable Implementation	43
10. E-machine Recursive Variable Implementation	44
11. Variable and Save Stack for a Variable X	46
12. Variable and Save Stack After Assignment to X	47
13. A Pascal Procedure Fragment <i>something</i>	49
14. Variable and Save Stack During Successive Calls	

	to Procedure <i>Something</i>	50
15.	Simple E-code Program With a Branch	53
16.	Simple E-code Program with a Loop	56
17.	General Label Stack	56
18.	Label Stack After 0 Loop Iterations	57
19.	Label Stack After 1 Loop Iteration	58

Figure	Page
20. Label Stack After 2 Loop Iterations	58
21. E-code Translation of X := X + Y - 17 * Z * Z;	60
22. Example Pascal Program	63
23. Symbol Table For Pascal Program	65
24. Ad Hoc Packetizing Method for Pascal	66
25. Packetizing and Determination of Critical Instructions	68
26. Example Packetized Pascal Program	69
27. Example In-Line E-code Translation of a Packetized Pascal Program	70-72

ABSTRACT

The teaching of computer programming contains aspects that are found in few other disciplines. The mixture of syntactic and semantic knowledge required for programming is enough to overwhelm many beginning students. An earlier attempt at addressing this problem resulted in a software system called DYNAMOD. DYNAMOD is a library of expertly constructed Pascal programs and an interpreter that displays the execution of these programs on a computer video terminal in a step by step fashion under user control, effectively demonstrating the dynamics, or meaning, of a program. As a "proof of concept" system, DYNAMOD is quite successful, but it suffers from several limitations. In recognition of both the successes and limitations of DYNAMOD, it was decided to begin afresh and design an entirely new, much more ambitious system. The basis of this system is a virtual machine called the E-Machine which provides a very flexible structure for displaying the dynamic aspects of algorithms without distracting the student with syntactic details. The design of the E-machine is the primary work of this thesis. Important new features incorporated into the E-machine include, among others, internal structures which allow forward and backward execution of instructions and simple implementation of value and reference parameters. The E-machine will eventually form the basis of a comprehensive hypertext system for teaching and learning programming and other fundamental concepts of computer science.