

AN EVALUATION OF ML/I (EPS) MACROS  
FOR STRUCTURED FORTRAN EXTENSIONS

by

Soo Kyung Park

B.A., Ewha Womans University, 1968

A MASTER'S REPORT

Submitted in partial fulfillment of the  
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1977

Approved by:

  
Major Professor

LD  
2668  
R4  
1977  
P27  
C.2  
Document

TABLE OF CONTENTS

I. Introduction

A. Purpose of Paper ..... 1

B. Organization ..... 2

II. Macro Preprocessor for FORTRAN Extensions

A. Design Considerations ..... 4

B. Control Structures ..... 6

C. Use of the ML/I Macro Preprocessor ..... 10

III. Implementation

A. Introductory Features of ML/I ..... 13

B. Definition of Macros ..... 21

C. How to Execute the Preprocessor ..... 32

D. User Constraints ..... 37

IV. Evaluation

A. Problems of Macro Definition ..... 44

B. Time and Size of Macros ..... 56

C. Conclusions ..... 56

APPENDIX

A. Sample Structured FORTRAN program(TEST SFORT) . A- 1

B. Macro Listings(SOOMACRO EPS) ..... A- 3

C. Listing of S00 EXEC procedure ..... A- 5

D. Listing of a Sample EPSIN EPS ..... A- 7

E. Listings of Sample Structured FORTRAN output  
from EPS II ..... A-10

# **ILLEGIBLE DOCUMENT**

**THE FOLLOWING  
DOCUMENT(S) IS OF  
POOR LEGIBILITY IN  
THE ORIGINAL**

**THIS IS THE BEST  
COPY AVAILABLE**

## FIGURES

Figure II.1	For Use of ML/I Macro Preprocessor .....	11
Figure III.1	The Description of the File EPSIN EPS .....	34
Figure III.2	Files Processed by S00 EXEC Using EPS II ...	35
Figure IV.1	NEXT/QUIT Examples .....	46
Figure IV.2	IF...FI Examples .....	52
Figure IV.3	Examples for CASE...ENDCASE .....	54

## I. INTRODUCTION

### A. PURPOSE OF PAPER

The objective of this project is to investigate the use of a general purpose macro processor to implement structured programming extensions to FORTRAN.

The use of macro processors has developed since the 1950's. Using macro processors to modify or extend a compiler language has since occurred to many programmers. Text replacement has been the most distinguishing property of macro processors.

ML/I (Bro67) is a powerful macro processor for conveniently extending source languages. It has been used as a preprocessor to several compilers and assemblers. Its operation is to interpret macro definitions and translate input text to output text. Input to the macro processor is in the form of macro definitions and source text. The output text is derived from the source text by replacing all the macro calls that occur in it.

A version of ML/I, EPS II, is now available at KSU. This system is a load module written in IBM system/360 assembler language. EPS II can be run interactively under CMS.

The idea of introducing structured programming extensions of FORTRAN is not new. There has been a growing awareness of the need to improve unstructured FORTRAN since the 'structured programming' became an important issue in the computer world.

Many well-formed FORTRAN extensions have been defined by eliminating the GO TO and using only sequential statements and control structures in the form of block structures. By using ML/I, structured programming can be implemented to FORTRAN without changing over to a completely new compiler.

For this study, a set of macro extensions to FORTRAN will be defined and tested through EPS II. The main study area for structured programming will be focussed on IF THEN ELSE, CASE, WHILE, etc. Also, the efficiencies in terms of time using the macro processor will be investigated and evaluated.

## B. ORGANIZATION

The purpose of this paper has been briefly introduced at the beginning. The remainder of this paper is organized as follows. The second section presents the design impact of structured extension to FORTRAN. The extended control structures are described with the use of the macro preprocessor. The third section discusses the implementation procedures which could be viewed as the documentation for the user. Included is explanation of selected features of ML/I. All the macro definitions will be displayed and the process of execution will be shown. In the fourth section, use of ML/I will be evaluated. The problems in defining macros, timing and size, etc, will be topics of discussion. The possibility of further development for FORTRAN extensions using ML/I and future trends

will also be mentioned. The last part of this paper is an appendix which includes a list of macros defined and a test program listing.