UDC 681.322-181.4:681.3.06

SOFTWARE SUPPORT SYSTEM FOR MICROCOMPUTER

Tsuneyoshi Oboshi
Yasuyuki Kohara
Yoshiyuki Taguchi
Fumio Takahashi
Yusho Sato

I. FOREWORD

Today, many microcomputer application products are being developed. The functions of such products are sophisticated and complex. The difficulty of developing software for them is being given a close look. Fuji Electric tackled this problem from an early stage and announced the development of the microcomputer oriented software support tool MOGET (Microcomputer Oriented GEneral Tool) system in 1975.

The MOGET system is widely used as a uniform tool in FUJI MICREX-P/W/E development, including 1 chip microcomputer applied products, etc. However, to support development of more rational software to meet the appearance of the 32 bit microcomputer, expansion of address space, and other amazing hardware advances, the need for upgrading of the MOGET system and the development of a uniform support tool on the various host computer became high.

The N-MOGET (New-MOGET) system (see *Fig. 1*) developed as a support tool that is independent from the host computer and permits simple implementation of software support system for any microcomputer is introduced.

II. FEATURES OF N-MOGET SYSTEM

The features of the entire N-MOGET system are described here.

1. Portability of N-MOGET system

The N-MOGET can be run on an arbitrary host computer by means of a language processing technique called "metalanguage technology". (See *Fig. 2*)

The PANAFACOM U (PFU)-1000 series, FACOM M series, FACOM F230 series, are currently used as the host computer.

2. Adaptability to the target microcomputer

The N-MOGET system is basically a table controlled system. It can be used as a support tool for most microcomputers. (See *Fig. 3*)

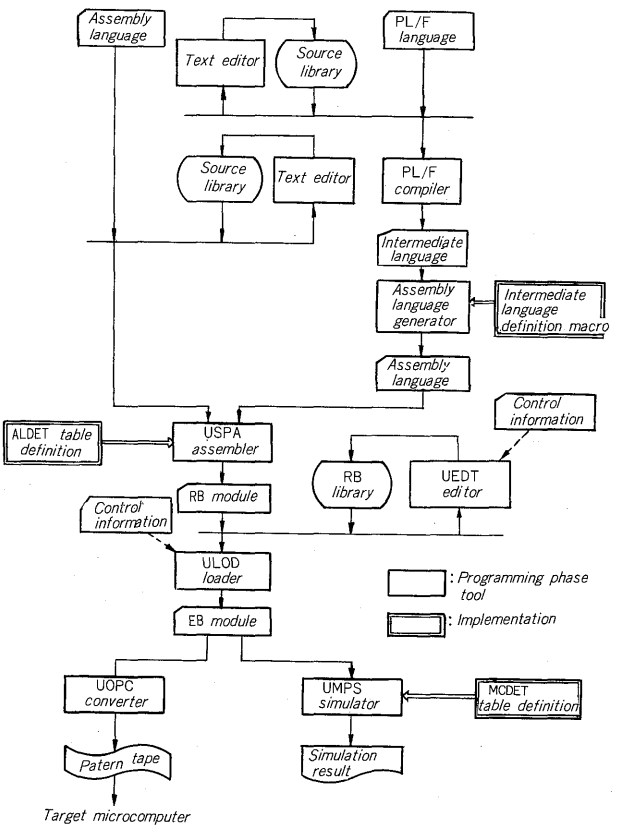


Fig. 1 Outline of N-MOGET system

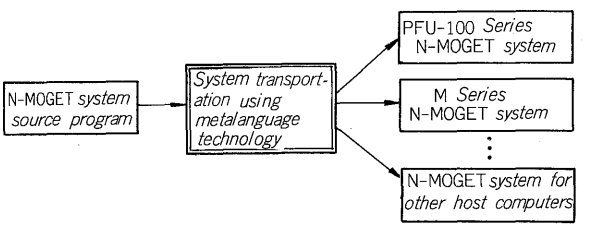


Fig. 2 Portability of N-MOGET system

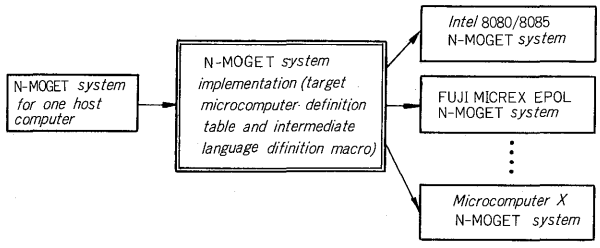


Fig. 3 Implementation of N-MOGET system

Therefore, the user does not have to learn a new support tool interface even when the target microcomputer is changed. This feature is continued from the MOGET system. However, the higher performance of the microcomputers has been accompanied by an following improvements:

1) The system can be applied to microcomputer of up to 32 bits/word.
2) Program linking has been simplified by making the assembler output relocatable.

3. Portability of application programs

The N-MOGET system is available with a high level language called PL/F (PL/I subset) in addition to assembly language. Application programs can be run on an arbitary microcomputer by writing the application program in PL/F language. (See *Fig. 4*.)

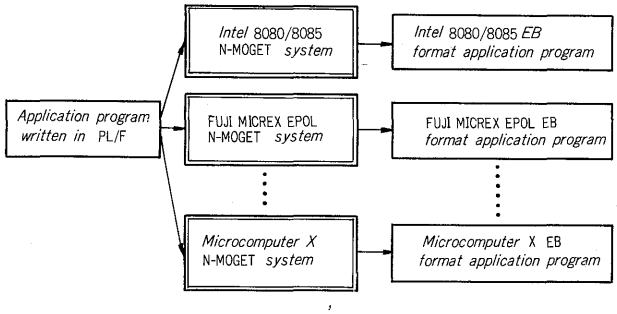


Fig. 4 Programming phase based on PL/F

4. Application program RB library

A user who develops application programs by using the N-MOGET system can manage the programs as an RB library without regard to whether the source language is PL/F or assembly language. Software development efficiency can be increased. (See *Fig. 6*)

III. OUTLINE OF N-MOGET SYSTEM

Fig. 1 outlines the N-MOGET system.
The N-MOGET system has the following three phases:

1. System transportation phase

This phase generates the N-MOGET system for the host computer from the N-MOGET system source program so that the N-MOGET system can be run on an aribtrary host computer. (See *Fig. 2*)

All the software forming the N-MOGET system are written in a system implementation language that guarantees the portability of the object program. Programs written in this system implementation language can be converted into programs written in an arbitary computer assembly language by a string of language processing. Therefore, all the software forming the N-MOGET system can be converted to programs written in the assembly language of an arbitrary computer. N-MOGET system that can run on a certain computer can be generated by processing the assembly language format program of the converted result with the assembler and editor available with the computer. That computer becomes the N-MOGET system host computer.

The following software are produced at the system transportation phase. (See section IV.)

1) PL/F compiler
2) Assembly language generator
3) UEDT editor
4) ULOD loader
5) UTEC1 table processor
6) UTEC2 table processor
7) UOPC object program converter
8) USPA assembler
9) UMPS simulator

2. Implementation phase

This phase generates the N-MOGET system for an arbitrary microcomputer. (See *Fig. 3*)

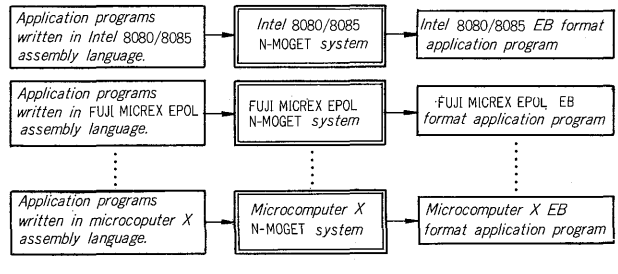


Fig. 5 Programming phase based on assembly language

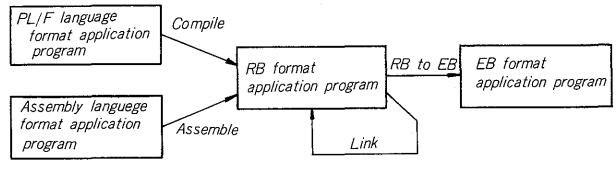


Fig. 6 Linkage of programs

The following tables must be produced at the implementation phase. These tables must be produced for each target microcomputer.

1) Assembly language definition table (ALDET table)
2) Microcomputer definition table (MCDET table)
3) Intermediate language definition macro

3. Programming phase

This phase generates the application programs for an arbitrary target microcomputer by using the microcomputer oriented N-MOGET system created through the implementation phase. (See *Figs. 4* and *5*)

Fig. 4 shows the programming phase based on PL/F. *Fig. 5* shows the programming phase based on assembly language. In case of *Fig. 5*, portability of the application program is lost. However, application programs can be linked without regard to whether the application program notation language is PL/F or assembly language as shown in *Fig. 6*.

IV. OUTLINE OF PROCESSING AND FEATURES OF SOFTWARE

The outline of processing and the features of the software in the N-MOGET system shown in *Fig. 1* are described.

The software not shown in *Fig. 1* are described hear. These software are used at implementation phase.

1. UTEC1 table processor

UTEC1 (Universal TExt Converter 1) is used when implementing the USPA assembler. It is a table processor that analyzes the table called ALDET (Assembly Language DEfinition Table) that defines the language format of the assembler source program that is input to the USPA assembler. This ALDET table must be defined for each microcomputer and has the following features:

1) The assembly language format can be freely defined by calling basic subroutines, called semantic routines, provided by USPA module.
2) The machine code may have variable bit length and any configuration can be used.
3) The printing format of address and machine code can be freely decided.
4) A knowledge of the machine code of the target microcomputer permits simple definition of this table.

2. USPA assembler

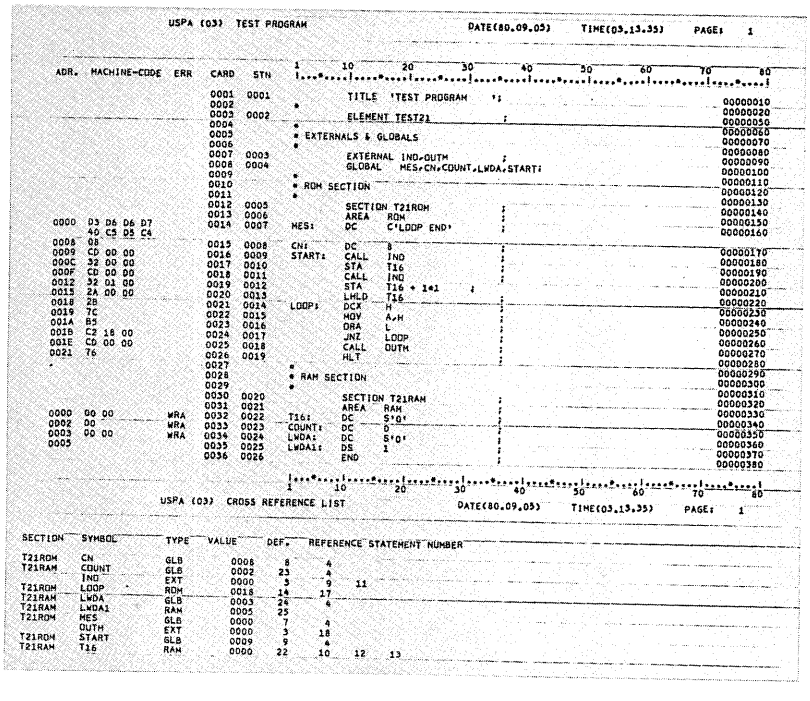
The USPA (Universal Symbolic Program Assembler) assembler is an assembler for an arbitrary microcomputer that is generated through USPA implementation.

It has the following main features:

1) RB module is output.
2) Since assembling is performed by a table controlled system, the assembly language format can be freely selected for each microcomputer.
3) Abundant constant definitions are available.
4) EBCDIC or ASCII code can be selected for character string data.
5) One instruction can be represented in free-form at several cards.
6) Cross reference list is output.
7) Continuous assembly is possible.
8) Since divided assembly is possible, only the programs to be correct should be assembled.

3. UEDT editor

The UEDT (Universal linkage EDiTor) produces one RB module by linking arbitrary number of RB modules output by the USPA assembler and also outputs a link map



```
USPA (03)  TEST PROGRAM                                   DATE(80.09.05)   TIME(03.13.35)   PAGE:  1

ADR.  MACHINE-CODE  ERR   CARD  STN
                          0001  0001          TITLE  'TEST PROGRAM    ';                    00000010
                          0002                *                                           00000020
                          0003  0002          ELEMENT TEST21          ;                    00000030
                          0004                *                                           00000050
                          0005                * EXTERNALS & GLOBALS                       00000060
                          0006                *                                           00000070
                          0007  0003          EXTERNAL IND,OUTH       ;                    00000080
                          0008  0004          GLOBAL   MES,CN,COUNT,LWDA,START;            00000090
                          0009                *                                           00000100
                          0010                * ROM SECTION                                00000110
                          0011                *                                           00000120
                          0012  0005          SECTION T21ROM          ;                    00000130
                          0013  0006          AREA   ROM              ;                    00000140
0000  D3 D6 D6 D7         0014  0007  MES:     DC     C'LOOP END'      ;                    00000150
      40 C5 D5 C4                                                                          00000160
0008  08                 0015  0008  CN:      DC     8                ;                    00000170
0009  CD 00 00           0016  0009  START:   CALL   IND              ;                    00000180
000C  32 00 00           0017  0010          STA    T16              ;                    00000190
000F  CD 00 00           0018  0011          CALL   IND              ;                    00000200
0012  32 01 00           0019  0012          STA    T16 + 1#1        ;                    00000210
0015  2A 00 00           0020  0013          LHLD   T16              ;                    00000220
0018  2B                 0021  0014  LOOP:   DCX    H                ;                    00000230
0019  7C                 0022  0015          MOV    A,H              ;                    00000240
001A  B5                 0023  0016          ORA    L                ;                    00000250
001B  C2 18 00           0024  0017          JNZ    LOOP             ;                    00000260
001E  CD 00 00           0025  0018          CALL   OUTH             ;                    00000270
0021  76                 0026  0019          HLT                     ;                    00000280
                          0027                *                                           00000290
                          0028                * RAM SECTION                               00000300
                          0029                *                                           00000310
                          0030  0020          SECTION T21RAM          ;                    00000320
                          0031  0021          AREA   RAM              ;                    00000330
0000  00 00         WRA    0032  0022  T16:     DC     5'0'             ;                    00000340
0002  00           WRA    0033  0023  COUNT:   DC     0                ;                    00000350
0003  00 00         WRA    0034  0024  LWDA:    DC     5'0'             ;                    00000360
0005                      0035  0025  LWDA1:   DS     1                ;                    00000370
                          0036  0026          END                     ;                    00000380

USPA (03)  CROSS REFERENCE LIST                           DATE(80.09.05)   TIME(03.13.35)   PAGE:  1

SECTION   SYMBOL   TYPE   VALUE   DEF.   REFERENCE STATEMENT NUMBER
T21ROM    CN       GLB    0008    8      4
T21RAM    COUNT    GLB    0002    23     4
          IND      EXT    0000    3      9    11
T21ROM    LOOP     ROM    0018    14     17
T21RAM    LWDA     GLB    0003    24     4
T21RAM    LWDA1    RAM    0005    25
T21ROM    MES      GLB    0000    7      4
          OUTH     EXT    0000    3      18
T21ROM    START    GLB    0009    9      4
T21RAM    T16      RAM    0000    22     10   12   13
```

Fig. 7 Example of output of USPA assembler

list. Since several common subroutines can be linked to one RB module by using UEDT, RB library management can be simplified.

4. ULOD loader

The ULOD (Universal linkage LOaDer) produces one EB module by linking an arbitrary number of RB modules output by the USPA assembler or UEDT editor and outputs a link map list.

5. UOPC object converter

The UOPC (Universal Object Program Converter) edits the EB module output by ULOD in accordance with input format of PROM programmer, etc. and has the following main features.

1) Write pattern list is output.
2) Parity bit pattern list is output.
3) Since the program name, serial No., programmer name, address, etc. are output in dotted character at the pattern tape, pattern tape management is simple.
4) PROM chip size, positive/negative logic, even/odd parity, and abundant other parameters are available.

6. UTEC2 table processor

The UTEC2 (Universal TExt Converter 2) is used when implementing UMPS.

It is a table processor that analyzes the table called MCDET (MicroComputer DEfinition Table). This table defines operations of target microcomputer's instructions. This MCDET table must be defined for each microcomputer and has the following main features:

1) Operation of each instruction can be easily defined by calling basic subroutines called the arithmetic/logical operation routines and flag opeation routines that are provided by the UMPS module.
2) Memory and address can be any bit length.

7. UMPS simulator

The UMPS (Universal Microcomputer Program Simulator) is a simulator for arbitrary microcomputers that is generated through UMPS implementation.

It has the following main features:

1) Interactive simulation is possible.
2) External names can be referenced by name. Local names and fetch address can be referenced by relative address.
3) Numerous debugging commands, register contents trace, I/O data contents desplay, run-time account, etc., are available.

8. PL/F compiler

The PL/F language is a high level programming language based on PL/I. The PL/F compiler compiles application programs written in the PL/F language. Use of the PL/F language offers the following advantages.

1) Top-down programming and structured programming are possible.
2) Programming efficiency is excellent.
3) Understandability of programs is improved.
4) Reliability of programs is improved.
5) Application programs can be developped without regard to the target microcomputer.

9. Assembly language generator

The assembly language generator is used when transporting the N-MOGET system on an arbitrary computer and when translating the intermediate language output by the PL/F compiler into the target microcomputer's assembly language.

V. CONCLUSION

The N-MOGET system was outlined above.

The assembly language generator is used when transporting the. N-MOGET system on an arbitrary computer and when translating the intermediate language output by the PL/F compiler into the target microcomputer's assembly.

Reference:

M. Nonaka, J. Hiramatsu: A General Purpose Software Support System for Microprocessors, Euromicro Symposium, Microprocessing and Microprogramming (Oct. 1976)