

ISTITUTO NAZIONALE DI FISICA NUCLEARE
Laboratori Nazionali di Frascati

LNF-82/28(NT)
22 Aprile 1982

M. L. Ferrer:

CROSS-ASSEMBLER FOR TEXAS TMS 9900 MICROPROCESSORS

M. L. Ferrer: CROSS-ASSEMBLER FOR TEXAS TMS 9900 MICROPROCESSORS.

A cross-assembler in BCPL language has been produced for VAX 11/780 (VMS V 2.4 operating system), by adapting a program originally written for IBM 370. The main features of this product are given and examples of its use are presented.

An extensive development of software oriented to produce machine code for microprocessors has been realized during the past five years at CERN⁽¹⁾. A program written in standard assembler language for a microprocessor (Intel 8080; Motorola 6800, 6801 and 6800; Texas Instruments 9900) can be edited, assembled and prepared for loading using host main computers such as IBM 370, CDC 6600 and NORD 10. The machine code produced is then loaded in the microcomputer memory using direct link with the host when it exists.

A system based on the microprocessor Texas TMS 9900 for CAMAC acquisition (CANDI) is actually running in the Laboratori Nazionali di Frascati (INFN)⁽²⁻⁶⁾, linked with a PDP 11/34 or VAX 11/780. No peripheral for mass storage is interfaced to CANDI because it uses the PDP or VAX disks as data and program support.

In order to implement the cross assembler on the VAX 11/780, the IBM version was used (32 bits as the VAX machine). The only important difference found was respect to the input/output definitions that were completely rewritten.

In the following, the procedure to use the cross-compiler is given.

Two files can be produced by the user as input.

- 1) An optional file (OPTIONS.DAT) containing listing and assembler options, one for each line, used to modify the default values of some parameters. The possible options are⁽⁷⁾:

ERROR LIMIT = n (default n = 50)

where n gives the maximum number of errors which can be detected before the assembly aborts.

FULL LISTING

indicating that each generated byte must be displayed on the assembly list.

LINE LENGTH = n (default n = 80)

where n gives the maximum number of characters that a source line may have.

TABLE MAGNITUDE = n (default n = 9)

where 2**n gives the maximum number of symbols that the symbol directory of the assembler can contain.

- 2) Source file containing the assembler language to be interpreted as described in the Assembly Language Programmer's Guide from TEXAS INSTRUMENTS. This source must define the absolute start address on the microprocessor memory.

The cross-assembler produces :

- 1) A file (CUFOM.DAT) containing in CUFOM format⁽⁸⁾ the object code. This file contains also information useful to run programs as linked and librarian produced at CERN and here not implemented. If the assembled program does not contain any external references, it can be given as input to a program "TEXAS.EXE" that produces a binary file containing only the physical microprocessor memory map. A program, "SERIAL", running in CANDI, activating another task on VAX, "CLERKS.EXE", transfers this binary file into the micro-computer memory.
- 2) A listing on SYS\$OUTPUT (terminal for interactive work of paper on batch mode) containing in the order: instruction number, object (in one or more lines) in exadecimale code, assembled instruction and comments. The assembler detects only one error per line. In case the error occurs, a short error message follows the line.

How to run the assembler on VAX 11/780.

The assembler (ASM.EXE) and the BCPL compiler (BCPL.EXE) are contained in the SYS\$DISK:[BCPL] directory. The BCPL language cannot read parameters appended to the program name. For that, a procedure ASM.COM exists on the same directory, that will be called by user. Also a global symbol ASM has been defined

```
$ ASM == @SYS$DISK:[BCPL]ASM.COM
```

If the file to be assembled is for example SERIAL.DAT, the command from the user must be :

```
$ ASM SERIAL.DAT
```

The procedure ASM.COM here listed

```
$ ASSIGN/USER_MODE SYS$COMMAND: SYS$INPUT:
```

```
$ IF P2.EQS. '' THEN GOTO FILELIST
```

```
$ ASSIGN/USER_MODE 'P2 SYS$OUTPUT:
```

```
$ FILELIST:
```

```
$ IF P1.EQS.' ' THEN GOTO RUNASM
```

```
$ OPEN/WRITE INPUT_FILE ASM.RUN
```

```
$ WRITE INPUT_FILE P1
```

```
$ CLOSE INPUT_FILE
```

```
$ RUNASM:
```

```
$ RUN [BCPL] ASM
```

```
$ SET NOVERIFY
```

will create a file ASM.RUN in the user's directory. After that, the assembler runs reading the file ASM.RUN in order to found input. The file ASM.RUN remains until deletion by the user. If a new assembly of the same file is requested, only the command:

```
$ ASM
```

is necessary.

In order to run the cross-assembler in batch mode, the user must create a command file containing the instruction

```
$ ASM SERIAL.DAT
```

In order to produces all listing on the file SERIAL.LIS, for example, the instruction must been:

```
$ ASM SERIAL.DAT SERIAL.LIS
```

In the following are given:

1) Source file as produced by editor

```
*****
```

```
** test on cross- **
```

```
** assembler for **
```

```
** TMS 9900 **
```

```
*****
```

```
START AORG >1000
```

```
TEXT 'THIS IS A TEST'
```

```
END START
```

2) Assembly listing without OPTIONS.DAT file

```
PAGE 1 27-MARZO-1982 12:50:53
```

```
1 *****
```

```
2 ** test on cross- **
```

```
3 ** assembler for **
```

```
4 ** TMS 9900 **
```

```
5 *****
```

```
6 1000          START AORG >1000
7 1000    54          TEXT  'THIS IS A TEST'
8 100E 1000          END START
```

..... 0 ERROR DETECTED IN THE SOURCE FILE <<<<<<<<

3) File OPTIONS.DAT

FULL LISTING

4) Assembly listing after creating the OPTIONS.DAT file

PAGE 1 27-MARZO-1982 12:47:38

THE OPTIONS ARE:

1 FULL LISTING

PAGE 2 27-MARZO-1982 12:47:40

1 *****

2 ** test on cross- **

3 ** assembler for **

4 ** TMS 9900 **

5 *****

6 1000 START AORG >1000

7 1000 54 TEXT 'THIS IS A TEST'

1001 48

1002 49

1003 53

1004 20

1005 49

1006 53

1007 20

1008 41

1009 20

100A 54

100B 45

100C 53

100D 54

8 100E 1000 END START

..... 0 ERROR DETECTED IN THE SOURCE FILE <<<<<<<<

I wish to thank Dr. L. Trasatti for continuing incouragement.

REFERENCES.

- (1) - H. Von Eicken et al., Cross software for microprocessor program development at CERN, CERN-Data Handling Division DD/81/10 (1981).
- (2) - O. Ciaffoni et al., A CAMAC system controller using the TEXAS TMS 9900 microprocessor as stand-alone and PDP 11 connected unit, Frascati Report LNF-80/27 (1980).
- (3) - O. Ciaffoni et al., Il nodo intelligente di acquisizione dati da CAMAC "CANDI", Frascati Report LNF-81/25 (1981).
- (4) - O. Ciaffoni et al., Data acquisition system for cosmic ray muon background tests under the Gran Sasso tunnel, Frascati Report LNF-81/36 (1981).
- (5) - O. Ciaffoni et al., CANDI, a microprocessor based CAMAC acquisition system with distributed intelligence features, Proceedings of the Summer School on 'Data Acquisition for High Energy Physics', Varenna (1981).
- (6) - O. Ciaffoni et al., CANDI USER'S GUIDE, Frascati Report LNF-81/65 (1981).
- (7) - J. Montuelle, The TMS 9900 Cross Assembler user manual, CERN write-up TMS 9900, (1981).
- (8) - J. Montuelle, CUFOM, the CERN universal format for object modules, CERN-Data Handling Division DD/78/21 (1978).