TASMON4 ADDENDUM - VERSION 1.11 (10/15/84)

The following changes should be made to the TASMON4 manual to keep it current with TASMON4 version 1.11 and INSTALL version 2.0:

Page 1 - INSTALLING TASMON4

Mark S. Barlow has made some changes to the INSTALL program in order to make it compatible with TRSDOS 6.2. INSTALL version 2.0 will now correctly install TASMON4 onto a disk formatted under either TRSDOS 6.1 or TRSDOS 6.2.

1) The files that come on your master TASMON4 disk are listed below with their function:

INSTALL/CMD - The program that will make SYS13/OBJ into a system file.

SYS13/OBJ - The overlay code that will become a system file.

TASMON4/CMD - This is the TASMON4 /CMD file - note that this file uses the SYS13/SYS file, which must be installed first!

TEST/CMD - A demonstration program.

2) INSTALL/CMD and SYS13/OBJ must be in ANY disk drive accessible by the system during the installation process (it is no longer a requirement that these files be on the disk in drive zero).

3) At the completion of the installation process, both INSTALL/CMD and SYS13/OBJ will be REMOVEd (killed) from the disk. Therefore, the proper procedure for installation is as follows:

a. COPY the following files from your TASMON4 master disk to any TRSDOS 6 system disk: TASMON4/CMD, INSTALL/CMD, and SYS13/OBJ. Then take your TASMON4 master disk out of the disk drive and put it away in a safe place.

b. Execute the INSTALL program (type INSTALL and press the ENTER key). More specific instructions for the particular version of TRSDOS 6 that you are using will be found below.

c. When the installation has been successfully completed, both TASMON4/CMD and a new SYS13/SYS file will remain on the disk.

It is important to note that in order to make copies of TASMON4, it will be necessary to either repeat the above procedure, or make a BACKUP of the entire system disk on which TASMON4 resides. TASMON4 will not work if you simply copy the TASMON4/CMD file over to another disk; it needs to have the SYS13/SYS file created by the INSTALL program present on the disk as well.

4) If INSTALL aborts back to TRSDOS with ANY sort of DOS error, it may be necessary to close some of the files that were opened during the unsuccessful installation process. This can be accomplished by using the RESET command in the form: RESET filename . Files that have been left open can be easily identified because they will have a question mark after their filename in the directory listing.

5) INSTALL will not work properly if the password protection of SYS13/SYS has been removed through the use of Super Utility Plus or a similar product. Either start again with a fresh backup of TRSDOS 6, or re-protect SYS13/SYS with the password "LSIDOS".

Installing TASMON4 using INSTALL version 2.0
INSTALLING on a TRSDOS 6.1 disk:

Type INSTALL . The screen will clear and the following message will be displayed:

INSTALL - TASMON4 installation program.
Written by Paul F. Snively, Version 2.0
Copyright (c) 1984 by The Alternate Source.

After a few seconds, if everything goes OK the following message will be displayed:

Installing TASMON4.
TASMON4 has been installed. Make BACKUPS from this working master.

TASMON4 has now been successfully installed on your disk. The only files that remain on your disk will be TASMON4 and SYS13/SYS. Note that the SYS13/SYS file previously existed but contained nothing; it now contains part of TASMON4.

However, if the following message appears:

SYS13/SYS is occupied on your TRSDOS 6.1 disk, CANNOT install TASMON4.

TASMON4 has not been installed and is not functional. The only way this error could occur is if TASMON4 has already been installed on the disk or another program is using the SYS13/SYS slot (neither error is likely). In order to correct the problem, use a fresh backup of your TRSDOS 6 system disk.

INSTALLING on a TRSDOS 6.2 disk:

Type INSTALL . The screen will clear and the following message will be displayed:

INSTALL - TASMON4 installation program.
Written by Paul F. Snively, Version 2.0
Copyright (c) 1984 by The Alternate Source.

After a few seconds, if everything goes OK the following message will be displayed:

Installing TASMON4.
TASMON4 has been installed. Make BACKUPS from this working master.

TASMON4 has now been successfully installed on your disk. The only files that remain on your disk will be TASMON4 and SYS13/SYS. Note that the SYS13/SYS file previously existed but contained nothing; it now contains part of TASMON4.

However, if the following message appears:

An ECI is already installed, cannot install TASMON4.

TASMON4 was not installed because an ECI exits on that disk. ECI stands for Extended Command Interpreter, and this might be present if another applications program is using the SYS13/SYS file. To correct the problem, use another TRSDOS 6 system disk that doesn't have an ECI. Note that this error should rarely or never be encountered.

If the following message appears:

SYS13/SYS occupied on your TRSDOS 6.2 disk.

This simply means that the SYS13/SYS file was not an ECI, but did appear to contain some type of data (even if only "garbage" data). This will be the case when a fresh copy of TRSDOS 6.2 is used. Don't be alarmed; TASMON4 will be installed correctly and will be ready to use when the installation process is complete.

Page 6 - DISASSEMBLED DUMP

The equivalent ASCII codes for hex bytes that fall in the range 20H-7FH are now displayed in a field between the hex byte display and the opcode display fields. This makes it easy to pick out text fields during a disassembly. The new display format is active while disassembling to screen or printer (using the "D" or "P" commands), but NOT while disassembling to a disk file.

Also, many of the so-called "undocumented" opcodes of the Z-80 microprocessor will now disassemble properly. An explanation of

these "undocumented" codes follows:

The SLS opcode is similar to other shift-type instructions (such as SLA, SRA, and SRL). SLS stands for "Shift Left and Set bit 0", which in effect means that the contents of the affected register are multiplied by two, and one in then added. This opcode is formed by the two-byte combination CB nn, where nn is any value from 30H through 37H.

The operands HX, LX, HY and LY are used to access one-half of the IX or IY register pairs. HX, for example, indicates the high (most significant) byte if the IX register pair, where LX is the low (least significant) byte. Instructions using these operands are formed by taking an instruction that normally accesses the H or L register, and adding a prefix of DDH (to affect the IX register pair) or FDH (to affect IY). For example, the instruction INC H is a single-byte instruction represented by a 24H byte. The bytes DD 24 would yield a INC HX instruction, while FD 24 would yield an INC HY instruction.

Certain ROTATE and SHIFT type instructions can affect both a memory location and a register. A typical opcode for such an instruction might be RLC B,(IX+nn) which would mean that that a rotate left circular operation is performed on the byte pointed to by (IX+nn), and then the contents of that memory location are copied into the B register! These four-byte opcodes are formed by the combination DD CB mm nn, where nn is a value from 00H through 3FH (note that some documented instructions also fall in this range), and mm is the offset (index) byte.

A similar effect can be observed with the SET and RESET instructions. The pattern is DD CB mm nn as above, except that nn is a value in the range 80H through BFH (for RES), or C0H through FFH (for SET). A typical opcode might be RES 0,B,(IX+nn) which would mean that bit 0 of memory location (IX+nn) is reset, and then the contents of that memory location are copied into the B register.

For input/output port users, there is a two-byte instruction created by the bytes ED 70 that reads the port specified by the C register, but does not put the result in any register. However, it DOES set the flags according to the result, similar to an instruction like IN A,(C) , IN B,(C) etc. TASMON uses the mnemonic IN -,(C) to represent this opcode. Also, the two-byte instruction ED 71 will output a zero byte to the port specified by the C register. TASMON uses OUT (C),0 to represent this one.

TASMON's disassembly routine will decode all of the above-mentioned "undocumented" opcodes in any of its disassembly functions

Page 15 - SINGLE STEP and TRACE commands

During these operations, the current PC address and the Z-80 mnemonic for each instruction executed will be displayed on TASMON's display

lines. Up to fifteen previously executed instructions can be displayed at one time.

Page 17 - KEEP SCREEN

Tasmon uses from column 48 to the right-hand side of the video display for its displays while tracing and single stepping (not just columns 56-72 as stated).

Page 37 - TASMON COMMAND SUMMARY

A new command has been added to TASMON. The "Q" command will compare two blocks of memory and report the first occurrence of any differences. For example, if you type:

Q A000 B000

The blocks of memory starting at A000 and B000 will be compared byte by byte. If the display then reads:

Q A000 B000 A154 B154

It would mean that the bytes stored in locations A154H and B154H do NOT match, but that the bytes previously compared did match. At this point, you could just type:

Q

(with no arguments) to resume the compare at A155H compared to B155H. In this case the comparison resumes at the addresses following the mismatched bytes.

If you enter the same address for both addresses called for by the "Q" command, there will be no effect since to attempt comparing an address to "itself" would possibly result in an "endless loop" type system lockup!

PAGE 39 - SAMPLE USER FUNCTIONS

TASMON now displays the last 15 instructions executed, so that function is no longer needed as part of the user function. To enable HARD COPY TRACING, see the revised Appendix C (below).

APPENDIX C - SAMPLE USER FUNCTION (REVISED 9/20/84)

This appendix will give an example of patching in a USER command (the "U" command). This routine will allow HARD COPY TRACING. This is the same as normal tracing except the current PC address and Z-80 mnemonic are sent to the printer. If the printer is not on when HARD COPY TRACE is selected, nothing is printed and execution continues as if the TRACE command had been selected.

This patched routine assumes that TASMON4 version 1.11 is being used. Also, TASMON should be located in memory starting at E000H. Start entering bytes at FE7BH by using TASMON's M command - type:

```
M H FE7B
```

and enter the following bytes from there:

```
FE7B:  00 00 CD A2 E9 32 E1 FE C3 63 E8 AF 32 E1 FE C3
FE8B:  3E E0 3A E1 FE B7 CA 08 E1 DB F8 E6 F0 FE 30 C2
FE9B:  08 E1 2A 1E F9 7C CD C9 FE 7D CD C9 FE 3E 20 CD
FEAB:  7F F8 CD 08 E1 06 16 21 30 00 E5 C5 06 01 3E 0F
FEBB:  EF CD 7F F8 C1 E1 23 10 F1 3E 0D C3 7F F8 F5 CB
FECB:  3F CB 3F CB 3F CB 3F CD D8 FE F1 E6 0F C6 90 27
FEDB:  CE 40 27 C3 7F F8 00
```

Then modify the following memory locations in TASMON, again using the M command:

Starting at E0ECH, enter 7D FE.
Starting at E755H, enter 8D FE.

NOTE: Do NOT hit the BREAK key to exit from this last memory modification (below) until both bytes have been correctly entered. If you attempt to use BREAK to exit the memory modification mode prematurely, your computer may "lock up" or be reset!

Starting at E106H, enter 86 FE.

To write the patched version of TASMON out under the file name "UPTASMON/CMD", enter the following command:

```
W E000 FFFF E000
UPTASMON/CMD <ENTER>
```

To execute the "HARD COPY TRACE", press the "U" key. Then enter the CALL stepping mode by pressing either "I" for CALLs stepped through, or a DOWN-ARROW for CALLs executed in full. TASMON will step through memory as it would with the normal TRACE command, except the following type output will be sent to the printer:

```
8000  LD      A,(37E8)
```

All TRACE command keys function with the "HARD COPY TRACE" patch.