flags were inadvertently changed. You have to expand macros before using XLT86 unless the number of invocations was small. In that event, expanding the macros by hand with an editor might be just as easy.

Sorcim's TRANS86

TRANS86 is an 8080/ACT80-to-8086 translator. It takes 8080 or ACT80 source code as input and creates a file compatible with the input to ACT86, an 8086 assembler.

The output of TRANS86 is incompatible with any assembler other than ACT86. The ACT86 mnemonics are different enough so that, unless the programmer has a sophisticated text processor and the talent (or patience) to do a great deal of text manipulation, TRANS86 should be used only with ACT86.

Both TRANS86 and ACT86 run on either 8080 or Z80 processors under CP/M-80, MP/M, or CDOS with a minimum of 24K bytes of RAM (random-access read/write memory). TRANS86 consists of an executable file, an overlay file, and a translation table. The input assembly source code must be in a form acceptable to the standard CP/M-80 assemblers (ASM, MAC, or RMAC), or to ACT80, Sorcim's Assembly Code Translator for 8080/Z80 processors.

Translation occurs on an instruction-by-instruction basis with some optimization rules applied to conditional jumps. There appears to be no limit as to the size of the source file that can be translated. A file is produced on the same disk as the source file with the same name and an .ASN extension. If a file by that name already exists, the user is asked if the file should be deleted or if the program should be aborted.

TRANS86 flags the following Z80 instructions as errors:

ACT80 code	Zilog/Mostek equivalent code
Mov A,R	Ld A,R
Mov R,A	Ld R,A
Mov A,I	Ld A,I
Mov I,A	Ld I,A
In,C reg	In reg,(C)
Inir	Inir
Otir	Otir
Rld	Rld
Rrd	Rrd

TRANS86 supports macros in the ACT80 format. Although TRANS86 acts on macros in the MAC format, there is no guarantee that the macros will expand correctly. The user is cautioned to examine the result of a macro expansion to determine if the sense of the macro has been maintained. Examples are given of some macros that work and some that do not. The *TRANS86 User's Reference Manual* includes a section that gives hints on how to hand-optimize the output of TRANS86; specifically, accumulator indirect loads through the DE and BC registers, 8080 conditional jumps, and Z80 block instructions.

Another section describes the differences between ACT80 and Z80 mnemonics. This information allows the programmer to manually convert assembly source code to a form acceptable to TRANS86. The ACT80 instruction set has some ASM-style instructions, some Z80-style instructions, and some instructions that are unique to the ACT80 assembler. If the source code is written in 8080 ASM mnemonics, TRANS86 will process it and output ACT86 assembler code. The 8080 instruction SPHL, however, was translated incorrectly in the current version of TRANS86.

Another section in the manual contains suggested constructions that can be manually entered to deal with Z80 op codes that are flagged as errors. Block input/output instructions and input/output through register C are described in detail.