Listing 1 continued:

```
Cnz Help         Jz :G3             Jz X000             Jz L 3
                   Call Help          Call Help           Call Help
                 :G3:               X000:               L 3:

Xthl             Mov BP,SP          Pop SI              Mov BP,SP
                 Xchg BX,[BP]       Xchg SI,BX          Xchg BX,[BP]
                                    Push SI

Pop PSW          Pop AX             Pop AX              Pop AX
                 Sahf               Xchg AH,AL          Xchg AL,AH
                                    Sahf                Sahf


Push PSW         Lahf               Lahf                Lahf
                 Push AX            Xchg AH,AL          Xchg AL,AH
                                    Push AX             Push AX
                                    Xchg AH,AL          Xchg AL,AH

Sphl             Mov BX,SP          Mov SP,BX           Mov SP,BX

(Z80 opcodes)

Djnz Label       Dec CX             Dec CH              -------
                 Jnz Label          Jnz Label

Cpir             Xchg BX,DI         UP                  -------
                 Cld                Mov DI,BX
                 Repnz              Repnz
                 Scasb              Scab
                 Xchg BX,DI         Mov BX,DI

Ldir             Xchg BX,SI         Up                  -------
                 Xchg DX,DI         Mov SI,BX
                 Cld                Mov DI,DX
                 Repnz              Rep
                 Movsb              Movb
                 Xchg BX,SI         Mov DX,DI
                 Xchg DX,DI         Mov BX,SI

Push Ix          Push DI            Push [Ix]           -------

Pop  Iy          Pop  SI            Pop [Iy]            -------
```

translated text, also. We therefore suspect that the program has a minor bug with regard to this particular instruction.

Since this test program was nonsense as far as logical program flow is concerned, XLT86 was at a loss to determine the active registers, and sometimes chose simpler instructions than it would have in a real program. The results for XLT86 could be different in different situations.

The conditional jumps for TRANS86 and XLT86 can vary, depending on the distance of the target label from where the jump is.

The conditional return in Seattle's translator references a label called RET. This refers to any RET within 128 bytes on either side of the statement. This is one reason why Seattle's translator should be used with ASM86; no other assembler will take advantage of this feature.

Note that TRANS86 and the Seattle translator treat the DJNZ instruction differently. TRANS86 uses a 16-bit register, CX, and the Seattle translator uses CH, an 8-bit register. A warning message comes out of the Seattle translator reminding the programmer that DJNZ does not affect the flags in the Z80 but that this sequence of instructions will affect the 8086 flags.

Register Mapping

Figure 6 shows a detailed, side-by-side comparison of the differences in register mapping performed by the three translators. Figure 6a deals with the 8080/8085-to-8086 mapping; figure 6b, with Z80-to-8086 mapping.

As the notes there state, TRANS86 does not preserve 8080 byte order on the stack.

The Seattle translator uses SI on loads from memory and DI for stores to memory.

TRANS86 and XLT86 do a register exchange between BX and the appropriate register to allow indirect addressing through BX, then a register exchange to fix up BX and the appropriate register.

Since the 8086 does not have some of the registers of the Z80, the translators can't support them. The programmer can, however, map those registers to a memory location.

TRANS86 generates memory references to storage locations supplied by the programmer to take care of the Z80's IX, IY, BC', DE', and HL' registers.

Summing Up the Translators

A general view is that Sorcim's TRANS86 is a useful product if the original source is in 8080 or ACT80 form and the user has ACT86 as a target 8086 assembler. The register and flag usage appear to be a little looser than for the other two programs. This requires more knowledge and more involvement from the programmer to make sure that the sense of the translated code is maintained. No limitations exist as to the size of the source file and macros are supported if the input is in ACT80 format. Sorcim's TRANS86 is sold separately from ACT86, but they should be used together.

The Seattle Computer Products' Z80-to-8086 translator is a straightforward code translator that uses Zilog mnemonics and runs only on Z80-based processors. There appear to be no limitations as to the size of the source program that may be translated since the program translates one instruction at a time. Register and flag usage are very conservative, protecting the source architecture as much as possible and providing warnings when potential problems could arise. The converted program has more of a chance of working the first time than a less conservative translation would have.