ASSEMBLY LANGUAGE TRANSLATORS

ITOZ ZTOI ITOT TTOI -- Z-System Tools

USER'S GUIDE

by

Robert Doolittle

TABLE OF CONTENTS

Author's Note

These programs have been thoroughly tested and are believed to be correct. If you find something not to your liking let us know. We welcome your comments, criticisms or questions. Please call or write if you experience any problems.

Robert Doolittle
Echelon Team Member
1290 Monument Street
Telephone 213/454-8270
Pacific Palisades, CA 90272

ITOZ and ZTOI USER'S GUIDE

These programs translate assembly source code from Intel extended 8080 to Zilog Z-80 mnemonics (ITOZ) and vice versa (ZTOI). The output source file is written to disk and is completely formatted and ready for assembly. The extended 8080 set is that defined by Digital Research in their macro library Z80.LIB furnished with their macro assembler MAC. Please note that if your source programs use only the 8080 subset of the Z-80 instructions then you need not be concerned with the Z80.LIB extensions.

The use of ITOZ and ZTOI are nearly identical. Therefore in what follows we shall just refer to ITOZ with the understanding that it also applies to ZTOI except where noted.

COMMAND LINE FORMAT

To execute ITOZ for translation of an 8080 source program to Zilog Z-80 mnemonics the Z-System (and CP/M) command line format is:

```
A>ITOZ ifilename[filetype][/O] [ofilename][filetype]
```

where the parameters in square brackets are optional.

ifilename=input source file name
ofilename=output source file name
/O=options as described below

To maintain compatibility with some assemblers two switch options are available. The slash (/) must immediately follow the input file name or filetype if used. Only one slash is necessary for either or both options.

The "c" option will add a colon after a label if one does not already exist. The "s" option will convert asterisks, used as comment line delimiters, to semicolons in the output file. To invoke both options, for example, the format would be:

/cs or /sc

If you are translating a file that uses a double colon to represent a public label do not use the "c" switch.

An optional output file name is also permitted on the command line. If this optional name is omitted then the default output file name for ITOZ is ZLG.ASM and for ZTOI is X80.ASM.

CP/M is a registered trademark of Digital Research; Z-System is a trademark of Echelon.

INPUT FILE FORMAT RULES

Certain input source file format rules must be followed to obtain a successful translation. Many of these are dictated by common assembler syntax rules. The rules are as follows:

1. Only one instruction per line is permitted
2. All instructions must be in upper case
3. Each line must terminate in a carriage return and line feed
4. No line numbers are permitted in this version
5. Lines must be less than 128 characters long including comments
6. All labels must start in column one
7. The mnemonic or instruction field must be preceded by at least one space or tab if no label is present
8. There must be at least one space or tab between the end of a label and the beginning of the instruction unless the label ends in a colon
9. EQU and SET pseudo-op statements may begin in any column
10. Other pseudo-ops or macro names must begin as defined for instructions in Rule 7
11. Comment lines must be preceded by a semicolon or asterisk and may begin in any column
12. Comments following instructions on the same line must also begin with a semicolon or asterisk and must be preceded by at least one space or tab only if the asterisk is used

Blank lines, as well as blank lines following a label, are supported.

Due to the variety of assembler pseudo-ops some compromises had to be made. There is a logical problem with pseudo-ops starting with an asterisk and comment lines delimited by an asterisk. This is solved in these translators by the following additional rule:

13. If pseudo-ops are used which start with an asterisk then a label must be used on the same line. The label, of course, may be a dummy. In addition, actual comment lines following a label must be delimited by a semicolon.

EXECUTION

When program execution begins ITOZ will print a message on the terminal indicating that the translation is underway. If an unrecognizable instruction is encountered a syntax error message is displayed at the console along with a line number indicator. This line is also flagged in the output file by the comment line "FOLLOWING INSTRUCTION NOT RECOGNIZED." The translator makes no attempt to translate this line but simply copies it to the output file.

Most of the common pseudo-ops are included in the translator's internal tables. If you get a syntax error for a pseudo-op or macro name which appears in your input source file you should then run the utility, ADNAME.COM as described below. This will prevent future occurrences of this error. As noted below, this utility does <u>not</u> translate or, more accurately, it results in a one-to-one translation (exact copy) of entire line.

In all CPU instructions that involve an index register displacement, the translators will interpret the explicit lack of a displacement argument to imply the value 0.

ADNAME.COM

This utility permits the user to add custom macro names or pseudo-ops to the translator's internal tables. It will operate on either ITOZ.COM or ZTOI.COM. It will also operate on either of these files if they have been renamed. Please note that no translation occurs for these names. The use of ADNAME permits the translators to recognize these names and to copy the line from the input file to the output file without generating an error message. If you experience difficulty with assembly of a translated file it may be necessary to edit the translated file to correct for the particular pseudo-ops used by your assembler. The command line format for ADNAME is:

```
A>ADNAME filename.COM
```

where 'filename' is ITOZ, ZTOI, or whatever you may have renamed them. The program will then prompt you for the new names to be entered. Each entry is terminated by a carriage return. A blank line (CR) terminates the program. The old file is renamed with the same filename and with a filetype of BAK. The new file is renamed with the same filename and with a filetype of COM. The new COM file will include your new names. If you wish you may erase the BAK file. We strongly recommend, however, that you always keep a backup copy of the original files.

The space allocated in the translators for this function is 200 bytes. If N is the number of new names and C is the total number of characters in the new names then 2*N+C must be <= 200. Attempts to exceed this number will not be accepted and a warning message will be issued.

ADNAME may be used on the current COM file as often as you please. Any new names will be added to those already included and a new COM file and new BAK file will be created.

EXTENDED 8080 VERSUS ZILOG Z-80 MNEMONIC TABLE

These translators use an extended 8080 mnemonic set which is similar to the Technical Design Laboratories (TDL) mnemonics. All Intel 8080 mnemonics are preserved. The Z-80 peculiar instructions differ from the ZILOG mnemonics as shown in the accompanying table. This mnemonic set is identical to that released by Digital Research as Z80.LIB to be used with their CP/M "MAC" Macro Assembler. The following conventions are used in the table.

r - any register or memory
rr - any register pair or stack pointer
nn - 8 bit immediate data (0 to 255)
d - 8 bit signed displacement (-128 to 127)
nnnn - 16 bit address or immediate data (0 to 65535)
b - bit number (0 to 7, 7 is most significant)
addr - 16 bit address within PC+127 through PC-128

In cases involving a displacement, d, this parameter is always last one in operand field.

EXT 8080	ZILOG	EXT 8080	ZILOG
LDX r,d	LD r,(IX+d)	LDIR	LDIR
LDY r,d	LD r,(IY+d)	LDD	LDD
STX r,d	LD (IX+d),r	LDDR	LDDR
STY r,d	LD (IY+d),r	CCI	CPI
MVIX nn,d	LD (IX+d),nn	CCIR	CPIR
MVIY nn,d	LD (IY+d),nn	CCD	CPD
LDAI	LD A,I	CCDR	CPDR
LDAR	LD A,R	ADDX d	ADD (IX+d)
STAI	LD I,A	ADDY d	ADD (IY+d)
STAR	LD R,A	ADCX d	ADC (IX+d)
LXIX nnnn	LD IX,nnnn	ADCY d	ADC (IY+d)
LXIY nnnn	LD IY,nnnn	SUBX d	SUB (IX+d)
LBCD nnnn	LD BC,(nnnn)	SUBY d	SUB (IY+d)
LDED nnnn	LD DE,(nnnn)	SBBX d	SBC (IX+d)
LSPD nnnn	LD SP,(nnnn)	SBBY d	SBC (IY+d)
LIXD nnnn	LD IX,(nnnn)	ANAX d	AND (IX+d)
LIYD nnnn	LD IY,(nnnn)	ANAY d	AND (IY+d)
SBCD nnnn	LD (nnnn),BC	XRAX d	XOR (IX+d)
SDED nnnn	LD (nnnn),DE	XRAY d	XOR (IY+d)
SSPD nnnn	LD (nnnn),SP	ORAX d	OR (IX+d)
SIXD nnnn	LD (nnnn),IX	ORAY d	OR (IY+d)
SIYD nnnn	LD (nnnn),IY	CMPX d	CP (IX+d)
SPIX	LD SP,IX	CMPY d	CP (IY+d)
SPIY	LD SP,IY	INRX d	INC (IX+d)
PUSHIX	PUSH IX	INRY d	INC (IY+d)
PUSHIY	PUSH IY	DCRX d	DEC (IX+d)

POPIX	POP IX	DCRY d	DEC (IY+d)
POPIY	POP IY	NEG	NEG
EXAF	EX AF,AF'	IM0	IM 0
EXX	EXX	IM1	IM 1
XTIX	EX (SP),IX	IM2	IM 2
XTIY	EX (SP),IY	DADC rr	ADC HL,rr
LDI	LDI	DSBC rr	SBC HL,rr
DADX rr	ADD IX,rr	OUTI	OUTI
DADY rr	ADD IY,rr	OUTIR	OTIR
INXIX	INC IX	IND	IND
INXIY	INC IY	INDR	INDR
DCXIX	DEC IX	OUTD	OUTD
DCXIY	DEC IY	OUTDR	OTDR
BIT b,r	BIT b,r	RLCR r	RLC r
SETB b,r	SET b,r	RLCX d	RLC (IX+d)
RES b,r	RES b,r	RLCY d	RLC (IY+d)
BITX b,d	BIT b,(IX+d)	RALR r	RL r
BITY b,d	BIT b,(IY+d)	RALX d	RL (IX+d)
SETX b,d	SET b,(IX+d)	RALY d	RL (IY+d)
SETY b,d	SET b,(IY+d)	RRCR r	RRC r
RESX b,d	RES b,(IX+d)	RRCX d	RRC (IX+d)
RESY b,d	RES b,(IY+d)	RRCY d	RRC (IY+d)
JR addr	JR addr	RARR r	RR r
JRC addr	JR C,addr	RARX d	RR (IX+d)
JRNC addr	JR NC,addr	RARY d	RR (IY+d)
JRZ addr	JR Z,addr	SLAR r	SLA r
JRNZ addr	JR NZ,addr	SLAX d	SLA (IX+d)
DJNZ addr	DJNZ, addr	SLAY d	SLA (IY+d)
PCIX	JP (IX)	SRAR r	SRA r
PCIY	JP (IY)	SRAX d	SRA (IX+d)
RETI	RETI	SRAY d	SRA (IY+d)
RETN	RETN	SRLR r	SRL r
INP r	IN r,(C)	SRLX d	SRL (IX+d)
OUTP r	OUT (C),r	SRLY d	SRL (IY+d)
INI	INI	RLD	RLD
INIR	INIR	RRD	RRD

* *

ITOT and TTOI USER'S GUIDE

These programs translate assembly source code from Intel extended 8080 to TDL/CDL (Xitan) mnemonics (ITOT) and vice versa (TTOI). The output source file is written to disk and is completely formatted and ready for assembly. The extended 8080 set is that defined by Digital Research in their macro library Z80.LIB furnished with their macro assembler MAC. Please note that if your source programs use only the 8080 subset of the Z-80 instructions then you need not be concerned with the Z80.LIB extensions.

The use of ITOT and TTOI are nearly identical. Therefore in what follows we shall just refer to ITOT with the understanding that it also applies to TTOI except where noted.

COMMAND LINE FORMAT

To execute ITOT for translation of an 8080 source program to TDL/CDL mnemonics the Z-System (and CP/M) command line format is:

```
A>ITOT ifilename[filetype] [ofilename][filetype]
```

where the parameters in square brackets are optional.

```
 ifilename=input source file name
ofilename=output source file name
```

The optional output file name defaults to X80.ASM for TTOI and to TDL.ASM for ITOT

ITOT will automatically insert colons after labels, if not present, since the TDL/CDL assembler requires them. It will also convert any asterisks, used as comment delimiters, to semicolons for the same reason. Furthermore, ITOT will automatically supply an .END statement if required.

INPUT FILE FORMAT RULES

Certain input source file format rules must be followed to obtain a successful translation. Many of these are dictated by common assembler syntax rules. The rules are as follows:

1. Only one instruction per line is permitted

2. All instructions must be in upper case

3. Each line must terminate in a carriage return and line feed

4. No line numbers are permitted in this version

5. Lines must be less than 128 characters long including comments

6. All labels must start in column one

7. The mnemonic or instruction field must be preceded by at least one space or tab if no label is present

8. There must be at least one space or tab between the end of a label and the beginning of the instruction unless the label ends in a colon

9. EQU and SET pseudo-op statements may begin in any column

10. Other pseudo-ops or macro names must begin as defined for instructions in Rule 7

11. Comment lines must be preceded by a semicolon or asterisk and may begin in any column. (See also Rule 13)

12. Comments following instructions on the same line must also begin with a semicolon or asterisk and must be preceded by at least one space or tab only if the asterisk is used

Blank lines, as well as blank lines following a label, are supported.

Due to the variety of assembler pseudo-ops some compromises had to be made. There is a logical problem with pseudo-ops starting with an asterisk and comment lines delimited by an asterisk. This is solved in these translators by the following additional rule:

13. If pseudo-ops are used which start with an asterisk then a label must be used on the same line. The label may, of course, be a dummy. In addition, actual comment lines following a label must be delimited by a semicolon.

EXECUTION

When program execution begins ITOT will print a message on the terminal indicating that the translation is underway. If an unrecognizable instruction is encountered a syntax error message is displayed at the console along with a line number indicator. This line is also flagged in the output file by the comment line "FOLLOWING INSTRUCTION NOT RECOGNIZED." The translator makes no attempt to translate this line but simply copies it to the output file.
Most of the common pseudo-ops are included in the translator's internal tables. If you get a syntax error for a pseudo-op or macro name which appears in your input source file you should then run the utility, ADNAME.COM as described below. This will prevent future occurrences of this error. As noted below, this utility does not translate or, more accurately, it results in a one-to-one translation (exact copy) of the entire line.

These translators do not support the TDL .DEFINE (MACRO) pseudo-op since there is not, in general, a unique translation. These can usually be handled after translation with an editor. The .EXIT (EXITM) pseudo-op is included, however.

All arithmetic and logical operators of the TDL set are supported with the exception of the unary radix change.

Three of the conditional assembly pseudo-ops of the TDL set are supported. These are: .IFN (IF) which is false if its argument evaluates to zero and true otherwise, .IFB (IF NUL) which is true if its argument is blank, and .IFNB (IF NOT NUL) which is true if its argument is not blank. The ELSE construct is also supported. No part of the body of the IF or ELSE is permitted on the same line as the IF statement itself. Left and right square brackets must surround the arguments of .IFB and

ADNAME.COM

This utility permits the user to add custom macro names or pseudo-ops to the translator's internal tables. It will operate on either ITOT.COM or TTOI.COM. It will also operate on either of these files if they have been renamed. Please note that no translation occurs for these names. The use of ADNAME permits the translators to recognize these names and to copy the line from the input file to the output file without generating an error message. If you experience difficulty with assembly of a translated file it may be necessary to edit the translated file to correct for the particular pseudo-ops used by your assembler. The command line format for ADNAME is:

```
A>ADNAME filename
```

where 'filename' is ITOT, TTOI or whatever you may have renamed them. A filetype of COM is automatically assigned by ADNAME. The program will then prompt you for the new names to be entered. Each entry is terminated by a carriage return. A blank line (CR) terminates the program. The old file is renamed with the same filename and with a filetype of BAK. The new file is renamed with the same filename and with a filetype of COM. The new COM file will include your new names. If you wish you may erase the BAK file. We strongly recommend, however, that you always keep a backup copy of the original files. The space allocated in the translators for this function is 200 bytes. If N is the number of new names and C is the total number of characters in the new names then 2*N+C must be <= 200. Attempts to exceed this number will not be accepted and a warning message will be issued.

ADNAME may be used on the current COM file as often as you please. Any new names will be added to those already included and a new COM file and new BAK file will be created.

EXTENDED 8080 VERSUS TDL/CDL MNEMONIC TABLE

The Z-80 peculiar instructions for TDL differ from the extended 8080 set as used here. The extended 8080 mnemonic set is identical to that released by Digital Research as Z80.LIB to be used with their CP/M "MAC" Macro Assembler. The following conventions are used in the table.

r - any register or memory
rr - any register pair or stack pointer
nn - 8 bit immediate data (0 to 255)
d - 8 bit signed displacement (-128 to 127)
nnnn - 16 bit address or immediate data (0 to 65535)
b - bit number (0 to 7, 7 is most significant)
addr - 16 bit address within PC+127 through PC-128

In cases involving a displacement, d, this parameter is always the last one in operand field.

EXT 8080	TDL	EXT 8080	TDL
LDX r,d	MOV r,d(X)	LDIR	LDIR
LDY r,d	MOV r,d(Y)	LDD	LDD
STX r,d	MOV d(X),r	LDDR	LDDR
STY r,d	MOV d(Y),r	CCI	CCI
MVIX nn,d	MVI d(X),nn	CCIR	CCIR
MVIY nn,d	MVI d(Y),nn	CCD	CCD
LDAI	LDAI	CCDR	CCDR
LDAR	LDAR	ADDX d	ADD d(X)
STAI	STAI	ADDY d	ADD d(Y)
STAR	STAR	ADCX d	ADC d(X)
LXIX nnnn	LXI X,nnnn	ADCY d	ADC d(Y)
LXIY nnnn	LXI Y,nnnn	SUBX d	SUB d(X)
LBCD nnnn	LBCD nnnn	SUBY d	SUB d(Y)
LDED nnnn	LDED nnnn	SBBX d	SBB d(X)
LSPD nnnn	LSPD nnnn	SBBY d	SBB d(Y)
LIXD nnnn	LIXD nnnn	ANAX d	ANA d(X)
LIYD nnnn	LIYD nnnn	ANAY d	ANA d(Y)
SBCD nnnn	SBCD nnnn	XRAX d	XRA d(X)
SDED nnnn	SDED nnnn	XRAY d	XRA d(Y)
SSPD nnnn	SSPD nnnn	ORAX d	ORA d(X)
SIXD nnnn	SIXD nnnn	ORAY d	ORA d(Y)
SIYD nnnn	SIYD nnnn	CMPX d	CMP d(X)
SPIX	SPIX	CMPY d	CMP d(Y)
SPIY	SPIY	INRX d	INR d(X)
PUSHIX	PUSH X	INRY d	INR d(Y)
PUSHIY	PUSH Y	DCRX d	DCR d(X)
POPIX	POP X	DCRY d	DCR d(Y)
POPIY	POP Y	NEG	NEG
EXAF	EXAF	IM0	IM0
EXX	EXX	IM1	IM1

XTIX	XTIX
XTIY	XTIY
LDI	LDI
DADX rr	DADX rr
DADY rr	DADY rr
INXIX	INX X
INXIY	INX Y
DCXIX	DCX X
DCXIY	DCX Y
BIT b,r	BIT b,r
SETB b,r	SET b,r
RES b,r	RES b,r
BITX b,d	BIT b,d(X)
BITY b,d	BIT b,d(Y)
SETX b,d	SET b,d(X)
SETY b,d	SET b,d(Y)
RESX b,d	RES b,d(X)
RESY b,d	RES b,d(Y)
JR addr	JMPR addr
JRC addr	JRC addr
JRNC addr	JRNC addr
JRZ addr	JRZ addr
JRNZ addr	JRNZ addr
DJNZ addr	DJNZ addr
PCIX	PCIX
PCIY	PCIY
RETI	RETI
RETN	RETN
INP r	INP r
OUTP r	OUTP r
INI	INI
INIR	INIR
IM2	IM2
DADC rr	DADC rr
DSBC rr	DSBC rr
OUTI	OUTI
OUTIR	OUTIR
IND	IND
INDR	INDR
OUTD	OUTD
OUTDR	OUTDR
RLCR r	RLCR r
RLCX d	RLCR d(X)
RLCY d	RLCR d(Y)
RALR r	RALR r
RALX d	RALR d(X)
RALY d	RALR d(Y)
RRCR r	RRCR r
RRCX d	RRCR d(X)
RRCY d	RRCR d(Y)
RARR r	RARR r
RARX d	RARR d(X)
RARY d	RARR d(Y)
SLAR r	SLAR r
SLAX d	SLAR d(X)
SLAY d	SLAR d(Y)
SRAR r	SRAR r
SRAX d	SRAR d(X)
SRAY d	SRAR d(Y)
SRLR r	SRLR r
SRLX d	SRLR d(X)
SRLY d	SRLR d(Y)
RLD	RLD
RRD	RRD

For TTOI the following additional translations are valid:

JPE	nnnn	JO	nnnn	JPO	nnnn	JNO	nnnn
CPE	nnnn	CO	nnnn	CPO	nnnn	CNO	nnnn
RPE		RO		RPO		RNO	

The following pseudo-ops and operators are also included in both translators:

SET	=	EQU	==
PAGE	.PAGE	TITLE	.TITLE
IF	.IFN	END	.END
IF NUL	.IFB	IF NOT NUL	.IFNB
DS(2)	.BLKW	DW	.WORD
DB	.BYTE	DB	.ASCII
DS	.BLKB	ORG	.LOC
ENTRY	.ENTRY	EXTRN	.EXTERN
ELSE	][	ENDIF	]
RADIX	.RADIX	LIST	.LIST
XLIST	.XLIST	EXIT	.EXIT
$	.	NOT	#
MOD	@	OR	!
XOR	^	AND	&
SHR	<	SHL	>

* *
* *
* *