INTRINSIC SUBROUTINES

INPUT (port,var)	Port Input
OUTPUT (port, exp)	Port Output

ISIS-II 8080/8085 COMPILER INVOCATION

The 8080/8085 FORTRAN compiler is invoked by the ISIS-II command:

–[device] FORT80 *sourcefile* [compiler controls]

ISIS-II COMPILER CONTROLS

The following list shows the controls available, the basic function they control, and whether they are primary or general (P/G). Default controls are italicized.

Controls	P/G	FUNCTION AREA
OBJECT /NOOBJECT	P	Object File
DEBUG/*NODEBUG*	P	Object File
OPTIMIZE (0)/*OPTIMIZE (1)*	P	Object File
PRINT /NOPRINT	P	Compiler Listing
LIST /NOLIST	G	Compiler Listing
SYMBOLS/*NOSYMBOLS*	P	Compiler Listing
CODE/*NOCODE*	G	Compiler Listing
XREF/*NOXREF*	P	Cross-Reference Listing
PAGING /NOPAGING	P	Listing Format
PAGELENGTH *(60)*	P	Listing Format
PAGEWIDTH *(120)*	P	Listing Format
DATE	P	Listing Format
TITLE	P	Listing Format
EJECT	G	Listing Format
REENTRANT	P	Procedure Reentrancy
DO77 /DO66	P	DO Loop Interpretation
STORAGE (INTEGER_*2_)	P	Storage Unit Length
STORAGE (LOGICAL_*1_)	P	Storage Unit Length
FREEFORM/*NOFREEFORM*	G	Source Line Format
INCLUDE	G	Source File Inclusion
WORKFILES *(:F1:,:F1:)*	P	Workfile Devices
SAVE	G	Stack Controls
RESTORE	G	Fetch Controls

LINKING RELOCATABLE OBJECT MODULES

The syntax of the ISIS-II LINK command is:

LINK *inputlist* TO *linkfile* [link controls]

inputlist must include the following:

[RMX8xx.LIB (START),] object-files, F80RUN.LIB, &

```
                           ⎧ FPSOFT.LIB ⎫
⎧ F80ISS.LIB ⎫             ⎪ FPHARD.LIB ⎪
⎨ F80RMX.LIB ⎬ ,FPEF.LIB,  ⎨ FPSFTX.LIB ⎬ ,[RMX-files,]PLM80.LIB
⎩ F80NIO.LIB ⎭             ⎪ FPHRDX.LIB ⎪
                           ⎪ FPHX10.LIB ⎪
                           ⎩ FPNULL.LIB ⎭
```

where braces indicate a choice and the items in brackets are required only under RMX/80. LINK controls are MAP, NAME, and PRINT.

To link non-RMX programs for which F80RUN.LIB, F80ISS.LIB, FPEF.LIB, FPSOFT.LIB, and PLM80.LIB are selected and LINK controls are not needed, use:

SUBMIT FLINK (*objectfile, linkfile* [,*lib-drive*])

where if *lib-drive* is omitted, the libraries are assumed to be on :F0:.

LOCATING MODULES

The syntax of the ISIS-II LOCATE command is:

LOCATE *inputfile* [TO *outputfile*] [locate controls]

LOCATE controls are:

ORDER (segids)	//(addr)	NAME (module)
CODE (addr)	PRINT fname	STACKSIZE (m)
DATA (addr)	NOPRINT	START (addr)
STACK (addr)	MAP	PURGE
MEMORY (addr)	LINES	RESTART0
/common name/ (addr)	PUBLICS	SYMBOLS

LOADING AND EXECUTING A PROGRAM ON INTELLEC® MICROCOMPUTER DEVELOPMENT SYSTEMS

The command syntax to load and execute a FORTRAN program is as follows:

[device] *object-file* [UNIT n = device] [,UNIT n = device]...

HEX-ASCII TABLE

NUL	00	+	2B	V	56
SOH	01	,	2C	W	57
STX	02	–	2D	X	58
ETX	03	.	2E	Y	59
EOT	04	/	2F	Z	5A
ENQ	05	0	30	[	5B
ACK	06	1	31	\	5C
BEL	07	2	32	]	5D
BS	08	3	33	∧ (↑)	5E
HT	09	4	34	– (←)	5F
LF	0A	5	35	’	60
VT	0B	6	36	a	61
FF	0C	7	37	b	62
CR	0D	8	38	c	63
SO	0E	9	39	d	64
SI	0F	:	3A	e	65
DLE	10	;	3B	f	66
DC1 (X-ON)	11	<	3C	g	67
DC2 (TAPE)	12	=	3D	h	68
DC3 (X-OFF)	13	>	3E	i	69
DC4 (TAPE)	14	?	3F	j	6A
NAK	15	@	40	k	6B
SYN	16	A	41	l	6C
ETB	17	B	42	m	6D
CAN	18	C	43	n	6E
EM	19	D	44	o	6F
SUB	1A	E	45	p	70
ESC	1B	F	46	q	71
FS	1C	G	47	r	72
GS	1D	H	48	s	73
RS	1E	I	49	t	74
US	1F	J	4A	u	75
SP	20	K	4B	v	76
!	21	L	4C	w	77
’’	22	M	4D	x	78
#	23	N	4E	y	79
$	24	O	4F	z	7A
%	25	P	50	{	7B
&	26	Q	51	|	7C
’	27	R	52	} (ALT MODE)	7D
(	28	S	53	~	7E
)	29	T	54	DEL (RUB OUT	7F
*	2A	U	55		

REFERENCES

FORTRAN-80 Programming Manual	9800481
ISIS-II FORTRAN-80 Compiler Operator's Manual	9800480
ISIS-II User's Guide	9800306
RMX/80 User's Guide	9800522

intel®

3065 Bowers Avenue, Santa Clara, California 95051
(408) 987-8080

Printed in U.S.A./B-131/1178/10K DD

intel®

FORTRAN-80 REFERENCE CARD

9800547C

CODING SEQUENCE

This diagram shows the order of statements and comment lines within a program unit. Statements on the same level can be interspersed. For example, FORMAT statements can be interspersed with executable statements.

<table>
<tr><td rowspan="5">Comment Lines</td><td colspan="2">PROGRAM, FUNCTION, SUBROUTINE, or BLOCK DATA Statement</td></tr>
<tr><td rowspan="4">FORMAT Statements</td><td>IMPLICIT Statement</td></tr>
<tr><td>Other Specification Statements</td></tr>
<tr><td>DATA Statement</td></tr>
<tr><td>Executable Statements</td></tr>
<tr><td colspan="3">END Statement</td></tr>
</table>

FORTRAN STATEMENTS

NONEXECUTABLE STATEMENTS

Main Program Definition:

PROGRAM name	Name main program.

Procedure Definition:

[typ] FUNCTION func ([d[,d]...])	Define FUNCTION Subprogram.
SUBROUTINE sub [([d[,d]...])]	Define SUBROUTINE Subprogram.

Specification Statements:

IMPLICIT typ (a[,a]...) [,typ (a[,a]...)]...	Specify implicit typing.
INTEGER [*len[,]] name[,name]...	Define integer variables.
REAL v [,v]...	Define real variables.
LOGICAL [*len[,]] name[,name]...	Define logical variables.
CHARACTER [*len[,]] name[,name]...	Define character variables.
DIMENSION a (d[,d...]) [,a(d[,d...])]...	Define array(s).
COMMON [/ [cb] /]nlist [[,] / [cb] / nlist]...	Define common block(s).
EQUIVALENCE (nlist) [,(nlist)]...	Specify shared memory.
EXTERNAL proc [,proc]...	Allow procedure to be actual argument.
INTRINSIC func [,func]...	Allow intrinsic to be actual argument.
SAVE /cb/ [, / cb /]...	Save common data.

Data Initialization:

BLOCK DATA [sub]	Define BLOCK DATA subprogram.
DATA nlist / clist / [[,]nlist / clist /]...	Specify data value(s).

Input/Output:

label FORMAT ([flist])	Specify I/O format.

'flist' Items

Repeatable		Nonrepeatable	
Iw	Integer	'string'	Literal Constant
Fw.d	Real No.	nHstring	Hollerith
Ew.d	Real No.	nX	Record Position
Ew.dEe	Real + Exponent	/	Record Skip
Lw	Logical	kP	Scale Factor
A	Alphanumeric	BN	Blank Interpretation
Aw	Alphanumeric	BZ	Blank Interpretation
Bw	Binary I/O	$	File Position
Zw	Hexadecimal I/O		

EXECUTABLE STATEMENTS

Subroutine Reference:

CALL sub [([a[,a]...])]	Subroutine call.
RETURN	Return from external procedure.

Value Assignment:

ASSIGN s TO ivname	Equate name and label.
v = e	Arithmetic, logical, or character assignment.

Execution Control:

GO TO s	Unconditional GO TO.
GO TO (s[,s]...) [,] exp	Computed GO TO.
GO TO i [[,] (s[,s]...)]	Assigned GO TO.
IF (exp) s1, s2, s3	Arithmetic IF.
IF (exp) stmt	Logical IF.
IF (exp) THEN	Begin IF block.
ELSE IF (exp) THEN	Begin ELSE IF block.
ELSE	Begin ELSE block.
END IF	End IF block.
DO s [,] i = e1, e2[,e3]	Loop control.
CONTINUE	Continue execution.

Program Termination:

PAUSE [n]	Halt program execution.
STOP [n]	Terminate program execution.
END	End program unit.

Input/Output (Data Transfer):

READ (cilist) [inlist]	Read input items.
READ f [,inlist]	Read default file.
WRITE (cilist) [outlist]	Write output items.
PRINT f [,outlist]	Write default file.

'cilist' Items

[UNIT =] u	Unit Specifier	IOSTAT = ios	I/O Status Specifier
[FMT =] f	Format Specifier	ERR = stmt	Error Specifier
REC = recno	Record No. Spec.	END = stmt	EOF Specifier

Input/Output (file Control):

OPEN (olist)	Connect unit/file.
CLOSE (clist)	Disconnect unit/file.
BACKSPACE u	Backspace file.
BACKSPACE (alist)	Backspace file.
REWIND u	Rewind file.
REWIND (alist)	Rewind file.
ENDFILE u	Mark end-of-file.
ENDFILE (alist)	Mark end-of-file.

'olist' Items

[UNIT] = u	Unit Specifier	ACCESS = acc	Access Method
IOSTAT = ios	I/O Status Spec.	FORM = fmt	Formatting Specifier
ERR = stmt	Error Specifier	RECL=reclen	Record Length
FILE = fnam	File Name	BLANK = blnk	Blank Interpretation
STATUS=stat	File Status	CARRIAGE=car	Carriage Control

'clist' Items

[UNIT =] u	Unit Specifier	ERR = stmt	Error Specifier
IOSTAT = ios	I/O Status Spec.	STATUS=stat	File Disposition

'alist' Items

[UNIT =] u	Unit Specifier	IOSTAT = ios	I/O Status Specifier
ERR = stmt	Error Specifier		

INTRINSIC FUNCTIONS

FORM	FUNCTION	TYPE OF	
		ARGUMENTS	FUNCTION
INT (a)	Convert a to type integer	Real	Integer
IFIX (a)	Convert a to type integer	Real	Integer
REAL (a)	Convert a to type real	Integer	Real
FLOAT (a)	Convert a to type real	Integer	Real
ICHAR (a)	Convert a to type integer	Character	Integer
AINT (a)	Truncate a to integer value	Real	Real
ANINT (a)	Round a to nearest whole number	Real	Real
NINT (a)	Round a to nearest integer	Real	Integer
IABS (a)	Return absolute value of a	Integer	Integer
ABS (a)	Return absolute value of a	Real	Real
MOD (a1,a2)	Return remainder from a1/a2	Integer	Integer
AMOD (a1,a2)	Return remainder from a1/a2	Real	Real
ISIGN (a1,a2)	Transfer sign of a2 to a1	Integer	Integer
SIGN (a1,a2)	Transfer sign of a2 to a1	Real	Real
IDIM (a1,a2)	Return a1-a2 if>0; otherwise 0	Integer	Integer
DIM (a1,a2)	Return a1-a2 if>0; otherwise 0	Real	Real
MAX0 (a1,...,an)	Select largest value from list	Integer	Integer
AMAX1 (a1,...,an)	Select largest value from list	Real	Real
AMAX0 (a1,...,an)	Select largest value from list	Integer	Real
MAX1 (a1,...,an)	Select largest value from list	Real	Integer
MIN0 (a1,...,an)	Select smallest value from list	Integer	Integer
AMIN1 (a1,...,an)	Select smallest value from list	Real	Real
AMIN0 (a1,...,an)	Select smallest value from list	Integer	Real
MIN1 (a1,...,an)	Select smallest value from list	Real	Integer
SQRT (a)	Return $\sqrt{a}$ for a>0	Real	Real
EXP (a)	Return e**a	Real	Real
ALOG (a)	Return log (a) for a>0	Real	Real
ALOG10 (a)	Return log 10 (a) for a>0	Real	Real
SIN (a)	Return sine of a	Real	Real
COS (a)	Return cosine of a	Real	Real
TAN (a)	Return tangent of a	Real	Real
ASIN (a)	Return arcsine of a	Real	Real
ACOS (a)	Return arccosine of a	Real	Real
ATAN (a)	Return arctangent of a	Real	Real
ATAN2 (a1,a2)	Return arctangent of a1/a2	Real	Real
SINH (a)	Return hyperbolic sine of a	Real	Real
COSH (a)	Return hyperbolic cosine of a	Real	Real
TANH (a)	Return hyperbolic tangent of a	Real	Real
LGE (a1,a2)	Return TRUE if $a1 \geq a2$, else return FALSE	Character	Logical
LGT (a1,a2)	Return TRUE if $a1 > a2$, else return FALSE	Character	Logical
LLE (a1,a2)	Return TRUE if $a1 \leq a2$, else return FALSE	Character	Logical
LLT (a1,a2)	Return TRUE if $a1 < a2$, else return FALSE	Character	Logical