FORTRAN LANGUAGE STATEMENT SYNTAX

ASSIGNMENT

```
v=e      (Arithmetic and logical assignment)
ASSIGN s TO i
```

CONTROL

```
GO TO s  (Unconditional)
GO TO (s[,s]...)[,]e  (Computed)
GO TO i[,](s[,s]...)]  (Assigned)
IF (e) s1,s2,s3  (3-branch arithmetic; -,0,+)
IF (e) s1,s2  (2-branch arithmetic; ≠0, =0)
IF (e) s≠  (Direct logical; true)
IF (e) e1,e2  (Indirect logical; true, false)
DO s[,]i=e1,e2[,e3]
CALL sub [([a[,a]...])]
RETURN [ie]
PAUSE [id]
STOP [id]
END
CONTINUE
IF (e) THEN
ELSE
ELSE IF (e) THEN
ENDIF
```

INPUT/OUTPUT

```
READ ([UNIT=]u[,[FMT=]f][,ERR=s][,END=s][,REC=r][,IOSTAT=ios])[iolist]
READ f [,iolist]
WRITE ([UNIT=]u[,[FMT=]f][,ERR=s][,REC=r][,IOSTAT=ios])[iolist]
WRITE f [,iolist]
ENCODE (n,f,dent)[iolist]
DECODE (n,f,aent)[iolist]
PRINT f [,iolist]
PUNCH f [,iolist]
BUFFER IN (u,m)(bloc,eloc)
BUFFER OUT (u,m)(bloc,eloc)
ENDFILE u
ENDFILE ([UNIT=]u[,IOSTAT=ios][,ERR=s])
BACKSPACE u
BACKSPACE ([UNIT=]u[,IOSTAT=ios][,ERR=s])
REWIND u
REWIND ([UNIT=]u[,IOSTAT=ios][,ERR=s])
INQUIRE (ilist)
OPEN (ilist)
CLOSE (ilist)
```

PROGRAM UNIT SPECIFICATION

```
PROGRAM pgm [(fp)]
FUNCTION fun ([d[,d]...])
INTEGER FUNCTION fun ([d[,d]...])
REAL FUNCTION fun ([d[,d]...])
DOUBLE [PRECISION] FUNCTION fun ([d[,d]...])
COMPLEX FUNCTION fun ([d[,d]...])
LOGICAL FUNCTION fun ([d[,d]...])
CHARACTER [*k] FUNCTION fun ([d[,d]...])
SUBROUTINE sub [([d[,d]...])]
ENTRY en [([d[,d]...])]
EXTERNAL proc [,proc]...
BLOCK DATA [sub]
```

DATA SPECIFICATION

```
PARAMETER (p=e[,p=e]...)
DIMENSION a(d) [,a(d)] ...
IMPLICIT type [*n](a[-b][,a[-b]]...)
    [,type[*n](a[-b][,a[-b]]...)
IMPLICIT NONE
INTEGER [*n]v[,v]...
REAL [*n]v[,v]...
DOUBLE [PRECISION] [*n]v[,v]...
COMPLEX [*n]v[,v]...
LOGICAL [*n]v[,v]...
CHARACTER [*n[,]]v[*n][,v[*n]]...
EQUIVALENCE (nlist)[,(nlist)]...
COMMON [/[cb]/nlist[[,]/[cb]/nlist]...
DATA nlist /clist/[[,]nlist/clist/]...
FORMAT flist
fun ([d[,d]...])=e  (Statement function definition)
NAMELIST /group/u[,u]...[[,]/group/u[,u]...
POINTER (p,a [,(p,a)]...
INTRINSIC fun[,fun]...
SAVE [a[,a]...]
```

FORTRAN LANGUAGE PARAMETER DEFINITIONS

a	CALL statement: An actual argument; IMPLICIT statement: Beginning letter of all names in a type specification ((*a-b* specifies a range); POINTER statement: Pointee; SAVE statement: variable name, array name, or common block name preceded and followed by a /
a(d)	An array declarator
b	Last letter of a type range
bloc	Symbolic name of the beginning location
cb	Symbolic name of a common block
clist	A list of constants or symbolic names of constants
d	A dummy argument
dent	The symbolic name of a destination variable, array element, or array
e	An expression
eloc	Symbolic name of the ending location
en	Function or subroutine name used as an entry
f	A format identifier or namelist group name
flist	A format specification
fun	Symbolic name of a function subprogram
group	Group name
h	A character string recognized as a comment
i	An integer variable name
id	A string of 8 digits
ie	Integer expression
ilist	A list of specifiers
iolist	An input/output list
ios	Status specifier
m	Mode specifier
n	Number of characters to be processed
nlist	DATA statement: A list of names and/or implied-DO lists; COMMON and EQUIVALENCE statements: A list of names
**n*	Data length (Integer *2 is 24-bit integer)
p	PARAMETER statement: A symbolic name; POINTER statement: A pointer
pgm	Symbolic name of the main program
proc	The name of an external procedure
s	A statement label
sent	Symbolic name of a source variable, array element, or array
st	Any executable statement that is not a DO, END, or logical IF
sub	Symbolic name of a subroutine or subroutine entry
type	Desired data type
u	Unit identifier or dataset identifier
(u)	Unit identifier specified as a Hollerith constant, left-justified and zero-filled
v	Name of a constant, variable, function, dummy procedure name, or array name
w	Local variable or array name

COMPILER DIRECTIVES

Compiler directive lines begin with characters CDIR$ in columns 1 through 5 and any of the directives listed below in columns 7 through 72.

DIRECTIVE	FUNCTION
EJECT	Ejects to top of next page
LIST	Resumes listable output
NOLIST	Suppresses production of listable output
CODE	Produces listing of assembler code in selected blocks
NOCODE	Suppresses listing of CFT-generated assembler code
VECTOR	Enables vectorization of inner DO-loops
NOVECTOR	Suppresses vectorization of inner DO-loops
NORECURRENCE[=*n*]	Causes DO-loops containing recurrences to execute in scalar or vector mode
NEXTSCALAR	Inhibits vectorization of the next DO-loop
SHORTLOOP	Eliminates run-time check for DO-loop termination (for loops with 1 ≤ iteration count ≤ 64)
IVDEP	Ignores vector dependencies in the next DO-loop
INT24 *list*	Identifies listed variables and arrays as 24-bit integers
INT64 *list*	Identifies listed variables and arrays as 64-bit integers
FASTMD	Uses the fast 46-bit integer multiply and divide in the current block
SLOWMD	Uses the slow 64-bit integer multiply and divide in the current block
BLOCK	Causes a new optimization block to begin with the next statement
NO SIDE EFFECTS *list*	Declares that external subprograms have no side effect on variables used by the calling program
NOIFCON	Disables optimization of conditional replacement statements
RESUMEIFCON	Enables optimization of conditional replacement statements
FLOW	Enables flowtrace
NOFLOW	Disables flowtrace
DYNAMIC *list*	Declares dynamic common blocks
BOUNDS[([*list*])]	Checks arrays for out-of-bounds subscripts
VFUNCTION *list*	Declares that a vector version of an external function exists
ALIGN	Aligns loops and entry points on instruction buffer boundaries
DEBUG	Enables debugging mode
NODEBUG	Disables debugging mode
RESUMEDOREP	Re-enables replacement of successive 1-line DO-loops
NODOREP	Disables replacement of successive 1-line DO-loops
BL	Enables bottom loading of DO-loops
NOBL	Disables bottom loading of DO-loops
SAFEIF	Disables movement of code over a branch
UNSAFEIF	Enables movement of code over a branch

VALUE RANGES REPRESENTED IN THE CRAY-1

BINARY RANGE	APPROXIMATE DECIMAL RANGE
$-2^{63} \le I < 2^{63}$	64-BIT INTEGER $0 \le \|I\| < 10^{19}$
$-2^{23} \le I < 2^{23}$	24-BIT INTEGER $-8388608 \le I < 8388608$
$2^{-8192}\cdot(1/2) \le \|R\| < 2^{8192}\cdot(1/2)$	REAL $10^{-2466} < \|R\| < 10^{2466}$
$2^{-8192}\cdot(1/2) \le \|D\| < 2^{8192}\cdot(1/2)$	DOUBLE PRECISION $10^{-2466} < \|D\| < 10^{2466}$
$2^{-8129}\cdot(1/2) \le \|C_R\|, \|C_I\| < 2^{8129}\cdot(1/2)$	COMPLEX $10^{-2466} < \|C_R\|, \|C_I\| < 10^{2466}$

LOGICAL RANGE	DECIMAL RANGE
.TRUE. or .T.	$n<0$
.FALSE. or .F.	$n\ge 0$

COMPILER OPTIONS

OPTION	DESCRIPTION	DEFAULT
A	Aborts job after compilation if any of the program units contains a fatal error	OFF
B	Lists beginning sequence number of each code generation block (G implies B)	OFF
C	Lists common block names and lengths listed on *ldn* after each program unit	ON
D	Lists DO-loop table	OFF
E	Enables recognition of compiler directive lines	ON
F	Enables FLOWTRACE option. (Also see FLOW/NOFLOW directives)	OFF
G	Lists generated CAL code for each program unit. (See CODE/NOCODE directives)	OFF
H	Causes listing of only the first statement of each program unit. All other list options are ignored or disabled.	OFF
I	Enters compiler-generated statement labels in symbol table	OFF
J	Causes all DO-loops to be executed at least once	OFF
L	Enables recognition of output listing control directives	ON
N	Enters null symbols in symbol table (defined but not referenced)	OFF
O	Identifies execution-time array references with out-of-bounds subscripts	OFF
P	Double precision; 64-bit values	ON
Q	Aborts compilation when 100 fatal error messages counted	ON
R	Rounds results on multiply operation	ON
S	Lists FORTRAN source code	ON
T	Lists symbol table after each program unit	ON
U	Enables recognition of INTEGER*2 declaration	ON
V	Vectorizes inner DO-loops	ON
W	Compiles all floating-point operations as return jumps to user-supplied external routines	OFF
X	Lists symbol table with cross references after each program unit (X overrides T)	OFF
Z	Writes DEBUG Symbol Table on $BLD	OFF

CRAY RESEARCH, INC.

CFT REFERENCE CARD

CFT CONTROL STATEMENT FORMAT

```
CFT,I=idn,L=ldn,B=bdn,C=adn,E=n,ON=string,OFF=string,
```

```
TRUNC=nn,AIDS=aids,OPT=option,MAXBLOCK=mb,INT=iL,
```

```
ALLOC=allocation,CPU=cputype,DEBUG,SAVEALL.
```

Parameters can be in any order or can be omitted to cause the default options to apply.

idn Source input dataset name; default is $IN.

ldn Listable output dataset name; default is $OUT. L=0 suppresses all but fatal error messages, which are written on $OUT.

bdn Binary output dataset name; default is $BLD. No EOF is written. If B=0, no binary load files are written.

adn CAL dataset name; default is no dataset.

n Highest message level to be suppressed; default is 3. Fatal errors not suppressed.

Level	Severity
1	COMMENT
2	NOTE
3	CAUTION
4	WARNING
5	ERROR

string String of up to 15 letters that selects a compiler option when specified ON or OFF, respectively. Defaults described in the Compiler Option Chart.

nn Number of bits truncated for floating-point results. Truncated bits zeroed. Range is $0 \le nn < 47$. Default is 0.

aids Number of vectorization inhibition messages.

aids	Number of messages
LOOPNONE	None
LOOPPART	3 per compiler block; 100 messages per compilation (default)
LOOPALL	All

option Optimization level

option	Assumption
NOZEROINC	Constant increment integers cannot be incremented by zero variables (default)
ZEROINC	Constant increment integers can be incremented by zero variables
NOIFCON	Disables optimization of conditional replacement statements (default)
PARTIALIFCON	Enables optimization of conditional replacement statements not involving division and external functions
FULLIFCON	Enables optimization of conditional replacement statements involving division and external functions
FASTMD	Fast 46-bit integer multiply and divide
SLOWMD	Full 64-bit integer multiply and divide (default)

option	Assumption
SAFEDOREP	Enables replacement of 1-line DO-loops (without potential dependencies or equivalenced variables) with a call to a $SCILIB routine performing the same operation more efficiently (default)
FULLDOREP	Enables replacement of 1-line DO-loops (potential dependencies and equivalencies are ignored) with a call to a $SCILIB routine performing the same operation more efficiently
NODOREP	Disables replacement of 1-line DO-loops with a call to a $SCILIB routine
INVMOV	Enables movement of invariant code from DO-loops (default)
NOINVMOV	Disables movement of invariant code from DO-loops
UNSAFEIF	Enables instructions to move over a branch instruction
SAFEIF	Disables instructions moving over a branch instruction (default)
BL	Enables scalar loops to be bottom loaded (default)
NOBL	Disables scalar loops to be bottom loaded
BTREG	Causes CFT to allocate specific scalar variables in a program unit to T registers
NOBTREG	Causes CFT to allocate all user variables to memory. NOBTREG does not affect the allocation of compiler-generated variables to B or T registers or the use of B or T registers temporarily holding values during expression evaluation. (default, unless ALLOC=STACK)

mb Length of code block being optimized or vectorized. Default is 2560 words of internal intermediate text. Values greater than 2560 may increase optimization and internal compiler errors. MAXBLOCK=1 eliminates optimization or vectorization.

fl Integer length. INT=24 uses short 24-bit integers, INT=64 uses full 64-bit integers (default).

allocation
Memory allocation

allocation	Description
STATIC	All memory is statically allocated; a stack is not used (default).
STACK	Read-only constants and entities in a DATA or SAVE statement, or a common block are statically allocated. All other entities are allocated on the stack.
HEAP	Deferred implementation

cputype
Specifies mainframe type running the generated code; default is the machine running CFT.

cputype	Description
CRAY-1M	Generates code for CRAY-1 M Computer Systems
CRAY-1S	Generates code for CRAY-1 S Computer Systems
CRAY-XMP	Generates code for CRAY X-MP Computer Systems

DEBUG Writes sequence number labels at each executable FORTRAN statement to the Debug Symbol Table, allowing breakpoints to be set with SID at statement sequence numbers. DEBUG forces ON=IZ and sets MAXBLOCK=1.

SAVEALL
Compilation occurs as if a SAVE statement with an empty list were in each program unit. SAVEALL overrides OPT=BTREG.

FORTRAN CALLABLE SUBPROGRAMS

b=Boolean; i=integer; r=real; d=double precision; c=complex; l=logical; q=i, r or l; s=i, c or r; t=b, i, c or r; u=b, i, d, c or r; v=i or b; w=i or r; x=b, i, r, or l; y=b, i, d, c, or l; z=b, i, r, d, c, or l; h=Hollerith; ch=character

SUBPROGRAM CALLING SEQ.	CODE GENERATED	VECTORIZATION
TRIGONOMETRIC		
r=SIN(r)	External	Full
d=DSIN(d)	External	Full
c=CSIN(c)	External	Full
r=COS(r)	External	Full
d=DCOS(d)	External	Full
c=CCOS(c)	External	Full
r=TAN(r)	External	Pseudo
d=DTAN(d)	External	Pseudo
r=COT(r)	External	Pseudo
d=DCOT(d)	External	Pseudo
r=ASIN(r)	External	Pseudo
d=DASIN(d)	External	Pseudo
r=ACOS(r)	External	Pseudo
d=DACOS(d)	External	Pseudo
r=ATAN(r)	External	Pseudo
d=DATAN(d)	External	Pseudo
r=ATAN2(r,r)	External	Pseudo
d=DATAN2(d,d)	External	Pseudo
r=SINH(r)	External	Pseudo
d=DSINH(d)	External	Pseudo
r=COSH(r)	External	Pseudo
d=DCOSH(d)	External	Pseudo
r=TANH(r)	External	Pseudo
d=DTANH(d)	External	Pseudo
LOGARITHMIC		
r=ALOG(r)	External	Full
d=DLOG(d)	External	Pseudo
c=CLOG(c)	External	Pseudo
r=ALOG10(r)	External	Full
d=DLOG10(d)	External	Pseudo
TYPE CONVERSION		
r=FLOAT(v)	Inline	Full
i=IFIX(v)	Inline	Full
d=DBLE(u)	Inline	Full
r=SNGL(d)	Inline	Full
c=CMPLX(u[,u])	Inline	Full
i=INT24(v)	Inline	Full
i=LINT(i)	Inline	Full
ch=CHAR(v)	External	None
i=ICHAR(ch)	External	None
r=REAL(s)	Inline	Full
i=INT(u)	Inline	Full
i=IDINT(d)	Inline	Full
EXPONENTIAL		
r=SQRT(r)	External	Full
d=DSQRT(d)	External	Pseudo
c=CSQRT(c)	External	Pseudo
r=EXP(r)	External	Full
d=DEXP(d)	External	Pseudo
c=CEXP(c)	External	Pseudo

SUBPROGRAM CALLING SEQ.	CODE GENERATED	VECTORIZATION
CHARACTER		
l=LGE(ch,ch)	External	None
l=LGT(ch,ch)	External	None
l=LLE(ch,ch)	External	None
l=LLT(ch,ch)	External	None
i=LEN(ch)	Inline	None
i=INDEX(ch,ch)	External	None
ARITHMETIC		
i=IABS(i)	Inline	Full
r=ABS(r)	Inline	Full
d=DABS(d)	Inline	Full
c=CABS(c)	External	Full
i=IDIM(i,i)	Inline	Full
r=DIM(r,r)	Inline	Full
d=DDIM(d,d)	External	Pseudo
r=AINT(r)	Inline	Full
d=DINT(d)	External	Pseudo
r=ANINT(r)	Inline	Full
d=DNINT(d)	External	Pseudo
i=NINT(r)	Inline	Full
i=IDNINT(d)	External	Pseudo
i=MOD(i,i)	External	None
r=AMOD(r,r)	Inline	Full
d=DMOD(d,d)	External	Pseudo
i=ISIGN(i,i)	Inline	Full
r=SIGN(r,r)	Inline	Full
d=DSIGN(d,d)	Inline	Full
r=RANF( )	External	Full
i=RANGET( )	External	None
RANSET(v)	External	None
c=CONJG(c)	Inline	Full
r=REAL(c)	Inline	Full
r=AIMAG(c)	Inline	Full
d=DPROD(r,r)	Inline	Full
MAXIMUM/MINIMUM		
i=MAX0(i)	Inline	Full
r=AMAX0(i)	Inline	Full
r=MAX1(r)	Inline	Full
r=AMAX1(r)	Inline	Full
d=DMAX1(d)	Inline	Full
i=MIN0(i)	Inline	Full
r=AMIN0(i)	Inline	Full
i=MIN1(r)	Inline	Full
r=AMIN1(r)	Inline	Full
d=DMIN1(d)	Inline	Full
INPUT/OUTPUT		
EODW(i)	External	None
i=IEOF(i)	External	None
r=EOF(i)	External	None
r=UNIT(i)	External	None
i=LENGTH(i)	External	None
b=GETPOS(i)	External	None
SETPOS(i,i)	External	None

SUBPROGRAM CALLING SEQ.	CODE GENERATED	VECTORIZATION
DEBUG AIDS		
ENDRPV	External	None
SETRPV(h,q,v)	External	None
SYMDEBUG(h)	External	None
DUMPJOB(i)	External	None
BOOLEAN		
b=AND(x,x)	Inline	Full
b=OR(x,x)	Inline	Full
b=XOR(x,x)	Inline	Full
b=EQV(x,x)	Inline	Full
b=NEQV(x,x)	Inline	Full
b=COMPL(x)	Inline	Full
b=MASK(i)	Inline	Full
b=SHIFT(x,i)	Inline	Full
b=SHIFTL(x,i)	Inline	Full
b=SHIFTR(x,i)	Inline	Full
i=LEADZ(x)	Inline	Full
i=POPCNT(x)	Inline	Full
b=CSNG(x,x,x)	Inline	Full
TIME/DATE		
r=RTC( )	Inline	None
i=IRTC( )	Inline	None
r=TIMEF( )	External	None
r=SECOND(r)	External	None
SECOND(r)	External	None
b=CLOCK(w)	External	None
CLOCK(w)	External	None
b=DATE(w)	External	None
DATE(w)	External	None
b=JDATE(w)	External	None
JDATE(w)	External	None
SYSTEM		
EXIT	External	None
i=LOC(z)	Inline	None
SSWITCH(i,i)	External	None
ABORT(z)	External	None
ERREXIT	External	None
TRBK(i)	External	None
REMARK(z)	External	None
REMARK2(z)	External	None
REMARKF(i,x)	External	None
SENSEFI(i)	External	None
CLEARFI	External	None
SETFI	External	None
CLEARFIS	External	None
SETFIS	External	None
SENSEBT	External	None
CLEARBT	External	None
SETBT	External	None
CLEARBTS	External	None
SETBTS	External	None
i=NUMARG( )	Inline	None
i=SYSTEM(i)	External	None
VECTORIZATION		
b=CVMGP(x,x,x)	Inline	Full
b=CVMGM(x,x,x)	Inline	Full
b=CVMGZ(x,x,x)	Inline	Full
b=CVMGN(x,x,x)	Inline	Full
b=CVMGT(x,x,l)	Inline	Full

GENERIC AND SPECIFIC INTRINSIC FUNCTION NAMES

Generic name	Specific names
INT	INT, IFIX, IDINT
REAL	REAL, FLOAT, SNGL
DBLE	f
CMPLX	f
AINT	AINT, DINT
ANINT	ANINT, DNINT
NINT	NINT, IDNINT
ABS	IABS, ABS, DABS, CABS
MOD	MOD, AMOD, DMOD
SIGN	ISIGN, SIGN DSIGN
DIM	IDIM, DIM DDIM
MAX	MAX0, AMAX1, DMAX1
MIN	MIN0, AMIN1, DMIN1
SQRT	SQRT, DSQRT, CSQRT
EXP	EXP, DEXP, CEXP
LOG	ALOG, DLOG, CLOG
LOG10	ALOG10, DLOG10
SIN	SIN, DSIN, CSIN
COS	COS, DCOS, CCOS
COT ff	COT, DCOT
TAN	TAN, DTAN
ASIN	ASIN, DASIN
ACOS	ACOS, DACOS
ATAN	ATAN, DATAN
ATAN2	ATAN2, DATAN2
SINH	SINH, DSINH
COSH	COSH, DCOSH
TANH	TANH, DTANH

f No specific name
ff A function not specified by the ANSI standard

DO-LOOP FEATURES INHIBITING VECTORIZATION

- Parentheses in subscripts
- A dependent or ambiguous subscript
- A call to a subroutine
- A call to a non-vector function
- GO TO statement in a loop
- Any I/O statement
- Use of character entities

SYMBOLIC DEBUG PACKAGE

The symbolic debug package provides a symbolic memory dump to aid in determining the cause of a job abort. It can be invoked by the DEBUG control statement or the library routine SYMDEBUG. The ON=Z option in the CFT control statement must be specified.

REPRIEVE PROCESSING

Reprieve processing suspends normal system error processing and allows the user to attempt to recover from what normally would be an abort condition. The CRAY-OS Version 1 Reference Manual, publication SR-0011, lists reprievable abort conditions and selection codes.

MANUALS FOR FORTRAN USERS

- FORTRAN (CFT) Reference Manual - SR-0009

  Describes the Cray FORTRAN Language, the Cray FORTRAN Compiler, and related Cray Operating System characteristics.

- Library Reference Manual - SR-0014

  Summarizes the subprograms available to users of the CRAY-1 and CRAY X-MP Computer Systems.

CHARACTER SET

CHAR	ASCII	HEX	ASCII CARD CODE	CHAR	ASCII	HEX	ASCII CARD CODE
NUL	000	00	12-0-9-8-1	@	100	40	8-4
SOH	001	01	12-9-1	A	101	41	12-1
STX	002	02	12-9-2	B	102	42	12-2
ETX	003	03	12-9-3	C	103	43	12-3
EOT	004	04	9-7	D	104	44	12-4
ENQ	005	05	0-9-8-5	E	105	45	12-5
ACK	006	06	0-9-8-6	F	106	46	12-6
BEL	007	07	0-9-8-7	G	107	47	12-7
BS	010	08	11-9-6	H	110	48	12-8
HT	011	09	12-9-5	I	111	49	12-9
LF	012	0A	0-9-5	J	112	4A	11-1
VT	013	0B	12-9-8-3	K	113	4B	11-2
FF	014	0C	12-9-8-4	L	114	4C	11-3
CR	015	0D	12-9-8-5	M	115	4D	11-4
SO	016	0E	12-9-8-6	N	116	4E	11-5
SI	017	0F	12-9-8-7	O	117	4F	11-6
DLE	020	10	12-11-9-8-1	P	120	50	11-7
DC1	021	11	11-9-1	Q	121	51	11-8
DC2	022	12	11-9-2	R	122	52	11-9
DC3	023	13	11-9-3	S	123	53	0-2
DC4	024	14	9-8-4	T	124	54	0-3
NAK	025	15	9-8-5	U	125	55	0-4
SYN	026	16	9-2	V	126	56	0-5
ETB	027	17	0-9-6	W	127	57	0-6
CAN	030	18	11-9-8	X	130	58	0-7
EM	031	19	11-9-8-1	Y	131	59	0-8
SUB	032	1A	9-8-7	Z	132	5A	0-9
ESC	033	1B	0-9-7	[	133	5B	12-8-2
FS	034	1C	11-9-8-4	\	134	5C	0-8-2
GS	035	1D	11-9-8-5	]	135	5D	11-8-2
RS	036	1E	11-9-8-6	^	136	5E	11-8-7
US	037	1F	11-9-8-7	_	137	5F	0-8-5
Space	040	20	None	`	140	60	8-1
!	041	21	12-8-7	a	141	61	12-0-1
"	042	22	8-7	b	142	62	12-0-2
#	043	23	8-3	c	143	63	12-0-3
$	044	24	11-8-3	d	144	64	12-0-4
%	045	25	0-8-4	e	145	65	12-0-5
&	046	26	12	f	146	66	12-0-6
'	047	27	8-5	g	147	67	12-0-7
(	050	28	12-8-5	h	150	68	12-0-8
)	051	29	11-8-5	i	151	69	12-0-9
*	052	2A	11-8-4	j	152	6A	12-11-1
+	053	2B	12-8-6	k	153	6B	12-11-2
,	054	2C	0-8-3	l	154	6C	12-11-3
-	055	2D	11	m	155	6D	12-11-4
.	056	2E	12-8-3	n	156	6E	12-11-5
/	057	2F	0-1	o	157	6F	12-11-6
0	060	30	0	p	160	70	12-11-7
1	061	31	1	q	161	71	12-11-8
2	062	32	2	r	162	72	12-11-9
3	063	33	3	s	163	73	11-0-2
4	064	34	4	t	164	74	11-0-3
5	065	35	5	u	165	75	11-0-4
6	066	36	6	v	166	76	11-0-5
7	067	37	7	w	167	77	11-0-6
8	070	38	8	x	170	78	11-0-7
9	071	39	9	y	171	79	11-0-8
:	072	3A	8-2	z	172	7A	11-0-9
;	073	3B	11-8-6	{	173	7B	12-0
<	074	3C	12-8-4	|	174	7C	12-11
=	075	3D	8-6	}	175	7D	11-0
>	076	3E	0-8-6	~	176	7E	11-0-1
?	077	3F	0-8-7	DEL	177	7F	12-9-7

CRAY RESEARCH, INC. 1440 Northland Dr. Mendota Heights, MN 55120