XSPICE™

Code Model Subsystem Interface Design Document

F.L. Cox, III., W.B. Kuhn,
H.W. Li, J.P. Murray, S.D. Tynor

Georgia Tech Research Institute
Georgia Institute of Technology

Georgia Tech Research Corporation
Office of Technology Licensing
Atlanta, Georgia 30332-0415
Telephone (404) 894 - 6287
Facsimile (404) 894 - 9728

404 894-7046

Contents

List of Figures

List of Tables

1 Scope

1.1 Identification

This Interface Design Document describes the external interfaces of the XSPICE Code Model Subsystem Computer Software Configuration Item (CSCI) of version 2 of the Automatic Test Equipment Software Support Environment (ATESSE). This design is governed by the Software Requirements Specification for the Simulator of the Automatic Test Equipment Software Support Environment (ATESSE).

1.2 System Overview

The ATESSE is an integrated set of software tools designed to support all stages of the life cycle of software used to control Automatic Test Equipment (ATE) in testing analog and hybrid (analog/digital) circuit cards.

The ATESSE includes a mixed-mode (analog/digital) simulator called XSPICE which performs mathematical simulation of a circuit specified by the user. The XSPICE simulator takes input in the form of commands and circuit descriptions and produces output data which predicts the circuit's behavior. The simulator is based on the industry standard SPICE program developed at the University of California at Berkeley and is enhanced and modified to provide mixed-mode, board-level, and system-level simulation capabilities.

The XSPICE Code Model Subsystem described in this document works in conjunction with the XSPICE simulator to provide "code models" and "user-defined node" data types used in simulating circuits and systems.

Several predefined models and node types are delivered with the system. These components of the Code Model Subsystem are referred to as the Code Model Library and the User-Defined Node Library respectively. The following predefined code models are contained in the Code Model Library.

```
Analog Models:

     Gain
     Summer
     Multiplier
     Divider
     Limiter
     Controlled Limiter
     Piecewise Linear Controlled Source
     Analog Switch
     Zener Diode
     Current Limiter
     Hysteresis Block
     Differentiator
     Integrator
     S-Domain Transfer Function
     Slew Rate Block
     Inductive Coupling
     Magnetic Core
     Controlled Sine Wave Oscillator
     Controlled Triangle Wave Oscillator
     Controlled Square Wave Oscillator
     Controlled Oneshot
     Capacitor
     Capacitance Meter
     Inductor
     Inductance Meter

Hybrid Models:

     Digital-to-Analog Node Bridge
     Analog-to-Digital Node Bridge
     Controlled Digital Oscillator

 Digital Models:

     Buffer
     Inverter
     And
     Nand
     Or
     Nor
     Xor
     Xnor
     Tristate
     Open-collector Buffer
```

```
Open-Emitter Buffer
Pullup
Pulldown
D Flip Flop
JK Flip Flop
Toggle Flip Flop
Set-Reset Flip Flop
D Latch
Set-Reset Latch
State Machine
Frequency Divider
RAM
Digital Source
```

The following predefined node types are contained in the User-Defined Node Library.

```
Real
Int
```

The set of available code models and node types in the XSPICE simulator can also be modified and extended by a user through the use of the Code Model Subsystem's "Code Model Toolkit". The Code Model Toolkit consists of the Model Directory Generator, the User-Defined Node Directory Generator, the Code Model Preprocessor, and the Simulator Directory Generator utilities. These utilities work with the host computer operating system software (UNIX) to assist the user in the process of creating new models, node types, and customized simulator executables.

1.3 Document Overview

This document defines the external interfaces of the Code Modeling Toolkit, Code Model Library, and User-Defined Node Library. Section 2 provides a list of applicable documents. Section 3 includes the detailed design of the interfaces to each of the components manipulated by the Code Model Toolkit: Interface Specification Files, Compiled Interface Specification Files, Model Definition Files, User-Defined Node Definition Files, Model Path Files, User-Defined Node Path Files, Model Directory Generator Files, User-Defined Node Directory Generator Files, and Simulator Directory Generator Files. Also included in Section 3 is the external interface definition of each of the supplied code models and node types in the Code Model Library and User-Defined Node Library. Section 4 includes a glossary of technical terminology used throughout this document, a list of acronyms, and a summary of all Project Unique Identifiers (PUIs) related to the Code Model Subsystem.

1.4 Acknowledgement

The XSPICE simulator is based on the SPICE3 program developed by the Electronics Research Laboratory, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.

2 Referenced Documents

1. Software Requirements Specification for the Simulator of the Automatic Test Equipment Software Support Environment (ATESSE), F. L. Cox, W. B. Kuhn, J. P. Murray, S. D. Tynor, Georgia Tech Research Institute, Atlanta, GA, November, 1991.

2. Software User's Manual for the XSPICE Simulator of the Automatic Test Equipment Software Support Environment (ATESSE), F. L. Cox, W. B. Kuhn, H. W. Li, J. P. Murray, S. D. Tynor, M. J. Willis Georgia Tech Research Institute, Atlanta, GA, September, 1992.

3. SPICE3C.1 Nutmeg Programmer's Manual, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, April, 1987.

4. SPICE3 Version 3C1 User's Guide, Thomas L. Quarles, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, April, 1989.

5. SPICE 3C1 Nutmeg Programmer's Guide, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, April, 1989.

6. The Front End to Simulator Interface, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA, April, 1989.

7. The SPICE3 Implementation Guide, Thomas L. Quarles, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, April, 1989.

8. Adding Devices to SPICE3, Thomas L. Quarles, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, April, 1989.

9. The C Programming Language, Second Edition, Brian Kernighan and Dennis Ritchie, Prentice-Hall, Englewood Cliffs, NJ, 1988.

10. Software Design Document for the XSPICE Code Model Subsystem of the Automatic Test Equipment Software Support Environment (ATESSE), F. L. Cox, W. B. Kuhn, H. W. Li, J. P. Murray, S. D. Tynor, Georgia Tech Research Institute, Atlanta, GA, September, 1992.

11. Interface Design Document for the XSPICE Simulator of the Automatic Test Equipment Software Support Environment (ATESSE), F. L. Cox, W. B. Kuhn, H. W. Li, J. P. Murray, S. D. Tynor, Georgia Tech Research Institute, Atlanta, GA, September, 1992.

12. Software Design Document for the XSPICE Simulator of the Automatic Test Equipment Software Support Environment (ATESSE), F. L. Cox, W. B. Kuhn, H. W. Li, J. P. Murray, S. D. Tynor, Georgia Tech Research Institute, Atlanta, GA, September, 1992.

13. Program Design Specification (Volumes 1 and 2) for the Automatic Test Equipment Software Support Environment (ATESSE), F. L. Cox, R. M. Ingle, J. E. Doss, G. T. Fulton, A. M. Gilchrist, R. W. Kearney, W. B. Kuhn, D. A. Moreland, P. P. Warren, B. D. Williams, Georgia Tech Research Institute, Atlanta, GA, October 1988.

14. Data Base Design Document for the Automatic Test Equipment Software Support Environment (ATESSE), F. L. Cox, R. M. Ingle, J. E. Doss, G. T. Fulton, A. M. Gilchrist, R. W. Kearney, W. B. Kuhn, D. A. Moreland, P. P. Warren, B. D. Williams, Georgia Tech Research Institute, Atlanta, GA, October 1988.

15. Analysis of Performance and Convergence Issues for Circuit Simulation, Thomas L. Quarles, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, April, 1989.

16. SPICE2: A Computer Program to Simulate Semiconductor Circuits, Lawrence W. Nagel, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA, May, 1975.

3 Interface Design

3.1 Interface Diagram

Figure 3.1 illustrates the interfaces between the Code Model Subsystem and the rest of the ATESSE system. A user interacts with the Code Model Subsystem through the Simulator Interface (SI) process which provides menus and forms for creating and compiling code models and linking them into a simulator executable.

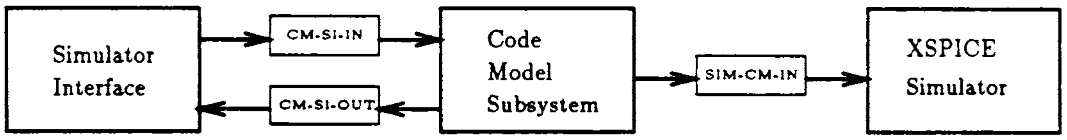

Figure 3.1 Code Model Subsystem External Interfaces.

Code Model Interface Specification files, Code Model Definition files, and User-Defined Node Definition files are created and read by the Simulator Interface editing facilities through the interfaces CM-SI-IN and CM-SI-OUT. The Code Model Subsystem is responsible for processing and compiling these files so that they can be linked with the simulator through interface SIM-CM-IN.

The XSPICE simulator and the Code Model Subsystem can also be used in a "stand-alone" fashion, independent of the ATESSE Simulator Interface process. In this case, the interfaces CM-SI-IN and CM-SI-OUT are replaced by the Operating System command line interface and text editing facilities.

3.2 Code Model Development

The following subsections detail the files and data involved in developing code models. A description of the steps required to create code models can be found in the Software Design Document for the XSPICE Code Model Subsystem of the Automatic Test Equipment Software Support Environment (ATESSE).

3.2.1 Model Directory Generator Files

The 'mkmoddir' utility creates a code model directory and installs a template Interface Specification File, template Model Definition File, and a Makefile. These files are created from the templates stored in /atesse/lib/cmt/mkmoddir and are placed in the newly created directory as:

```
Makefile
ifspec.ifs
cfunc.mod
```

3.2.2 Interface Specification File

A code model's external interface is described in an Interface Specification (IFS) file (ifspec.ifs). The IFS file contains the name of the code model, the name of its C function, and a description of valid ports, parameters, and static variables. The IFS file is a user-editable text file.

The information stored in an interface specification file is translated by the Code Model Preprocessor into data structures which are incorporated into the run-time XSPICE simulator executable. These data structures allow XSPICE to parse SPICE deck records associated with the code model and call the code model's C function, passing the expected model parameters and inputs and outputs as function arguments.

An Interface Specification File consists of a Name Table, one or more Port Tables, zero or more Parameter Tables, and zero or more Static Variable Tables. C-style (/* */) comments may be interspersed throughout the file.

3.2.2.1 Syntactic Conventions

3.2.2.1.1 Case Insensitivity

Keywords in the Interface Specification are recognized in any combination of lower and upper case. Case is preserved within string literals and for engineering suffixes (e.g. "12M" is read as "12 MEG" whereas "12m" is read as "12 milli").

3.2.2.1.2 Engineering Suffixes

Numeric constants may contain an optional scaling factor (engineering suffix). These largely correspond to standard SPICE scaling factors, but the Interface Specification File makes a distinction between capital M and lowercase m. All other suffixes are case insensitive. Table 3.1 summarizes the valid scaling factors:

Suffix	Scaling Factor
t	1e12
g	1e9
M (case sensitive)	1e6
MEG	1e6
k	1e3
m (case sensitive)	1e-3
u	1e-6
n	1e-9
p	1e-12
f	1e-15
mil	25.4e-6

Table 3.1 Engineering Suffixes.

Alphanumeric literals following an engineering suffix are ignored. Thus, the following all specify the same value: 2000.0, 2.0k, 2.0kOhms.

3.2.2.1.3 Comments

C-style comments may be added anywhere in the Interface Specification file. As in C, comments start with "/*" and are terminated by a "*/". Comments may not be nested. Hence, "/* /* this */ is an error */" consists of the comment "/* /* this */" followed by "is an error */". Comments may span multiple lines. Comments are treated as whitespace; they are ignored.

3.2.2.1.4 Ranges

The following range syntax is used whenever a range of values is required (for example, to specify the limits on a parameter value or to specify the minimum and maximum vector sizes for ports and parameters). A range is specified by a square bracket followed by a value separated by space from another value and terminated with a closing square bracket. For example, "[0 10]". The lower and upper bounds are inclusive; thus, the previous example is read as "the range of values from 0 to 10, inclusive". Either the lower or upper bound may be replaced by a hyphen ("-") to indicate that bound is unconstrained. Hence, "[10 -]" is read as "the range of values greater than or equal to 10". For a totally unconstrained range, a single hyphen with no surrounding brackets may be used. So, "-" is read as "any value".

3.2.2.1.5 Keywords

Interface Specification keywords all contain a colon as a suffix (e.g. "Port_Table:"). The colon may be separated from the alphanumeric portion of the keyword by any amount of whitespace (e.g. "Port_Table :"). Keywords always occur at the beginning of a line.

3.2.2.1.6 Table Format

The Interface Specification is made up of a number of tables. Each table is introduced by a table name keyword (e.g. "Port_Table:") and is comprised of a number of fields. Each field starts with a keyword and is followed by any number of values appropriate to that keyword. The first field of a table defines its width. All subsequent fields within the table must have the same number of elements. A table may be split up into any number of subtables by reintroducing the table name keyword:

```
Port_Table:

Name:      "in"     "out"    "inout"
Vector:    yes       no       yes
```

may be rewritten as:

```
Port_Table:

Name:      "in"     "out"
Vector:    yes      no

Port_Table:

Name:      "inout"
Vector:    yes
```

3.2.2.1.7 Value Literals

Whenever a numeric, boolean, complex, or string value is required, the value is written as described in the following sections.

3.2.2.1.7.1 Integer Literals

Integer literals are sequences of digits followed by an optional engineering suffix. The digits and suffix may not be separated by whitespace. The following examples show valid integer literals:

```
252     1K     1GB
```

The following examples are **invalid** integer literals:

```
24e3    -   Integer constants may not contain an exponent
1 A     -   Digits and suffix may not be separated by space
1.1     -   Integers may not contain a decimal point
```

3.2.2.1.7.2 Real Literals

Real literals are sequences of digits with an optional decimal point, a power of ten exponent, and an optional engineering suffix. A real literal may not contain any embedded whitespace. The following examples show valid real literals:

```
252.2    2e20   6.02e23   12.3MB    23.4mA
```

The following examples are **invalid** real literals:

```
12 e10    -   Embedded whitespace is invalid
12e10 mA  -   Embedded whitespace is invalid
```

3.2.2.1.7.3 Complex Literals

A complex literal is delimited by angle brackets (< >) and contains two real literals (the real and imaginary parts, in cartesian space) separated by a comma and/or whitespace. The following examples show valid complex literals:

```
<1 0>    <1.1 -2.2>    <2.2,1>    <2, 33>
```

3.2.2.1.7.4 Boolean Literals

A boolean literal is one of the keywords: **True, False, Yes, No. True** is equivalent to **Yes. False** is equivalent to **No.** The interpretation of these keywords is case insensitive.

3.2.2.1.7.5 String Literals

A string literal is a sequence of characters delimited by double quotes. Embedded double quotes must be escaped (preceded by a backslash (e.g. "\"). The following examples show valid string literals:

```
"Hello world"
"Embedded single quotes (') are ok"
"Embedded double quotes (\") require an escape"
```

The following example is an **invalid** string literal:

```
"Non-escaped embedded double quotes (") are NOT ok"
```

3.2.2.2 Name Table

The Name Table is introduced by the "Name_Table:" keyword. It defines the code model's C function name, its SPICE deck model name, and an optional textual description. The following sections define the valid fields that may be specified in the Name Table as shown in Table 3.2.

Field name	Value	Purpose
C_Function_Name:	Identifier	Name of C implementation function
Description:	String	Short description of the Code Model
SPICE_Model_Name:	Identifier	Name of SPICE .model card

Table 3.2 Interface Specification Name Table.

3.2.2.2.1 Description

The description string is used to describe the purpose and function of the code model. It is introduced by the "Description:" keyword followed by a string literal.

3.2.2.2.2 Model Name

The Model name is a valid SPICE identifier which will be used on SPICE deck .model records to refer to this code model. It may or may not be the same as the C function name. It is introduced by the "SPICE_Model_Name:" keyword followed by a valid SPICE identifier.

3.2.2.2.3 C Function Name

The C Function name is a valid C identifier which is the name of the main entry point (function) for the code model. It may or may not be the same as the SPICE model name. It is introduced by the "C_Function_Name:" keyword followed by a valid C identifier. To reduce the chance of name conflicts (not only with other code models, but also with variables and functions defined in the XSPICE simulator core), it is recommended that user-written code model names use the prefix "UCM_" for "user code model" (e.g. "UCM_gizmo"), or use the user's initials. The following prefixes are used by the XSPICE simulator core and should not be used for user code models: ASRC, BJT, BSIM, CAP, CCCS, CCVS, CKT, CM, CP, CSW, DEV, DIO, ENH, EVT, FTE, HLP, ICM, IDN, IND, INP, IPC, ISRC, JFET, MES, MFB, MIF, MOS1, MOS2, MOS2, NI, RES, SMP, SW, TRA, URC, VCCS, VCVS, VSRC.

3.2.2.3 Port Table

The port table is introduced by the "Port_Table:" keyword. It defines the set of valid ports available to the code model. Table 3.3 and the following sections define the valid fields that may be specified in the Port Table.

Field name	Value	Purpose
Allowed_Type:	Port Type Set	Specify allowed port types
Vector:	Boolean	Is this port a bus?
Vector_Bounds:	Int Range	Valid sizes for the port
Port_Name:	Identifier	Name of the port
Default_Type:	Port Type	Specify default port type
Description:	String	Short description of the port
Direction:	Direction	Specify dataflow of this port
Null_Allowed:	Boolean	Can this port be left unconnected?

Table 3.3 Interface Specification Port Table.

3.2.2.3.1 Description

The description string is used to describe the purpose and function of the port. It is introduced by the "Description:" keyword followed by a string literal.

3.2.2.3.2 Port Name

The port name must be a valid SPICE identifier. It is introduced by the "Port_Name:" keyword followed by a valid identifier.

3.2.2.3.3 Direction

The direction of a port specifies the dataflow direction through the port. A direction value must be one of in, out or inout. It is introduced by the "Direction:" keyword followed by a valid direction value.

3.2.2.3.4 Allowed Types

A port must specify the allowed types to which it can be connected. An allowed type value must be a list of type names (a blank or comma separated list of names delimited by square brackets, e.g. "[v vd i id]") from Table 3.4.

Type Name	Valid Directions	Description
d	in, out	digital
g	inout	Conductance
gd	inout	Differential Conductance
h	inout	Resistance
hd	inout	Differential Resistance
i	in, out	Current
id	in, out	Differential Current
v	in, out	Voltage
vd	in, out	Differential Voltage
vnam	in	Current through named voltage source
<identifier>	in, out	User-defined type

Table 3.4 Port Types.

User-defined type names are any valid SPICE identifier except for the predefined types listed above.

The allowed types are introduced by the "Allowed_Types:" keyword followed by a valid type set as described above.

3.2.2.3.5 Default Type

The default type field specifies the type used for the port when no type is explicitly specified on a SPICE card. The default type is introduced with the "Default_Type:" keyword followed by a valid type. The type name must be a member of the Allowed_Types set.

3.2.2.3.6 Vector

A port can have any number of subports. A port which is a vector can be thought of as a bus. The vector field is introduced with the "Vector:" keyword followed by a boolean value: yes, true, no, or false. The values yes and true are equivalent and specify that this port is a vector. Likewise, no and false specify that the port is not a vector. Vector ports must have a corresponding Vector Bounds field which specifies valid sizes of the vector port.

3.2.2.3.7 Vector Bounds

If a port is a vector, its valid size must be specified by the Vector Bounds field. The Vector Bounds field specifies the upper and lower bounds on the size of the vector. The Vector Bounds field is introduced by the "Vector_Bounds:" keyword followed by a range of integers. If the range is unconstrained or the associated port is not a vector, the vector bounds may be specified by a hyphen ("-"). The lower bound of the vector specifies the minimum number of elements in the vector, and the upper bound specifies the maximum number of elements.

3.2.2.3.8 Null Allowed

In some cases, it is desirable to permit a port to be left unconnected. The Null Allowed field specifies whether it is an error to leave a port unconnected. The Null Allowed field is introduced by the "Null_Allowed:" keyword and is followed by a boolean constant: yes, true, no, or false. The values yes and true are equivalent and specify that it is legal to leave this port unconnected. No and false specify that the port must be connected.

3.2.2.4 Parameter Table

The parameter table is introduced by the "Parameter_Table:" keyword. It defines the set of valid parameters available to the code model. Table 3.5 and the following sections define the valid fields that may be specified in the Parameter Table.

Field name	Value	Purpose
Vector:	Boolean	Is the parameter a vector?
Vector_Bounds:	Int Range	Valid sizes for the parameter or vector
Data_Type:	Data Type	Specifies underlying data type for the parameter values
Default_Value:	<context dependent>	If explicit value not specified in SPICE deck, use this value
Description:	String	Short description of the parameter
Limits:	Range	Valid range of values
Null_Allowed:	Boolean	Can this parameter remain unspecified?
Parameter_Name:	Identifier	Name of the parameter

Table 3.5 Interface Specification Parameter Table.

3.2.2.4.1 Description

The description string is used to describe the purpose and function of the parameter. It is introduced by the "Description:" keyword followed by a string literal.

3.2.2.4.2 Parameter Name

The parameter name must be a valid SPICE identifier which will be used on SPICE deck .model cards to refer to this parameter. It is introduced by the "Parameter_Name:" keyword followed by a valid SPICE identifier.

3.2.2.4.3 Data Type

The parameter's data type is specified by the Data Type field. This data type corresponds to the underlying C data type (e.g., double), not the conceptual type of the parameter (e.g., voltage). The Data Type field is introduced by the keyword "Data_Type:" and is followed by a valid data type. Valid data types and their corresponding C implementations are shown in the Table 3.6.

3.2.2.4.4 Default Value

If Null Allowed is true, a default value may be specified. This value is supplied for the parameter in the event that the SPICE deck .model line does not supply a value for the parameter. The default value must be of the correct type. The Default Value field is introduced by the "Default_Value:" keyword followed by a numeric, boolean, complex or string literal, as appropriate.

IFS Data Type	C Data Type
boolean	Boolean_t
complex	Complex_t
int	int
real	double
string	char*

Table 3.6 Parameter Types.

3.2.2.4.5 Limits

Integer and Real parameters may be constrained to accept a limited range of values. The parameter value limits are introduced by the "Limit:" keyword followed by a range.

3.2.2.4.6 Vector

The Vector field is used to specify whether a parameter is a vector or a scalar. Like the Port Vector field, it is introduced by the "Vector:" keyword and followed by a boolean value. True or yes specifies that the parameter is a vector. False or no specifies that it is a scalar.

3.2.2.4.7 Vector Bounds

The valid sizes for a vector parameter are specified in the same manner as those for ports. In addition to using a numeric range to specify valid vector bounds, it is possible to refer to the name of a port. When a parameter's vector bounds are specified in this way, the bounds on the vector size will be the same as the bounds of the specified vector port.

3.2.2.4.8 Null Allowed

The Null Allowed field is introduced by the "Null Allowed:" keyword followed by a Boolean literal. A value of true or yes specifies that it is valid for the corresponding SPICE deck .model card to omit a value for this parameter. If the parameter is omitted, the default value is used. If there is no default value, an undefined value is passed to the code model. If the value of Null Allowed is false or no, then XSPICE will flag an error if the SPICE deck .model card includes no value for this parameter.

3.2.2.5 Static Variable Table

The Static Variable table is introduced by the "Static_Variable_Table:" keyword. It defines the set of valid Static variables available to the code model. Table 3.7 and the following sections define the valid fields that may be specified in the Static Variable Table.

Field name	Value	Purpose
Description:	String	Short description of the static variable
Static_Var_Name:	Identifier	Name of the static variable
Data_Type:	Data Type	Type of the variable

Table 3.7 Interface Specification Static Variable Table.

3.2.2.5.1 Description

The description string is used to describe the purpose and function of the static variable. It is introduced by the "Description:" keyword followed by a string literal.

3.2.2.5.2 Name

The Static variable name must be a valid C identifier which will be used in the code model to refer to this static variable. It is introduced by the "Static_Var_Name:" keyword followed by a valid C identifier.

3.2.2.5.3 Data Type

The static variable's data type is specified by the Data Type field. This data type corresponds to the underlying C data type (e.g., double), not the conceptual type of the parameter (e.g., voltage). The Data Type field is introduced by the keyword "Data_Type:" followed by a valid data type. Table 3.8 provides a summary of valid data types and their corresponding C implementations. Note that the type "pointer" is used for Static Var arrays. The code model allocates space for the array and assigns the pointer to the Static Var so that it can be retrieved on subsequent code model calls.

3.2.3 Compiled Interface Specification File

The Interface Specification File described above is translated by the Code Model Preprocessor (cmpp -ifs) into a C language file (ifspec.c) which is compiled and linked into the XSPICE simulator executable. The specification for this file is given in the Interface Design

IFS Data Type	C Data Type
boolean	Boolean_t
complex	Complex_t
int	int
real	double
string	char*
pointer	void*

Table 3.8 Static Variable Types.

Document for the XSPICE Simulator of the Automatic Test Equipment Software Support Environment (ATESSE).

3.2.4 Model Definition File

The model definition file (cfunc.mod) defines the C function which implements the code model. It is processed by the Code Model Preprocessor (cmpp -mod) to produce a C language file which is compiled and linked into the XSPICE simulator executable. A Model Definition File is a C language file using special "accessor macros" which are expanded by the Code Model Preprocessor.

3.2.5 Translated Model Definition File

The translated model definition file (cfunc.c) is the generated C file produced by the code model preprocessor (cmpp -mod) from the model definition file (cfunc.mod). Each of the accessor macros is translated as described in the Interface Design Document for the XSPICE Simulator of the Automatic Test Equipment Software Support Environment (ATESSE).

3.2.6 Accessor Macros

Table 3.9 describes the accessor macros available to the Model Definition File programmer and their C types. Those accessor macros with types labeled CD (context dependent) return the type as specified for that port or parameter in the Interface Specification File. Arguments listed with "[i]" take an optional square bracket delimited index in the case that the corresponding port or parameter is a vector. The index may be any C expression - possibly involving calls to other accessor macros (e.g. "OUTPUT(out[PORT_SIZE(out) - 1])").

Name	Type	Args	Description
AC_GAIN	Complex_t	y[i],x[i]	AC gain of output y with respect to input x
ANALYSIS	enum	none	Type of analysis: DC, AC, TRANSIENT
ARGS	Mif_Private_t	none	Standard argument to all code model functions
CALL_TYPE	enum	none	Type of model evaluation call: ANALOG or EVENT
INIT	Boolean_t	none	Is this the first call to the model?
INPUT	double or void *	name[i]	Value of analog input port, or value of structure pointer for user-defined node port.
INPUT_STATE	enum	name[i]	State of a digital input: ZERO, ONE, or UNKNOWN.
INPUT_STRENGTH	enum	name[i]	Strength of digital input: STRONG, RESISTIVE, HI_IMPEDANCE, or UNDETERMINED
INPUT_TYPE	char *	name[i]	The port type of the input
LOAD	double	name[i]	The digital load value placed on a port by this model.
MESSAGE	char *	name[i]	A message output by a model on an event-driven node.
NEW_TIMEPOINT	Boolean_t	<none>	First call at this analysis point?
OUTPUT	double or void *	name[i]	Value of the analog output port or value of structure pointer for user-defined node port.
OUTPUT_CHANGED	Boolean_t	name[i]	Has a new value been assigned to this event-driven output by the model?
OUTPUT_DELAY	double	name[i]	Delay in seconds for an event-driven output
OUTPUT_STATE	enum	name[i]	State of a digital output: ZERO, ONE, or UNKNOWN.
OUTPUT_STRENGTH	enum	name[i]	Strength of digital output: STRONG, RESISTIVE, HI_IMPEDANCE, or UNDETERMINED
OUTPUT_TYPE	char *	name[i]	The port type of the output
PARAM	CD	name[i]	Value of the parameter
PARAM_NULL	Boolean_t	name[i]	Was the parameter not included on the SPICE .model card?
PARAM_SIZE	int	name	Size of parameter vector
PARTIAL	double	y[i],x[i]	Partial derivative of output y with respect to input x
PORT_NULL	Boolean_t	name[i]	Has this port been specified as unconnected?
PORT_SIZE	int	name	Size of port vector
RAD_FREQ	double	<none>	Current analysis frequency in radians per second
STATIC_VAR	CD	name	Value of a static variable
STATIC_VAR_SIZE	int	name	Size of static var vector (currently unused).
T [<n>]	double	<none>	History of the previous 8 analysis times (TIME = T[0])
TEMPERATURE	double	<none>	Current analysis temperature
TIME	double	<none>	Current analysis time (same as T[0])
TOTAL_LOAD	double	name[i]	The total of all loads on the node attached to this event-driven port.

Table 3.9 Model Definition File Macros.

3.3 User-Defined Node Development

The following subsections detail the files and data involved in developing user-defined nodes. A description of the steps required to create user-defined nodes can be found in the Software Design Document for the XSPICE Code Model Subsystem of the Automatic Test Equipment Software Support Environment (ATESSE).

3.3.1 User-Defined Node Directory Generator Files

The 'mkudndir' utility creates a user-defined node directory and installs a template User-Defined Node Definition File, and a Makefile. These files are created from the templates stored in `xspice/lib/cmt/mkudndir` and are placed in the newly created directory as:

```
Makefile
udnfunc.c
```

3.3.2 User-Defined Node Definition File

The User-Defined Node Definition file (udnfunc.c) defines the C functions which implement basic operations on user-defined nodes such as data structure creation, initialization, copying, and comparison. Unlike the Model Definition File which uses the Code Model Preprocessor to translate Accessor Macros, the User-Defined Node Definition File is a pure C language file. This file uses macros to isolate the user from data structure definitions. These macros are defined in a standard header file (EVTudn.h) and translations are performed by the standard C preprocessor, rather than by the Code Model Preprocessor.

When a directory is created for a new user-defined node with 'mkudndir', a structure of type 'Evt_Udn_Info_t' is placed at the bottom of the User-Defined Node Definition File. This structure contains the type name for the node, a description string, and pointers to each of the functions that define the node type. This structure is complete except for a text string that describes the node type. This string is stubbed out and may be edited by the user if desired.

The functions (required and optional) that define a user-defined node are listed below. Optional functions can be deleted from the udnfunc.c file template created by 'mkudndir' and the corresponding pointers in the Evt_Udn_Info_t structure can be changed to NULL.

Required functions:

create	Allocate data structure used as input and output of code models.
initialize	Set structure to appropriate initial value for first use as model input.
copy	Make a copy of a structure's contents into created but possibly uninitialized structure.
compare	Determine if two structures are equal in value.

Optional functions:

dismantle	Free allocations referenced by structure (but not structure itself).
invert	Invert logical value of structure.
resolve	Determine the resultant when multiple outputs are connected to a node.
plot_val	Output a real value for specified structure component for use in plotting.
print_val	Output a string value for specified structure component for use in printing.
ipc_val	Output a binary representation for the data and an integer giving its size.

The required actions for each of these functions are described in the following subsections. In each function, 'mkudndir' replaces the XXX with the node type name specified by the user when mkudndir is invoked. The macros used in implementing the functions are described in a later section.

3.3.2.1 Function udn_XXX_create

Allocate space for the data structure defined for the user-defined node to pass data between models. Then, assign pointer created by the storage allocator (e.g., malloc) to MALLOCED_PTR.

3.3.2.2 Function udn_XXX_initialize

Assign STRUCT_PTR to a pointer variable of the defined type, and then initialize the value of the structure.

3.3.2.3 Function udn_XXX_compare

Assign STRUCT_PTR_1 and STRUCT_PTR_2 to pointer variables of the defined type. Compare the two structures and assign either TRUE or FALSE to EQUAL.

3.3.2.4 Function udn_XXX_copy

Assign INPUT_STRUCT_PTR and OUTPUT_STRUCT_PTR to pointer variables of the defined type, and then copy the elements of the input structure to the output structure.

3.3.2.5 Function udn_XXX_dismantle

Assign STRUCT_PTR to a pointer variable of defined type, and then free any allocated substructures (but not the structure itself!). If there are no substructures, the body of this function may be left null.

3.3.2.6 Function udn_XXX_invert

Assign STRUCT_PTR to a pointer variable of the defined type, and then invert the logical value of the structure.

3.3.2.7 Function udn_XXX_resolve

Assign INPUT_STRUCT_PTR_ARRAY to a variable declared as an array of pointers of the defined type - e.g.:

```
<type> **struct_array;
struct_array = INPUT_STRUCT_PTR_ARRAY;
```

Then, the number of elements in the array may be determined from the integer valued INPUT_STRUCT_PTR_ARRAY_SIZE macro.

Assign OUTPUT_STRUCT_PTR to a pointer variable of the defined type.

Scan through the array of structures, and compute the resolved value and assign it into the output structure.

3.3.2.8 Function udn_XXX_plot_val

Assign STRUCT_PTR to a pointer variable of the defined type. Then, access the member of the structure specified by the string in STRUCT_MEMBER_ID, and assign some real valued quantity for this member to PLOT_VALUE.

Name	Type	Description
MALLOCED_PTR	void *	Assign pointer to alloced structure to this macro
STRUCT_PTR	void *	A pointer to a structure of the defined type
STRUCT_PTR_1	void *	A pointer to a structure of the defined type
STRUCT_PTR_2	void *	A pointer to a structure of the defined type
EQUAL	Mif_Boolean_t	Assign TRUE or FALSE to this macro according to the results of structure comparison
INPUT_STRUCT_PTR	void *	A pointer to a structure of the defined type
OUTPUT_STRUCT_PTR	void *	A pointer to a structure of the defined type
INPUT_STRUCT_PTR_ARRAY	void **	An array of pointers to structures of the defined type
INPUT_STRUCT_PTR_ARRAY_SIZE	int	The size of the array
STRUCT_MEMBER_ID	char *	A string naming some part of the structure
PLOT_VAL	double	The value of the specified structure member for plotting purposes
PRINT_VAL	char *	The value of the specified structure member for printing purposes

Table 3.10 User-Defined Node Macros.

3.3.2.9 Function udn_XXX_print_val

Assign STRUCT_PTR to a pointer variable of the defined type. Then, access the member of the structure specified by the string in STRUCT_MEMBER_ID, and assign some string valued quantity for this member to PRINT_VALUE.

If the string is not static, a new string should be allocated on each call. Do not free the allocated strings.

3.3.2.10 Function udn_XXX_ipc_val

Use STRUCT_PTR to access the data, and then assign a pointer to a binary representation of the data to IPC_VAL, and assign an integer giving the size of this binary representation to IPC_VAL_SIZE. Typically STRUCT_PTR is assigned directly to IPC_VAL, and the sizeof() operator is used to get the structure size and assign it to IPC_VAL_SIZE.

3.3.3 Accessor Macros

The macros used in the User-Defined Node Definition file to access and assign data are defined in Table 3.10. These macros do not take arguments. The translations of the macros, and of macros used in the function argument lists, are defined in the document Interface Design Document for the XSPICE Simulator of the Automatic Test Equipment Software Support Environment (ATESSE).

3.4 Simulator Development

The following subsections detail the files and data involved in developing a new XSPICE executable. A description of the steps required to create new executables can be found in the Software Design Document for the XSPICE Code Model Subsystem of the Automatic Test Equipment Software Support Environment (ATESSE).

3.4.1 Simulator Directory Generator Files

The 'mkmoddir' utility creates a simulator directory and installs a template Model Path File, template User-Defined Node Path File, and a Makefile. These files are created from the templates stored in /atesse/lib/cmt/mksimdir and are placed in the newly created directory as:

```
Makefile
modpath.lst
udnpath.lst
```

3.4.2 Model Path File

The Model Path file (modpath.lst) is used to specify the set of code models to be linked into a given XSPICE simulator executable. It is a text file, with a complete pathname, one per line, of each code model's directory. Each of these directories is assumed to contain an Interface Specification file named ifspec.ifs and object files for the compiled Interface Specification and Model Definition File.

3.4.3 User-Defined Node Path File

The User-Defined Node Path file (udnpath.lst) is used to specify the set of event-driven, user-defined node types to be linked into a given XSPICE simulator executable. It is a text file, with a complete pathname, one per line, of each user-defined node's directory. Each of these directories is assumed to contain an object file for the compiled User-Defined Node Definition File.

3.5 Code Model Library

The interface for each of the prewritten XSPICE code models is given in the following subsections in the form of an Interface Specification file. A definition of the model's operation can be found in the Software Design Document for the XSPICE Code Model Subsystem of the Automatic Test Equipment Software Support Environment (ATESSE).

3.5.1 Analog Models

3.5.1.1 Gain

NAME_TABLE:

C_Function_Name:	cm_gain
Spice_Model_Name:	gain
Description:	"A simple gain block"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	in_offset	gain	out_offset
Description:	"input offset"	"gain"	"output offset"
Data_Type:	real	real	real
Default_Value:	0.0	1.0	0.0
Limits:	-	-	-
Vector:	no	no	no
Vector_Bounds:	-	-	-
Null_Allowed:	yes	yes	yes

3.5.1.2 Summer

NAME_TABLE:

C_Function_Name:	cm_summer
Spice_Model_Name:	summer
Description:	"summer block"

PORT_TABLE:

Port_Name:	in	out
Description:	"input array"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	yes	no
Vector_Bounds:	[2 -]	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	in_offset	in_gain
Description:	"input offset array"	"input gain array"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	yes	yes
Vector_Bounds:	in	in
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	out_gain	out_offset
Description:	"output gain"	"output offset"
Data_Type:	real	real
Default_Value:	1.0	0.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.1.3 Multiplier

NAME_TABLE:

C_Function_Name:	cm_mult
Spice_Model_Name:	mult
Description:	"multiplier block"

PORT_TABLE:

Port_Name:	in	out
Description:	"input array"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	yes	no
Vector_Bounds:	[2 -]	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	in_offset	in_gain
Description:	"input offset array"	"input gain array"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	yes	yes
Vector_Bounds:	in	in
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	out_gain	out_offset
Description:	"output gain"	"output offset"
Data_Type:	real	real
Default_Value:	1.0	0.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.1.4 Divider

NAME_TABLE:

C_Function_Name:	cm_divide
Spice_Model_Name:	divide
Description:	"divider block"

PORT_TABLE:

Port_Name:	num	den	out
Description:	"numerator"	"denominator"	"output"
Direction:	in	in	out
Default_Type:	v	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no	no
Vector_Bounds:	-	-	-
Null_Allowed:	no	no	no

PARAMETER_TABLE:

Parameter_Name:	num_offset	num_gain
Description:	"numerator offset"	"numerator gain"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	den_offset	den_gain
Description:	"denominator offset"	"denominator gain"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	den_lower_limit	den_domain
Description:	"denominator lower limit"	"denominator smoothing domain"
Data_Type:	real	real
Default_Value:	1.0e-10	1.0e-16
Limits:	[1.0e-10 -]	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	fraction
Description:	"smoothing fraction/absolute value switch"
Data_Type:	boolean
Default_Value:	false
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	out_gain	out_offset
Description:	"output gain"	"output offset"
Data_Type:	real	real
Default_Value:	1.0	0.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.1.5 Limiter

NAME_TABLE:

C_Function_Name:	cm_limit
Spice_Model_Name:	limit
Description:	"limit block"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	in_offset	gain
Description:	"input offset"	"gain"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	out_lower_limit	out_upper_limit
Description:	"output lower limit"	"output upper limit"
Data_Type:	real	real
Default_Value:	-	-
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	limit_range	fraction
Description:	"upper & lower sm. range"	"smoothing percent/abs switch"
Data_Type:	real	boolean
Default_Value:	1.0e-6	FALSE
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.1.6 Controlled Limiter

NAME_TABLE:

C_Function_Name:	cm_climit
Spice_Model_Name:	climit
Description:	"controlled limiter block"

PORT_TABLE:

Port_Name:	in	cntl_upper
Description:	"input"	"upper lim. control input"
Direction:	in	in
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id,vnam]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PORT_TABLE:

Port_Name:	cntl_lower	out
Description:	"lower limit control input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	in_offset	gain
Description:	"input offset"	"gain"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	upper_delta	lower_delta
Description:	"output upper delta"	"output lower delta"
Data_Type:	real	real
Default_Value:	0.0	0.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	limit_range	fraction
Description:	"upper & lower sm. range"	"smoothing %/abs switch"
Data_Type:	real	boolean
Default_Value:	1.0e-6	FALSE
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.1.7 Piecewise Linear Controlled Source

NAME_TABLE:

C_Function_Name:	cm_pwl
Spice_Model_Name:	pwl
Description:	"piecewise linear controlled source"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	x_array	y_array
Description:	"x-element array"	"y-element array"
Data_Type:	real	real
Default_Value:	-	-
Limits:	-	-
Vector:	yes	yes
Vector_Bounds:	[2 -]	[2 -]
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	input_domain	fraction
Description:	"input sm. domain"	"smoothing %/abs switch"
Data_Type:	real	boolean
Default_Value:	0.01	TRUE
Limits:	[1e-12 0.5]	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

STATIC_VAR_TABLE:

Static_Var_Name:	last_x_value
Data_Type:	pointer

Description: "iteration holding variable for limiting"

STATIC_VAR_TABLE:

Static_Var_Name:	x	y
Data_Type:	pointer	pointer
Description:	"x-coefficient array"	"y-coefficient array"

3.5.1.8 Analog Switch

NAME_TABLE:

C_Function_Name:	cm_aswitch
Spice_Model_Name:	aswitch
Description:	"analog switch"

PORT_TABLE:

Port_Name:	cntl_in	out
Description:	"input"	"resistive output"
Direction:	in	inout
Default_Type:	v	gd
Allowed_Types:	[v,vd,i,id,vnam]	[gd]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	cntl_off	cntl_on
Description:	"control 'off' val"	"control 'on' val"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	log	r_off
Description:	"Log-linear switch"	"off resistance"
Data_Type:	boolean	real
Default_Value:	TRUE	1.0e12
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	r_on
Description:	"on resistance"
Data_Type:	real
Default_Value:	1.0
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.1.9 Zener Diode

NAME_TABLE:

C_Function_Name:	cm_zener
Spice_Model_Name:	zener
Description:	"zener diode"

PORT_TABLE:

Port_Name:	z
Description:	"zener"
Direction:	inout
Default_Type:	gd
Allowed_Types:	[gd]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	no

PARAMETER_TABLE:

Parameter_Name:	v_breakdown	i_breakdown
Description:	"breakdown voltage"	"breakdown current"
Data_Type:	real	real
Default_Value:	-	2e-2
Limits:	[1e-6 1e6]	[1e-9 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	yes

PARAMETER_TABLE:

Parameter_Name:	r_breakdown	i_rev
Description:	"breakdown resistance"	"reverse current"
Data_Type:	real	real
Default_Value:	1.0	1e-6
Limits:	[1e-12 -]	[1e-9 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	i_sat	n_forward
Description:	"saturation current"	"forward emission co"
Data_Type:	real	real
Default_Value:	1e-12	1.0
Limits:	[1e-15 -]	[.1 10]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	limit_switch
Description:	"switch for on-board limiting (convergence aid)"
Data_Type:	boolean
Default_Value:	FALSE
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

STATIC_VAR_TABLE:

Static_Var_Name:	previous_voltage
Data_Type:	pointer
Description:	"iteration holding variable for limiting"

3.5.1.10 Current Limiter

NAME_TABLE:

C_Function_Name:	cm_ilimit
Spice_Model_Name:	ilimit
Description:	"current limiter block"

PORT_TABLE:

Port_Name:	in	pos_pwr
Description:	"input"	"positive power supply"
Direction:	in	inout
Default_Type:	v	g
Allowed_Types:	[v,vd,i,id,vnam]	[g,gd]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	yes

PORT_TABLE:

Port_Name:	neg_pwr	out
Description:	"negative power supply"	"output"
Direction:	inout	inout
Default_Type:	g	g
Allowed_Types:	[g,gd]	[g,gd]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	no

PARAMETER_TABLE:

Parameter_Name:	in_offset	gain
Description:	"input offset"	"gain"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	r_out_source	r_out_sink
Description:	"sourcing resistance"	"sinking resistance"

Data_Type:	real	real
Default_Value:	1.0	1.0
Limits:	[1e-9 1e9]	[1e-9 1e9]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	i_limit_source	i_limit_sink
Description:	"current sourcing limit"	"current sinking limit"
Data_Type:	real	real
Default_Value:	10.0e-3	10.0e-3
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	v_pwr_range	i_source_range
Description:	"pwr. smoothing range"	"sourcing cur sm. rng"
Data_Type:	real	real
Default_Value:	1e-6	1e-9
Limits:	[1e-15 -]	[1e-15 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	i_sink_range	r_out_domain
Description:	"sinking cur sm. rng"	"output resistance sm. domain"
Data_Type:	real	real
Default_Value:	1e-9	1e-9
Limits:	[1e-15 -]	[1e-15 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.1.11 Hysteresis Block

NAME_TABLE:

C_Function_Name:	cm_hyst
Spice_Model_Name:	hyst
Description:	"hysteresis block"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	in_low	in_high
Description:	"input low value"	"input high value"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	hyst	out_lower_limit
Description:	"hysteresis"	"output lower limit"
Data_Type:	real	real
Default_Value:	0.1	0.0
Limits:	[0 -]	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	out_upper_limit	input_domain
Description:	"output upper limit"	"input smoothing domain"
Data_Type:	real	real

Default_Value:	1.0	0.01
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	fraction
Description:	"smoothing percent/abs switch"
Data_Type:	boolean
Default_Value:	TRUE
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.1.12 Differentiator

NAME_TABLE:

C_Function_Name:	cm_d_dt
Spice_Model_Name:	d_dt
Description:	"differentiator block"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	out_offset	gain
Description:	"output offset"	"gain"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	out_lower_limit	out_upper_limit
Description:	"output lower limit"	"output upper limit"
Data_Type:	real	real
Default_Value:	-	-
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	limit_range
Description:	"upper & lower sm. range"
Data_Type:	real
Default_Value:	1.0e-6
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.1.13 S-Domain Transfer Function

NAME_TABLE:

Spice_Model_Name:	s_xfer
C_Function_Name:	cm_s_xfer
Description:	"s-domain transfer function block"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	in_offset	gain
Description:	"input offset"	"gain"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	num_coeff	den_coeff
Description:	"numerator poly coef"	"denominator poly coef"
Data_Type:	real	real
Default_Value:	-	-
Limits:	-	-
Vector:	yes	yes
Vector_Bounds:	[1 -]	[1 -]
Null_Allowed:	yes	yes

```
PARAMETER_TABLE:


Parameter_Name:     int_ic
Description:        "int stage init. cond"
Data_Type:          real
Default_Value:      0.0
Limits:             -
Vector:             yes
Vector_Bounds:      -
Null_Allowed:       yes

PARAMETER_TABLE:


Parameter_Name:     denormalized_freq
Description:        "freq. (rads/s) at which to denormalize coeffs"
Data_Type:          real
Default_Value:      1.0
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       yes
```

3.5.1.14 Slew Rate Block

NAME_TABLE:

C_Function_Name:	cm_slew
Spice_Model_Name:	slew
Description:	"a simple slew rate follower block"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_slope	fall_slope
Description:	"rising slew limit"	"falling slew limit"
Data_Type:	real	real
Default_Value:	1.0e9	1.0e9
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.1.15 Inductive Coupling

NAME_TABLE:

C_Function_Name:	cm_lcouple
Spice_Model_Name:	lcouple
Description:	"inductive coupling (for use with 'core' model)"

PORT_TABLE:

Port_Name:	l	mmf_out
Description:	"inductor"	"mmf output (in Ampere-turns)"
Direction:	inout	inout
Default_Type:	hd	hd
Allowed_Types:	[h,hd]	[hd]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	num_turns
Description:	"number of inductor turns"
Data_Type:	real
Default_Value:	1.0
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.1.16 Magnetic Core

NAME_TABLE:

C_Function_Name:	cm_core
Spice_Model_Name:	core
Description:	"magnetic core"

PORT_TABLE:

Port_Name:	mc
Description:	"magnetic core"
Direction:	inout
Default_Type:	gd
Allowed_Types:	[g,gd]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	no

PARAMETER_TABLE:

Parameter_Name:	H_array	B_array
Description:	"magnetic field array"	"flux density array"
Data_Type:	real	real
Default_Value:	-	-
Limits:	-	-
Vector:	yes	yes
Vector_Bounds:	[2 -]	[2 -]
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	area	length
Description:	"cross-sectional area"	"core length"
Data_Type:	real	real
Default_Value:	-	-
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	input_domain
Description:	"input sm. domain"
Data_Type:	real

Default_Value:	0.01
Limits:	[1e-12 0.5]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	fraction
Description:	"smoothing fractional/abs switch"
Data_Type:	boolean
Default_Value:	TRUE
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	mode
Description:	"mode switch (1 = pwl, 2 = hyst)"
Data_Type:	int
Default_Value:	1
Limits:	[1 2]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	in_low	in_high
Description:	"input low value"	"input high value"
Data_Type:	real	real
Default_Value:	0.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	hyst	out_lower_limit
Description:	"hysteresis"	"output lower limit"
Data_Type:	real	real
Default_Value:	0.1	0.0
Limits:	[0 -]	-
Vector:	no	no
Vector_Bounds:	-	-

Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	out_upper_limit
Description:	"output upper limit"
Data_Type:	real
Default_Value:	1.0
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.1.17 Controlled Sine Wave Oscillator

NAME_TABLE:

C_Function_Name:	cm_sine
Spice_Model_Name:	sine
Description:	"controlled sine wave oscillator"

PORT_TABLE:

Port_Name:	cntl_in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	cntl_array	freq_array
Description:	"control in array"	"frequency array"
Data_Type:	real	real
Default_Value:	0.0	1.0e3
Limits:	-	[0 -]
Vector:	yes	yes
Vector_Bounds:	[2 -]	[2 -]
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	out_low	out_high
Description:	"output low value"	"output high value"
Data_Type:	real	real
Default_Value:	-1.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.1.18 Controlled Triangle Wave Oscillator

NAME_TABLE:

C_Function_Name:	cm_triangle
Spice_Model_Name:	triangle
Description:	"controlled triangle wave oscillator"

PORT_TABLE:

Port_Name:	cntl_in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	cntl_array	freq_array
Description:	"control in array"	"frequency array"
Data_Type:	real	real
Default_Value:	0.0	1.0e3
Limits:	-	[0 -]
Vector:	yes	yes
Vector_Bounds:	[2 -]	[2 -]
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	out_low	out_high
Description:	"output low value"	"output high value"
Data_Type:	real	real
Default_Value:	-1.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	duty_cycle
Description:	"rise time duty cycle"
Data_Type:	real
Default_Value:	0.5
Limits:	[1e-6 .999999]

Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.1.19 Controlled Square Wave Oscillator

NAME_TABLE:

C_Function_Name:	cm_square
Spice_Model_Name:	square
Description:	"controlled square wave oscillator"

PORT_TABLE:

Port_Name:	cntl_in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,i,id,vnam]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	cntl_array	freq_array
Description:	"control in array"	"frequency array"
Data_Type:	real	real
Default_Value:	0.0	1.0e3
Limits:	-	[0 -]
Vector:	yes	yes
Vector_Bounds:	[2 -]	[2 -]
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	out_low	out_high
Description:	"output low value"	"output high value"
Data_Type:	real	real
Default_Value:	-1.0	1.0
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	duty_cycle	rise_time
Description:	"duty cycle"	"rise time"
Data_Type:	real	real
Default_Value:	0.5	1.0e-9
Limits:	[1e-6 .999999]	-

Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	fall_time
Description:	"fall time"
Data_Type:	real
Default_Value:	1.0e-9
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.1.20 Controlled One-shot

NAME_TABLE:

C_Function_Name:	cm_oneshot
Spice_Model_Name:	oneshot
Description:	"one-shot"

PORT_TABLE:

Port_Name:	clk	cntl_in
Description:	"clock input"	"input"
Direction:	in	in
Default_Type:	v	v
Allowed_Types:	[v,vd,vnam,i,id]	[v,vnam,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	yes

PORT_TABLE:

Port_Name:	clear	out
Description:	"clear signal"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v,vd,vnam,i,id]	[v,vd,i,id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	no

PARAMETER_TABLE:

Parameter_Name:	cntl_array	pw_array
Description:	"control in array"	"pulse width array"
Data_Type:	real	real
Default_Value:	0.0	1.0e-6
Limits:	-	[0 -]
Vector:	yes	yes
Vector_Bounds:	[2 -]	[2 -]
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	clk_trig	pos_edge_trig
Description:	"clock trigger value"	"pos/neg edge trigger switch"
Data_Type:	real	boolean
Default_Value:	0.5	TRUE
Limits:	-	-

```
Vector:            no                    no
Vector_Bounds:     -                     -
Null_Allowed:      no                    no


PARAMETER_TABLE:

Parameter_Name:    out_low               out_high
Description:       "output low value"    "output high value"
Data_Type:         real                  real
Default_Value:     0.0                   1.0
Limits:            -                  -
Vector:            no                    no
Vector_Bounds:     -                     -
Null_Allowed:      yes                   yes

PARAMETER_TABLE:

Parameter_Name:    rise_time
Description:       "output rise time"
Data_Type:         real
Default_Value:     1.0e-9
Limits:            -
Vector:            no
Vector_Bounds:     -
Null_Allowed:      yes

PARAMETER_TABLE:

Parameter_Name:    rise_delay                      fall_delay
Description:       "output delay from trigger"     "output delay from pw"
Data_Type:         real                            real
Default_Value:     1.0e-9                          1.0e-9
Limits:            -                               -
Vector:            no                              no
Vector_Bounds:     -                               -
Null_Allowed:      yes                             yes

PARAMETER_TABLE:

Parameter_Name:    fall_time                retrig
Description:       "output rise time"       "retrigger switch"
Data_Type:         real                     boolean
Default_Value:     1.0e-9                   FALSE
Limits:            -                        -
Vector:            no                       no
Vector_Bounds:     -                        -
Null_Allowed:      yes                      yes
```

3.5.1.21 Capacitance Meter

NAME_TABLE:

Spice_Model_Name:	cmeter
C_Function_Name:	cm_cmeter
Description:	"ATESSE 1 compatible capacitance meter"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v, vd]	[v, vd, i, id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	gain
Description:	"C to voltage conversion factor"
Data_Type:	real
Default_Value:	1.0
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

STATIC_VAR_TABLE:

Static_Var_Name:	c
Data_Type:	real
Description:	"capacitance connected to input node"

3.5.1.22 Inductance Meter

NAME_TABLE:

Spice_Model_Name:	lmeter
C_Function_Name:	cm_lmeter
Description:	"ATESSE 1 compatible inductance meter"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	v
Allowed_Types:	[v, vd]	[v, vd, i, id]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	gain
Description:	"L to voltage conversion factor"
Data_Type:	real
Default_Value:	1.0
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

STATIC_VAR_TABLE:

Static_Var_Name:	l
Data_Type:	real
Description:	"inductance connected to input node"

3.5.2 Hybrid Models

3.5.2.1 Digital-to-Analog Node Bridge

```
NAME_TABLE:


C_Function_Name:       cm_dac_bridge
Spice_Model_Name:      dac_bridge
Description:           "digital-to-analog converter node bridge"


PORT_TABLE:


Port_Name:           in                out
Description:         "input"           "output"
Direction:           in                out
Default_Type:        d                 v
Allowed_Types:       [d]               [v,vd,i,id]
Vector:              yes               yes
Vector_Bounds:       -                 -
Null_Allowed:        no                no



PARAMETER_TABLE:


Parameter_Name:     out_low
Description:        "analog output for 'ZERO' digital input"
Data_Type:          real
Default_Value:      0.0
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       yes


PARAMETER_TABLE:


Parameter_Name:     out_high
Description:        "analog output for 'ONE' digital input"
Data_Type:          real
Default_Value:      1.0
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       yes
```

PARAMETER_TABLE:

Parameter_Name:	out_undef
Description:	"analog output for 'UNDEFINED' digital input"
Data_Type:	real
Default_Value:	0.5
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"capacitive input load (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	t_rise	t_fall
Description:	"rise time 0 -> 1"	"fall time 1 -> 0"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.2.2 Analog-to-Digital Node Bridge

NAME_TABLE:

Spice_Model_Name:	adc_bridge
C_Function_Name:	cm_adc_bridge
Description:	"analog-to-digital converter node bridge"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	v	d
Allowed_Types:	[v,vd,i,id,vnam]	[d]
Vector:	yes	yes
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	in_low
Description:	"maximum 0-valued analog input"
Data_Type:	real
Default_Value:	0.1
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	in_high
Description:	"minimum 1-valued analog input"
Data_Type:	real
Default_Value:	0.9
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real

Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.2.3 Controlled Digital Oscillator

NAME_TABLE:

Spice_Model_Name:	d_osc
C_Function_Name:	cm_d_osc
Description:	"controlled digital oscillator"

PORT_TABLE:

Port_Name:	cntl_in	out
Description:	"control input"	"output"
Direction:	in	out
Default_Type:	v	d
Allowed_Types:	[v,vd,i,id]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	cntl_array	freq_array
Description:	"control array"	"frequency array"
Data_Type:	real	real
Default_Value:	0.0	1.0e6
Limits:	-	[0 -]
Vector:	yes	yes
Vector_Bounds:	[2 -]	[2 -]
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	duty_cycle	init_phase
Description:	"output duty cycle"	"initial phase of output"
Data_Type:	real	real
Default_Value:	0.5	0
Limits:	[1e-6 0.999999]	[-180.0 +360.0]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real

Default_Value:	1e-9	1e-9
Limits:	[0 -]	[0 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.3 Digital Models

3.5.3.1 Buffer

NAME_TABLE:

Spice_Model_Name:	d_buffer
C_Function_Name:	cm_d_buffer
Description:	"digital one-bit-wide buffer"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.2 Inverter

NAME_TABLE:

C_Function_Name:	cm_d_inverter
Spice_Model_Name:	d_inverter
Description:	"digital one-bit-wide inverter"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.3 And

NAME_TABLE:

C_Function_Name:	cm_d_and
Spice_Model_Name:	d_and
Description:	"digital n-input and gate"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	no
Vector_Bounds:	[2 -]	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.4 Nand

NAME_TABLE:

C_Function_Name:	cm_d_nand
Spice_Model_Name:	d_nand
Description:	"digital n-input nand gate"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	no
Vector_Bounds:	[2 -]	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.5 Or

NAME_TABLE:

C_Function_Name:	cm_d_or
Spice_Model_Name:	d_or
Description:	"digital n-input or gate"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	no
Vector_Bounds:	[2 -]	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.6 Nor

NAME_TABLE:

C_Function_Name:	cm_d_nor
Spice_Model_Name:	d_nor
Description:	"digital n-input nor gate"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	no
Vector_Bounds:	[2 -]	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (pF)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.7 Xor

NAME_TABLE:

C_Function_Name:	cm_d_xor
Spice_Model_Name:	d_xor
Description:	"digital n-input xor gate"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	no
Vector_Bounds:	[2 -]	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.8 Xnor

NAME_TABLE:

C_Function_Name:	cm_d_xnor
Spice_Model_Name:	d_xnor
Description:	"digital n-input xnor gate"

PORT_TABLE:

Port_Name:	in	out
Description:	"input"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	no
Vector_Bounds:	[2 -]	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (pF)"
Data_Type:	real
Default_Value:	1.0
Limits:	[0.0 -]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.9 Tristate

NAME_TABLE:

Spice_Model_Name:	**d_tristate**
C_Function_Name:	**cm_d_tristate**
Description:	**"digital one-bit-wide tristate buffer"**

PORT_TABLE:

Port_Name:	in	enable	out
Description:	"input"	"enable"	"output"
Direction:	in	in	out
Default_Type:	d	d	d
Allowed_Types:	[d]	[d]	[d]
Vector:	no	no	no
Vector_Bounds:	-	-	-
Null_Allowed:	no	no	no

PARAMETER_TABLE:

Parameter_Name:	delay
Description:	"delay"
Data_Type:	real
Default_Value:	1.0e-9
Limits:	[1e-12 -]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	input_load
Description:	"input load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	enable_load
Description:	"enable load value (F)"
Data_Type:	real
Default_Value:	1.0e-12

```
Limits:          -
Vector:          no
Vector_Bounds:   -
Null_Allowed:    yes
```

3.5.3.10 Pullup

NAME_TABLE:

Spice_Model_Name:	d_pullup
C_Function_Name:	cm_d_pullup
Description:	"digital pullup resistor"

PORT_TABLE:

Port_Name:	out
Description:	"output"
Direction:	out
Default_Type:	d
Allowed_Types:	[d]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	no

PARAMETER_TABLE:

Parameter_Name:	load
Description:	"load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.11 Pulldown

NAME_TABLE:

Spice_Model_Name:	d_pulldown
C_Function_Name:	cm_d_pulldown
Description:	"digital pulldown resistor"

PORT_TABLE:

Port_Name:	out
Description:	"output"
Direction:	out
Default_Type:	d
Allowed_Types:	[d]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	no

PARAMETER_TABLE:

Parameter_Name:	load
Description:	"load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.12 D Flip Flop

NAME_TABLE:

C_Function_Name:	cm_d_dff
Spice_Model_Name:	d_dff
Description:	"digital d-type flip flop"

PORT_TABLE:

Port_Name:	data	clk
Description:	"input data"	"clock"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PORT_TABLE:

Port_Name:	set	reset
Description:	"asynch. set"	"asynch. reset"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PORT_TABLE:

Port_Name:	out	Nout
Description:	"data output"	"inverted data output"
Direction:	out	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	clk_delay	set_delay
Description:	"delay from clk"	"delay from set"

```
Data_Type:          real                real
Default_Value:      1.0e-9              1.0e-9
Limits:             [1e-12 -]           [1e-12 -]
Vector:             no                  no
Vector_Bounds:      -                   -
Null_Allowed:       yes                 yes


PARAMETER_TABLE:

Parameter_Name:     reset_delay         ic
Description:        "delay from reset"  "output initial state"
Data_Type:          real                int
Default_Value:      1.0e-9              0
Limits:             [1e-12 -]           [0 2]
Vector:             no                  no
Vector_Bounds:      -                   -
Null_Allowed:       yes                 yes


PARAMETER_TABLE:

Parameter_Name:     rise_delay                  fall_delay
Description:        "rise delay"                "fall delay"
Data_Type:          real                        real
Default_Value:      1.0e-9                      1.0e-9
Limits:             [1e-12 -]                   [1e-12 -]
Vector:             no                          no
Vector_Bounds:      -                           -
Null_Allowed:       yes                         yes


PARAMETER_TABLE:

Parameter_Name:     data_load               clk_load
Description:        "data load value (F)"   "clk load value (F)"
Data_Type:          real                    real
Default_Value:      1.0e-12                 1.0e-12
Limits:             -                       -
Vector:             no                      no
Vector_Bounds:      -                       -
Null_Allowed:       yes                     yes


PARAMETER_TABLE:

Parameter_Name:     set_load                reset_load
Description:        "set load value (F)"    "reset load value (F)"
Data_Type:          real                    real
Default_Value:      1.0e-12                 1.0e-12
Limits:             -                       -
Vector:             no                      no
```

Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.3.13 JK Flip Flop

NAME_TABLE:

C_Function_Name:	cm_d_jkff
Spice_Model_Name:	d_jkff
Description:	"digital jk-type flip flop"

PORT_TABLE:

Port_Name:	j	k
Description:	"j input"	"k input"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PORT_TABLE:

Port_Name:	clk
Description:	"clock"
Direction:	in
Default_Type:	d
Allowed_Types:	[d]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	no

PORT_TABLE:

Port_Name:	set	reset
Description:	"asynch. set"	"asynch. reset"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PORT_TABLE:

Port_Name:	out	Nout
Description:	"data output"	"inverted data output"

```
Direction:          out                out
Default_Type:       d                  d
Allowed_Types:      [d]                [d]
Vector:             no                 no
Vector_Bounds:      -                  -
Null_Allowed:       yes                yes


PARAMETER_TABLE:

Parameter_Name:     clk_delay          set_delay
Description:        "delay from clk"   "delay from set"
Data_Type:          real               real
Default_Value:      1.0e-9             1.0e-9
Limits:             [1e-12 -]          [1e-12 -]
Vector:             no                 no
Vector_Bounds:      -                  -
Null_Allowed:       yes                yes


PARAMETER_TABLE:

Parameter_Name:     reset_delay        ic
Description:        "delay from reset" "output initial state"
Data_Type:          real               int
Default_Value:      1.0e-9             0
Limits:             [1e-12 -]          [0 2]
Vector:             no                 no
Vector_Bounds:      -                  -
Null_Allowed:       yes                yes


PARAMETER_TABLE:

Parameter_Name:     rise_delay                  fall_delay
Description:        "rise delay"                "fall delay"
Data_Type:          real                        real
Default_Value:      1.0e-9                      1.0e-9
Limits:             [1e-12 -]                   [1e-12 -]
Vector:             no                          no
Vector_Bounds:      -                           -
Null_Allowed:       yes                         yes


PARAMETER_TABLE:

Parameter_Name:     jk_load                 clk_load
Description:        "j,k load values (F)"   "clk load value (F)"
Data_Type:          real                    real
Default_Value:      1.0e-12                 1.0e-12
Limits:             -                       -
Vector:             no                      no
```

Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	set_load	reset_load
Description:	"set load value (F)"	"reset load value (F)"
Data_Type:	real	real
Default_Value:	1.0e-12	1.0e-12
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.3.14 Toggle Flip Flop

```
NAME_TABLE:


C_Function_Name:      cm_d_tff
Spice_Model_Name:     d_tff
Description:          "digital toggle flip flop"


PORT_TABLE:

Port_Name:           t                  clk
Description:         "toggle input"     "clock"
Direction:           in                 in
Default_Type:        d                  d
Allowed_Types:       [d]                [d]
Vector:              no                 no
Vector_Bounds:       -                  -
Null_Allowed:        no                 no


PORT_TABLE:

Port_Name:           set                reset
Description:         "asynch. set"      "asynch. reset"
Direction:           in                 in
Default_Type:        d                  d
Allowed_Types:       [d]                [d]
Vector:              no                 no
Vector_Bounds:       -                  -
Null_Allowed:        yes                yes


PORT_TABLE:

Port_Name:           out                Nout
Description:         "data output"      "inverted data output"
Direction:           out                out
Default_Type:        d                  d
Allowed_Types:       [d]                [d]
Vector:              no                 no
Vector_Bounds:       -                  -
Null_Allowed:        yes                yes


PARAMETER_TABLE:

Parameter_Name:     clk_delay           set_delay
Description:        "delay from clk"    "delay from set"
```

Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	reset_delay	ic
Description:	"delay from reset"	"output initial state"
Data_Type:	real	int
Default_Value:	1.0e-9	0
Limits:	[1e-12 -]	[0 2]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	t_load	clk_load
Description:	"toggle load value (F)"	"clk load value (F)"
Data_Type:	real	real
Default_Value:	1.0e-12	1.0e-12
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	set_load	reset_load
Description:	"set load value (F)"	"reset load value (F)"
Data_Type:	real	real
Default_Value:	1.0e-12	1.0e-12
Limits:	-	-
Vector:	no	no

Vector_Bounds:	**-**	**-**
Null_Allowed:	**yes**	**yes**

3.5.3.15 Reset-Set Flip Flop

NAME_TABLE:

C_Function_Name:	cm_d_srff
Spice_Model_Name:	d_srff
Description:	"digital set-reset flip flop"

PORT_TABLE:

Port_Name:	s	r
Description:	"s input"	"r input"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PORT_TABLE:

Port_Name:	clk
Description:	"clock"
Direction:	in
Default_Type:	d
Allowed_Types:	[d]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	no

PORT_TABLE:

Port_Name:	set	reset
Description:	"asynch. set"	"asynch. reset"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PORT_TABLE:

Port_Name:	out	Nout
Description:	"data output"	"inverted data output"

Direction:	out	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	clk_delay	set_delay
Description:	"delay from clk"	"delay from set"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	reset_delay	ic
Description:	"delay from reset"	"output initial state"
Data_Type:	real	int
Default_Value:	1.0e-9	0
Limits:	[1e-12 -]	[0 2]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	sr_load	clk_load
Description:	"s,r load values (F)"	"clk load value (F)"
Data_Type:	real	real
Default_Value:	1.0e-12	1.0e-12
Limits:	-	-
Vector:	no	no

Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	set_load	reset_load
Description:	"set load value (F)"	"reset load value (F)"
Data_Type:	real	real
Default_Value:	1.0e-12	1.0e-12
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.3.16 D Latch

NAME_TABLE:

C_Function_Name:	cm_d_dlatch
Spice_Model_Name:	d_dlatch
Description:	"digital d-type latch"

PORT_TABLE:

Port_Name:	data	enable
Description:	"input data"	"enable"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PORT_TABLE:

Port_Name:	set	reset
Description:	"asynch. set"	"asynch. reset"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PORT_TABLE:

Port_Name:	out	Nout
Description:	"data output"	"inverted data output"
Direction:	out	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	data_delay
Description:	"delay from data"

```
Data_Type:          real
Default_Value:      1.0e-9
Limits:             [1e-12 -]
Vector:             no
Vector_Bounds:      -
Null_Allowed:       yes


PARAMETER_TABLE:

Parameter_Name:     enable_delay        set_delay
Description:        "delay from clk"    "delay from set"
Data_Type:          real                real
Default_Value:      1.0e-9              1.0e-9
Limits:             [1e-12 -]           [1e-12 -]
Vector:             no                  no
Vector_Bounds:      -                   -
Null_Allowed:       yes                 yes


PARAMETER_TABLE:

Parameter_Name:     reset_delay         ic
Description:        "delay from reset"  "output initial state"
Data_Type:          real                int
Default_Value:      1.0e-9              0
Limits:             [1e-12 -]           [0 2]
Vector:             no                  no
Vector_Bounds:      -                   -
Null_Allowed:       yes                 yes


PARAMETER_TABLE:

Parameter_Name:     rise_delay                    fall_delay
Description:        "rise delay"                  "fall delay"
Data_Type:          real                          real
Default_Value:      1.0e-9                        1.0e-9
Limits:             [1e-12 -]                     [1e-12 -]
Vector:             no                            no
Vector_Bounds:      -                             -
Null_Allowed:       yes                           yes


PARAMETER_TABLE:

Parameter_Name:     data_load               enable_load
Description:        "data load value (F)"   "clk load value (F)"
Data_Type:          real                    real
Default_Value:      1.0e-12                 1.0e-12
Limits:             -                       -
Vector:             no                      no
```

Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	set_load	reset_load
Description:	"set load value (F)"	"reset load value (F)"
Data_Type:	real	real
Default_Value:	1.0e-12	1.0e-12
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.3.17 Set-Reset Latch

```
NAME_TABLE:


C_Function_Name:       cm_d_srlatch
Spice_Model_Name:      d_srlatch
Description:           "digital sr-type latch"


PORT_TABLE:

Port_Name:           s                  r
Description:         "s input"          "r input"
Direction:           in                 in
Default_Type:        d                  d
Allowed_Types:       [d]                [d]
Vector:              no                 no
Vector_Bounds:       -                  -
Null_Allowed:        no                 no


PORT_TABLE:

Port_Name:           enable
Description:         "enable"
Direction:           in
Default_Type:        d
Allowed_Types:       [d]
Vector:              no
Vector_Bounds:       -
Null_Allowed:        no


PORT_TABLE:

Port_Name:           set                reset
Description:         "asynch. set"      "asynch. reset"
Direction:           in                 in
Default_Type:        d                  d
Allowed_Types:       [d]                [d]
Vector:              no                 no
Vector_Bounds:       -                  -
Null_Allowed:        yes                yes


PORT_TABLE:

Port_Name:           out                Nout
Description:         "data output"      "inverted data output"
```

```
Direction:          out                 out
Default_Type:       d                   d
Allowed_Types:      [d]                 [d]
Vector:             no                  no
Vector_Bounds:      -                   -
Null_Allowed:       yes                 yes


PARAMETER_TABLE:

Parameter_Name:     sr_delay
Description:        "delay from s or r input change"
Data_Type:          real
Default_Value:      1.0e-9
Limits:             [1e-12 -]
Vector:             no
Vector_Bounds:      -
Null_Allowed:       yes


PARAMETER_TABLE:

Parameter_Name:     enable_delay        set_delay
Description:        "delay from clk"    "delay from set"
Data_Type:          real                real
Default_Value:      1.0e-9              1.0e-9
Limits:             [1e-12 -]           [1e-12 -]
Vector:             no                  no
Vector_Bounds:      -                   -
Null_Allowed:       yes                 yes


PARAMETER_TABLE:

Parameter_Name:     reset_delay         ic
Description:        "delay from reset"  "output initial state"
Data_Type:          real                int
Default_Value:      1.0e-9              0
Limits:             [1e-12 -]           [0 2]
Vector:             no                  no
Vector_Bounds:      -                   -
Null_Allowed:       yes                 yes


PARAMETER_TABLE:

Parameter_Name:     rise_delay                    fall_delay
Description:        "rise delay"                  "fall delay"
Data_Type:          real                          real
Default_Value:      1.0e-9                        1.0e-9
Limits:             [1e-12 -]                     [1e-12 -]
Vector:             no                            no
```

Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	sr_load	enable_load
Description:	"s & r load values (F)"	"clk load value (F)"
Data_Type:	real	real
Default_Value:	1.0e-12	1.0e-12
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	set_load	reset_load
Description:	"set load value (F)"	"reset load value (F)"
Data_Type:	real	real
Default_Value:	1.0e-12	1.0e-12
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.3.18 State Machine

NAME_TABLE:

C_Function_Name:	cm_d_state
Spice_Model_Name:	d_state
Description:	"digital state machine"

PORT_TABLE:

Port_Name:	in	clk
Description:	"input"	"clock"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	no
Vector_Bounds:	-	-
Null_Allowed:	yes	no

PORT_TABLE:

Port_Name:	reset	out
Description:	"reset"	"output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	yes
Vector_Bounds:	-	[1 -]
Null_Allowed:	yes	no

PARAMETER_TABLE:

Parameter_Name:	clk_delay	reset_delay
Description:	"delay from CLK"	"delay from reset"
Data_Type:	real	real
Default_Value:	1.0e-9	1.0e-9
Limits:	-	-
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

```
PARAMETER_TABLE:


Parameter_Name:     state_file
Description:        "state transition specification file name"
Data_Type:          string
Default_Value:      "state.txt"
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       no


PARAMETER_TABLE:


Parameter_Name:     reset_state
Description:        "default state on RESET & at DC"
Data_Type:          int
Default_Value:      0
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       no


PARAMETER_TABLE:


Parameter_Name:     input_load
Description:        "input loading capacitance (F)"
Data_Type:          real
Default_Value:      1.0e-12
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       no


PARAMETER_TABLE:


Parameter_Name:     clk_load
Description:        "clock loading capacitance (F)"
Data_Type:          real
Default_Value:      1.0e-12
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       no
```

PARAMETER_TABLE:

Parameter_Name:	reset_load
Description:	"reset loading capacitance (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	no

3.5.3.19 Frequency Divider

NAME_TABLE:

C_Function_Name:	cm_d_fdiv
Spice_Model_Name:	d_fdiv
Description:	"digital frequency divider"

PORT_TABLE:

Port_Name:	freq_in	freq_out
Description:	"frequency input"	"frequency output"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	no	no

PARAMETER_TABLE:

Parameter_Name:	div_factor	high_cycles
Description:	"divide factor"	"number of high clock cycles"
Data_Type:	int	int
Default_Value:	2	1
Limits:	[1 -]	[1 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	i_count
Description:	"output initial count value"
Data_Type:	int
Default_Value:	0
Limits:	[0 -]
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

PARAMETER_TABLE:

Parameter_Name:	rise_delay	fall_delay
Description:	"rise delay"	"fall delay"
Data_Type:	real	real

Default_Value:	1.0e-9	1.0e-9
Limits:	[1e-12 -]	[1e-12 -]
Vector:	no	no
Vector_Bounds:	-	-
Null_Allowed:	yes	yes

PARAMETER_TABLE:

Parameter_Name:	freq_in_load
Description:	"freq_in load value (F)"
Data_Type:	real
Default_Value:	1.0e-12
Limits:	-
Vector:	no
Vector_Bounds:	-
Null_Allowed:	yes

3.5.3.20 RAM

NAME_TABLE:

C_Function_Name:	cm_d_ram
Spice_Model_Name:	d_ram
Description:	"digital random-access memory"

PORT_TABLE:

Port_Name:	data_in	data_out
Description:	"data input line(s)"	"data output line(s)"
Direction:	in	out
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	yes
Vector_Bounds:	[1 -]	[1 -]
Null_Allowed:	no	no

PORT_TABLE:

Port_Name:	address	write_en
Description:	"address input line(s)"	"write enable"
Direction:	in	in
Default_Type:	d	d
Allowed_Types:	[d]	[d]
Vector:	yes	no
Vector_Bounds:	[1 -]	-
Null_Allowed:	no	no

PORT_TABLE:

Port_Name:	select
Description:	"chip select line(s)"
Direction:	in
Default_Type:	d
Allowed_Types:	[d]
Vector:	yes
Vector_Bounds:	[1 16]
Null_Allowed:	no

PARAMETER_TABLE:

Parameter_Name:	select_value
Description:	"decimal active value for select line comparison"

```
Data_Type:          int
Default_Value:      1
Limits:             [0 32767]
Vector:             no
Vector_Bounds:      -
Null_Allowed:       yes


PARAMETER_TABLE:

Parameter_Name:     ic
Description:        "initial bit state @ DC"
Data_Type:          int
Default_Value:      2
Limits:             [0 2]
Vector:             no
Vector_Bounds:      -
Null_Allowed:       yes


PARAMETER_TABLE:

Parameter_Name:     read_delay
Description:        "read delay from address/select/write_en active"
Data_Type:          real
Default_Value:      100.0e-9
Limits:             [1e-12 -]
Vector:             no
Vector_Bounds:      -
Null_Allowed:       yes


PARAMETER_TABLE:

Parameter_Name:     data_load                   address_load
Description:        "data_in load value (F)"    "address line load value (F)"
Data_Type:          real                        real
Default_Value:      1.0e-12                     1.0e-12
Limits:             -                           -
Vector:             no                          no
Vector_Bounds:      -                           -
Null_Allowed:       yes                         yes


PARAMETER_TABLE:

Parameter_Name:     select_load               enable_load
Description:        "select load value (F)" "enable line load value (F)"
Data_Type:          real                      real
Default_Value:      1.0e-12                   1.0e-12
Limits:             -                         -
Vector:             no                        no
```

Vector_Bounds:	-	-
Null_Allowed:	yes	yes

3.5.3.21 Digital Source

```
NAME_TABLE:


C_Function_Name:       cm_d_source
Spice_Model_Name:      d_source
Description:           "digital signal source"


PORT_TABLE:


Port_Name:           out
Description:         "output"
Direction:           out
Default_Type:        d
Allowed_Types:       [d]
Vector:              yes
Vector_Bounds:       -
Null_Allowed:        no



PARAMETER_TABLE:


Parameter_Name:     input_file
Description:        "digital input vector filename"
Data_Type:          string
Default_Value:      "source.txt"
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       no


PARAMETER_TABLE:


Parameter_Name:     input_load
Description:        "input loading capacitance (F)"
Data_Type:          real
Default_Value:      1.0e-12
Limits:             -
Vector:             no
Vector_Bounds:      -
Null_Allowed:       no
```

3.6 User-Defined Node Library

The following two predefined user-defined node types are included in the User-Defined Node Library.

```
real
int
```

Unlike the code models described above, there is no Interface Specification file associated with user-defined nodes. Instead, a structure of type 'Evt_Udn_Info_t' is placed at the bottom of the User-Defined Node Definition file by the utility 'mkudndir' when a user-defined node directory is created. This structure contains the type name for the node, a description string, and pointers to each of the functions that define the node.

The following subsections give the definition of the 'Evt_Udn_Info_t' structure associated with the 'real' and 'int' node types. The functions that define each of these user-defined nodes are documented in the Software Design Document for the XSPICE Code Model Subsystem of the Automatic Test Equipment Software Support Environment (ATESSE).

3.6.1 Real

```
Evt_Udn_Info_t udn_real_info =

    "real",
    "real valued data",

    udn_real_create,
    udn_real_dismantle,
    udn_real_initialize,
    udn_real_invert,
    udn_real_copy,
    udn_real_resolve,
    udn_real_compare,
    udn_real_plot_val,
    udn_real_print_val,
    udn_real_ipc_val

;
```

3.6.2 Int

```
Evt_Udn_Info_t udn_int_info =

    "int",
    "integer valued data",

    udn_int_create,
    udn_int_dismantle,
    udn_int_initialize,
    udn_int_invert,
    udn_int_copy,
    udn_int_resolve,
    udn_int_compare,
    udn_int_plot_val,
    udn_int_print_val,
    udn_int_ipc_val

 ;
```

4 Notes

4.1 Glossary

ATESSE	Automatic Test Equipment Software Support Environment. An integrated set of software tools designed to aid in development of programs for testing mixed-mode (analog/digital) printed circuit cards.
C	A programming language developed in the early 70's at Bell Labs. It is the standard programming language used for system development on Unix machines.
Code Model Library	The set of code models supplied with the XSPICE simulator.
Code Model Preprocessor	A code model development tool that assists in adding new code models to the XSPICE simulator. It is automatically run by 'make' to convert user-written files into C language source files that are compiled and linked with the simulator.
Code Model Toolkit	A set of tools that assist in adding new code models to the XSPICE simulator.
Interprocess Communication	Communication between two separate programs running simultaneously on one or more computers.
Make	A UNIX software development tool that automates the compilation and linking of a program.
Makefile	A file that instructs the UNIX make utility how to compile and link a program.

malloc	A standard C library function for dynamically allocating memory in a program.
Model Directory Generator	A code model development tool that assists in adding new code models to the XSPICE simulator. It creates a directory in which a code model will be created.
Simulator Directory Generator	A code model development tool that assists in adding new code models to the XSPICE simulator. It creates a directory in which a copy of the simulator will be built.
SPICE	An analog simulation program developed at the University of California at Berkeley.
User-Defined Node Directory Generator	A development tool that assists in adding new user-defined nodes to the XSPICE simulator. It creates a directory in which a new node type will be created.
User-Defined Node Library	The set of user-defined node types supplied with the XSPICE simulator.

4.2 Acronyms

ANSI	American National Standards Institute
ATE	Automatic Test Equipment
ATESSE	Automatic Test Equipment Software Support Environment
CDRL	Contract Deliverable Requirement List
CSCI	Computer Software Configuration Item
IFS	Interface Specification File
IPC	Interprocess Communication
PUI	Project Unique Identifier
PWL	Piecewise-Linear
SPICE	Simulation Program with Integrated Circuit Emphasis
UDN	User-Defined Node
XSPICE	Extended SPICE

4.3 Project Unique Identifiers

This section lists the Project Unique Identifiers (PUI) referred to in this document.

CM-SI-IN	Interface between the Code Model (CM) Subsystem and the ATESSE Simulator Interface (SI).
CM-SI-OUT	Interface between the Code Model (CM) Subsystem and the ATESSE Simulator Interface (SI).
SIM-CM-IN	Interface between the Simulator (SIM) and the Code Model Subsystem (CM). This interface is defined in the Interface Design Document for the Simulator of the ATESSE.