cadence

PSpice Device Modeling Interface API Reference

Product Version 23.1
September 2023

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Contents

1 Introduction 1

2 Data Structure Index 3

2.1 Data Structures . . . 3

3 File Index 5

3.1 File List . . . 5

4 Data Structure Documentation 7

4.1 pspBit Class Reference . . . 7

4.1.1 Detailed Description . . . 7

4.1.2 Member Enumeration Documentation . . . 7

4.1.2.1 pspBitLevels . . . 8

4.1.3 Constructor & Destructor Documentation . . . 8

4.1.3.1 pspBit . . . 8

4.1.3.2 pspBit . . . 8

4.1.3.3 pspBit . . . 8

4.1.4 Member Function Documentation . . . 8

4.1.4.1 operator char . . . 8

4.1.4.2 operator int . . . 8

4.1.4.3 operator= . . . 8

4.1.4.4 operator= . . . 8

4.1.5 Friends And Related Function Documentation . . . 8

4.1.5.1 operator& . . . 8

4.1.5.2 operator== . . . 9

4.1.5.3 operator^ . . . 9

4.1.5.4 operator"| . . . 9

4.1.5.5 operator~ . . . 9

4.2 PSpiceAnyScalar Class Reference . . . 9

4.2.1 Detailed Description . . . 9

4.2.2 Field Documentation . . . 9

4.2.2.1 exp . . . 9

4.2.2.2 Real . . . 9
4.2.2.3 String . . . 10
4.3 PSpiceAnyValue Class Reference . . . 10
4.3.1 Detailed Description . . . 10
4.3.2 Field Documentation . . . 10
4.3.2.1 Scalar . . . 10
4.3.2.2 Type . . . 10
4.4 PSpiceCMIParam Class Reference . . . 10
4.4.1 Detailed Description . . . 11
4.4.2 Constructor & Destructor Documentation . . . 11
4.4.2.1 PSpiceCMIParam . . . 11
4.4.2.2 ∼PSpiceCMIParam . . . 11
4.4.3 Field Documentation . . . 11
4.4.3.1 Desc . . . 11
4.4.3.2 Value . . . 11
4.5 PSpiceConstraint Class Reference . . . 11
4.5.1 Detailed Description . . . 12
4.5.2 Field Documentation . . . 12
4.5.2.1 mFreq . . . 12
4.5.2.2 mSetupHold . . . 12
4.5.2.3 mWidth . . . 12
4.6 PSpiceDelay Class Reference . . . 12
4.6.1 Detailed Description . . . 12
4.6.2 Field Documentation . . . 12
4.6.2.1 mMaxDelay . . . 12
4.6.2.2 mMinDelay . . . 12
4.6.2.3 mTypDelay . . . 13
4.7 PSpiceDeviceInst Class Reference . . . 13
4.7.1 Detailed Description . . . 13
4.7.2 Field Documentation . . . 13
4.7.2.1 mData . . . 13
4.7.2.2 mInstID . . . 13
4.7.2.3 mModelData . . . 13
4.7.2.4 mSignals . . . 13
4.7.2.5 mState . . . 13
4.7.2.6 mStencil . . . 14
4.8 PSpiceDeviceMiscInfo Class Reference . . . 14
4.8.1 Detailed Description . . . 14
4.8.2 Field Documentation . . . 14
4.8.2.1 CKTag . . . 14

4.8.2.2 CKTomega . . . 14
4.8.2.3 CurrentAnalysisNumber . . . 14
4.8.2.4 GMin . . . 14
4.8.2.5 MeasurementTemperature . . . 14
4.8.2.6 RelTol . . . 15
4.8.2.7 SkipBP . . . 15
4.8.2.8 Temperature . . . 15
4.8.2.9 Vt . . . 15
4.9 PSpiceDeviceModel Class Reference . . . 15
4.9.1 Detailed Description . . . 15
4.9.2 Field Documentation . . . 15
4.9.2.1 mData . . . 15
4.9.2.2 mModelID . . . 15
4.10 PSpiceFreqConstraint Class Reference . . . 16
4.10.1 Detailed Description . . . 16
4.10.2 Constructor & Destructor Documentation . . . 16
4.10.2.1 PSpiceFreqConstraint . . . 16
4.10.3 Field Documentation . . . 16
4.10.3.1 errorflags . . . 16
4.10.3.2 max_freq . . . 16
4.10.3.3 mFreqSpecified . . . 16
4.10.3.4 min_freq . . . 17
4.10.3.5 mInputNode . . . 17
4.11 PSpiceInputSpec Class Reference . . . 17
4.11.1 Detailed Description . . . 17
4.11.2 Field Documentation . . . 17
4.11.2.1 inR . . . 17
4.11.2.2 load . . . 17
4.11.2.3 Tstore . . . 18
4.12 PSpiceNetsList Class Reference . . . 18
4.12.1 Detailed Description . . . 18
4.12.2 Constructor & Destructor Documentation . . . 18
4.12.2.1 PSpiceNetsList . . . 18
4.12.3 Field Documentation . . . 18
4.12.3.1 mNetName . . . 18
4.12.3.2 mNext . . . 18
4.13 PSpiceOutputSpec Class Reference . . . 18
4.13.1 Detailed Description . . . 19
4.13.2 Constructor & Destructor Documentation . . . 19
4.13.2.1 PSpiceOutputSpec . . . 19

4.13.3 Field Documentation . . . 19
4.13.3.1 h_drive . . . 19
4.13.3.2 l_drive . . . 19
4.13.3.3 load . . . 19
4.13.3.4 pwrt . . . 20
4.13.3.5 tswhl . . . 20
4.13.3.6 tswlh . . . 20
4.13.3.7 z_drive . . . 20
4.14 PSpiceParamDesc Class Reference . . . 20
4.14.1 Detailed Description . . . 20
4.14.2 Field Documentation . . . 20
4.14.2.1 mVersion . . . 20
4.14.2.2 Name . . . 21
4.15 PSpicePort Class Reference . . . 21
4.15.1 Detailed Description . . . 21
4.15.2 Field Documentation . . . 21
4.15.2.1 mName . . . 21
4.15.2.2 mNumber . . . 21
4.15.2.3 mType . . . 21
4.16 PSpiceSetupHoldConstraint Class Reference . . . 21
4.16.1 Detailed Description . . . 22
4.16.2 Constructor & Destructor Documentation . . . 22
4.16.2.1 PSpiceSetupHoldConstraint . . . 22
4.16.2.2 ∼PSpiceSetupHoldConstraint . . . 22
4.16.3 Field Documentation . . . 22
4.16.3.1 clk_assertion . . . 22
4.16.3.2 holdtime_hi . . . 22
4.16.3.3 holdtime_lo . . . 22
4.16.3.4 mClockName . . . 22
4.16.3.5 mCountData . . . 23
4.16.3.6 mNetsList . . . 23
4.16.3.7 mSetupHoldSpecified . . . 23
4.16.3.8 releasetime_hl . . . 23
4.16.3.9 releasetime_lh . . . 23
4.16.3.10 setuptime_hi . . . 23
4.16.3.11 setuptime_lo . . . 23
4.17 PSpiceSignalNodeList Class Reference . . . 23
4.17.1 Detailed Description . . . 23
4.17.2 Field Documentation . . . 23
4.17.2.1 mNodeCount . . . 23

4.17.2.2 mNodeNames . . . 24
4.18 PSpiceState Class Reference . . . 24
4.18.1 Detailed Description . . . 24
4.18.2 Member Function Documentation . . . 24
4.18.2.1 getLevel . . . 24
4.18.2.2 isZ . . . 24
4.18.2.3 operator= . . . 25
4.18.3 Field Documentation . . . 25
4.18.3.1 _filler . . . 25
4.18.3.2 fields . . . 25
4.18.3.3 hazardtype . . . 25
4.18.3.4 level . . . 25
4.18.3.5 msgid . . . 25
4.18.3.6 multiple . . . 25
4.18.3.7 notposted . . . 25
4.18.3.8 persistent . . . 25
4.18.3.9 stateVal . . . 25
4.18.3.10 str0 . . . 25
4.18.3.11 str1 . . . 25
4.18.3.12 val . . . 26
4.19 PSpiceWidthConstraint Class Reference . . . 26
4.19.1 Detailed Description . . . 26
4.19.2 Constructor & Destructor Documentation . . . 26
4.19.2.1 PSpiceWidthConstraint . . . 26
4.19.3 Field Documentation . . . 26
4.19.3.1 min_high . . . 26
4.19.3.2 min_low . . . 26
4.19.3.3 mInputNode . . . 27
4.19.3.4 mWidthSpecified . . . 27

5 File Documentation **29**

5.1 code_latest/PSpiceCMIApiDefs.h File Reference . . . 29
5.1.1 Typedef Documentation . . . 33
5.1.1.1 descSetAC_Load_t . . . 33
5.1.1.2 descSetAddInternalNodes_t . . . 33
5.1.1.3 descSetBindTerminals_t . . . 34
5.1.1.4 descSetCheckModel_t . . . 34
5.1.1.5 descSetCheckPointSize_t . . . 34
5.1.1.6 descSetCheckTopology_t . . . 34
5.1.1.7 descSetDefaultInstance_t . . . 35

5.1.1.8 descSetDefaultModel_t . . . 35
5.1.1.9 descSetDefaultState_t . . . 35
5.1.1.10 descSetDeleteInstance_t . . . 35
5.1.1.11 descSetDeleteModel_t . . . 36
5.1.1.12 descSetGetIntercept_t . . . 36
5.1.1.13 descSetGetMatrixPointers_t . . . 36
5.1.1.14 descSetInstDataStructSize_t . . . 36
5.1.1.15 descSetLoadCheckpoint_t . . . 37
5.1.1.16 descSetMaxTerminalCount_t . . . 37
5.1.1.17 descSetMinTerminalCount_t . . . 37
5.1.1.18 descSetModelDataStructSize_t . . . 38
5.1.1.19 descSetNoise_t . . . 38
5.1.1.20 descSetPreload_t . . . 38
5.1.1.21 descSetReserveNodes_t . . . 38
5.1.1.22 descSetSaveCheckpoint_t . . . 39
5.1.1.23 descSetSaveTopology_t . . . 39
5.1.1.24 descSetSetDevicePinCurrent_t . . . 39
5.1.1.25 descSetSetDevicePinCurrentComplex_t . . . 39
5.1.1.26 descSetSetInstanceParams_t . . . 40
5.1.1.27 descSetSetModelParams_t . . . 40
5.1.1.28 descSetSetTopologySize_t . . . 40
5.1.1.29 descSetSignalsStructSize_t . . . 41
5.1.1.30 descSetStateStructSize_t . . . 41
5.1.1.31 descSetStencilStructSize_t . . . 41
5.1.1.32 descSetTerminalNameCount_t . . . 42
5.1.1.33 descSetTerminalNames_t . . . 42
5.1.1.34 descSetTitle_t . . . 42
5.1.1.35 descSetTmpModDevice_t . . . 42
5.1.1.36 descSetTmpModModel_t . . . 43
5.1.1.37 descSetTranLoad_t . . . 43
5.1.1.38 descSetTrunc_t . . . 43
5.1.1.39 pAC_Load_t . . . 43
5.1.1.40 pAddInternalNodes_t . . . 44
5.1.1.41 pBindTerminals_t . . . 44
5.1.1.42 pCheckTopology_t . . . 44
5.1.1.43 pDefaultInstance_t . . . 45
5.1.1.44 pDefaultModel_t . . . 45
5.1.1.45 pDefaultModelParams_t . . . 45
5.1.1.46 pDefaultSignals_t . . . 45
5.1.1.47 pDefaultState_t . . . 46

5.1.1.48 pDefaultStencil_t . . . 46
5.1.1.49 pDeleteInstance_t . . . 46
5.1.1.50 pDeleteModel_t . . . 46
5.1.1.51 pDeleteSignals_t . . . 47
5.1.1.52 pDeleteState_t . . . 47
5.1.1.53 pDeleteStencil_t . . . 47
5.1.1.54 pGetBreakPoint_t . . . 47
5.1.1.55 pGetIntercept_t . . . 48
5.1.1.56 pGetLastVoltage_t . . . 48
5.1.1.57 pGetMatrixPointers_t . . . 48
5.1.1.58 pGetPWLData_t . . . 48
5.1.1.59 pGetPWLDataStr_t . . . 48
5.1.1.60 pInstallFunction_t . . . 48
5.1.1.61 pIsPWLModel_t . . . 49
5.1.1.62 pLoadCheckpoint_t . . . 49
5.1.1.63 pModChk_t . . . 49
5.1.1.64 pNoise_t . . . 49
5.1.1.65 pPreload_t . . . 50
5.1.1.66 pPrintDescription_t . . . 50
5.1.1.67 pPSpiceAddInternalNode_t . . . 50
5.1.1.68 pPSpiceAddInternalNodeByName_t . . . 50
5.1.1.69 pPSpiceAdjustValueItem_t . . . 51
5.1.1.70 pPSpiceApplyValueItem_t . . . 51
5.1.1.71 pPSpiceApplyValueItemComplex_t . . . 51
5.1.1.72 pPSpiceCurrentTErr_t . . . 51
5.1.1.73 pPSpiceGetCurrentStateIndex_t . . . 52
5.1.1.74 pPSpiceGetDelta_t . . . 52
5.1.1.75 pPSpiceGetDeltaPrevious_t . . . 52
5.1.1.76 pPSpiceGetFrequency_t . . . 52
5.1.1.77 pPSpiceGetMatrixPtr_t . . . 53
5.1.1.78 pPSpiceGetRHSPtr_t . . . 53
5.1.1.79 pPSpiceGetVoltageNodes_t . . . 53
5.1.1.80 pPSpiceGetVoltageNodesI_t . . . 53
5.1.1.81 pPSpiceIntegrate_t . . . 54
5.1.1.82 pPSpiceSetPWLDataDbl_t . . . 54
5.1.1.83 pPSpiceSetPWLDataStr_t . . . 54
5.1.1.84 pPSpiceUpdateStateVector_t . . . 55
5.1.1.85 pPSpiceVoltageTolerance_t . . . 55
5.1.1.86 pPWLModelType_t . . . 55
5.1.1.87 pReserveNodes_t . . . 55

5.1.1.88 PrimitivePtr . . . 55
5.1.1.89 pSaveCheckpoint_t . . . 56
5.1.1.90 pSaveTopology_t . . . 57
5.1.1.91 pSetDevicePinCurrent_t . . . 57
5.1.1.92 pSetDevicePinCurrentComplex_t . . . 57
5.1.1.93 pSetModelParams_t . . . 58
5.1.1.94 pSetTopologySize_t . . . 58
5.1.1.95 pTmpModDevice_t . . . 58
5.1.1.96 pTmpModModel_t . . . 59
5.1.1.97 pTranLoad_t . . . 59
5.1.1.98 pTrunc_t . . . 59
5.2 code_latest/PSpiceCommonAPIDefs.h File Reference . . . 60
5.2.1 Macro Definition Documentation . . . 62
5.2.1.1 CDLL_FUNC . . . 62
5.2.1.2 PSP_CMI_EXPORT . . . 62
5.2.2 Typedef Documentation . . . 62
5.2.2.1 descSetInstallFunction1_t . . . 63
5.2.2.2 descSetInstallFunction_t . . . 64
5.2.2.3 descSetName_t . . . 64
5.2.2.4 descSetVersion_t . . . 64
5.2.2.5 pDefaultInstanceParams_t . . . 64
5.2.2.6 pFnPtr1_t . . . 65
5.2.2.7 pFnPtr_t . . . 66
5.2.2.8 pPSpiceGetCurrentAnalogTime_t . . . 66
5.2.2.9 pPSpiceGetCurrentDigitalTime_t . . . 66
5.2.2.10 pPSpiceGetDevice_t . . . 66
5.2.2.11 pPSpiceGetLicenseString_t . . . 66
5.2.2.12 pPSPICEGetOptionsParams_t . . . 67
5.2.2.13 pPSpiceGetParamValue_t . . . 67
5.2.2.14 pPSpiceGetParamValueDbl_t . . . 67
5.2.2.15 pPSpiceSetProbeTitle_t . . . 67
5.2.2.16 pPSpiceSetSimulationTemperature_t . . . 67
5.2.2.17 pPSpiceWriteToOut_t . . . 68
5.2.2.18 pSetInstanceParams_t . . . 68
5.2.3 Enumeration Type Documentation . . . 68
5.2.3.1 initFlags . . . 68
5.2.3.2 modeFlags . . . 68
5.2.3.3 PSpiceAPIs . . . 69
5.2.3.4 PSpiceValueType . . . 71
5.2.4 Function Documentation . . . 71

5.2.4.1 PSpiceInstallFunction . . . 71
5.2.4.2 PSpiceInstallFunction1 . . . 72
5.2.4.3 pspiceSetFunctionList . . . 72
5.3 code_latest/PSpiceDigApiDefs.h File Reference . . . 72
5.3.1 Macro Definition Documentation . . . 74
5.3.1.1 MAXIOLEVEL . . . 74
5.3.1.2 PSP_VALUE_NOT_DEFINED . . . 74
5.3.1.3 UNSPEC . . . 74
5.3.2 Typedef Documentation . . . 74
5.3.2.1 descSetCreateDevice_t . . . 74
5.3.2.2 descSetDeleteDevice_t . . . 75
5.3.2.3 descSetEvaluateDevice_t . . . 75
5.3.2.4 descSetGetDeviceTermCount_t . . . 75
5.3.2.5 descSetGetDeviceTerminals_t . . . 75
5.3.2.6 descSetGetDeviceTermValue_t . . . 75
5.3.2.7 descSetInitDevice_t . . . 76
5.3.2.8 descSetSetDeviceTermCount_t . . . 76
5.3.2.9 descSetSetDeviceTermValue_t . . . 76
5.3.2.10 descSetSetParameter_t . . . 76
5.3.2.11 pCreateDevice_t . . . 76
5.3.2.12 pDeleteDevice_t . . . 77
5.3.2.13 pDigPrintDescription_t . . . 77
5.3.2.14 pGetDeviceTermCount_t . . . 77
5.3.2.15 pGetDeviceTermTypes_t . . . 77
5.3.2.16 pGetDeviceTermValue_t . . . 77
5.3.2.17 pGetTicksFromTime_t . . . 77
5.3.2.18 pGetTimeFromTicks_t . . . 78
5.3.2.19 pInitDevice_t . . . 78
5.3.2.20 pPSpiceChanged_t . . . 78
5.3.2.21 pPSpiceEvaluateDevice_t . . . 78
5.3.2.22 pPSpiceGetInputSpec_t . . . 78
5.3.2.23 pPSpiceGetOutputSpec_t . . . 79
5.3.2.24 pPSpiceGetParameterValue_t . . . 80
5.3.2.25 pPSpiceGetTimingModelValue_t . . . 80
5.3.2.26 pPSpiceGetTransition_t . . . 80
5.3.2.27 pPSpiceSetConstraint_t . . . 80
5.3.2.28 pPSpiceSetDelay_t . . . 81
5.3.2.29 pPSpiceSetInputSpec_t . . . 81
5.3.2.30 pPSpiceSetOutputSpec_t . . . 81
5.3.2.31 pPSpiceSetState_t . . . 81

5.3.2.32 pSetDeviceTermCount_t . . . 82
5.3.2.33 pSetDeviceTermValue_t . . . 82
5.3.2.34 pSetParameter_t . . . 82
5.3.3 Enumeration Type Documentation . . . 82
5.3.3.1 PSPICE_PORT_TYPE . . . 82
5.3.4 Function Documentation . . . 82
5.3.4.1 operator& . . . 82
5.3.4.2 operator== . . . 82
5.3.4.3 operator^ . . . 83
5.3.4.4 operator"| . . . 83
5.3.4.5 operator~ . . . 83
5.3.4.6 PSpiceGetInputSpec . . . 83
5.3.4.7 PSpiceGetOutputSpec . . . 83
5.3.4.8 PSpiceGetParameterValue . . . 83
5.3.4.9 PSpiceGetTimingModelValue . . . 83
5.3.4.10 PSpiceSetConstraint . . . 83
5.3.4.11 PSpiceSetDelay . . . 83
5.3.4.12 PSpiceSetInputSpec . . . 83
5.3.4.13 PSpiceSetOutputSpec . . . 83
5.3.4.14 PSpiceSetState . . . 83

Chapter 1

Introduction

This file lists 3 types of functions:

1. Templates for functions exposed by PSpice Engine
These are helper functions and are used by the model dll files. The function pointers of the functions are set into the model dll files by the API pspiceSetFunctionList. The function names start with pPSpice∗_t.

2. Templates for API's exposed by the model dll files
The function pointers of the functions is set by the model dll files into PSpice Engine using the descSet∗ functions. The function names start with p∗_t.

3. Templates for descSet functions
These functions are used by the model dll files to set function pointers into the PSpice Engine. The function names start with descSet∗. The Dll files must export the function "void PSpiceInstallFunction();". Inside this function, the function pointers for each of the model install function must be set.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

pspBit . . . 7
PSpiceAnyScalar
PSpice Parameter Value subclass . . . 9
PSpiceAnyValue
PSpice Parameter Value Class . . . 10
PSpiceCMIParam
PSpice Parameter Top-level class . . . 10
PSpiceConstraint
Composite class for definition of all constraints . . . 11
PSpiceDelay . . . 12
PSpiceDeviceInst
This class defines basic PSpice Device Instance which will be used to transmit data to and from CMI Models . . . 13
PSpiceDeviceMiscInfo . . . 14
PSpiceDeviceModel
This class defines basic PSpice Device Model which will be used to transmit data to and from CMI Models . . . 15
PSpiceFreqConstraint
Class for Frequency Constraint Definition . . . 16
PSpiceInputSpec
Input Buffer Specification . . . 17
PSpiceNetsList . . . 18
PSpiceOutputSpec
Output Buffer Specification . . . 18
PSpiceParamDesc
PSpice Parameter Descriptor . . . 20
PSpicePort
PSpice Port . . . 21
PSpiceSetupHoldConstraint . . . 21
PSpiceSignalNodeList . . . 23
PSpiceState
Digital State . . . 24
PSpiceWidthConstraint
Class for Width Constraint Definition . . . 26

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

code_latest/PSpiceCMIApiDefs.h 29
code_latest/PSpiceCommonAPIDefs.h 60
code_latest/PSpiceDigApiDefs.h 72

Chapter 4

Data Structure Documentation

4.1 pspBit Class Reference

`#include <PSpiceDigApiDefs.h>`

Public Types

- enum pspBitLevels {
  LO = 0, HI = 1, UNKNOWN = 2, RISE = 3,
  FALL = 4, HIZ = 5 }

Public Member Functions

- pspBit (char pChar)
- pspBit (PSpiceState &pState)
- pspBit (bool pBool=false)
- operator char () const
- operator int () const
- pspBit & operator= (const pspBit &pBit)
- pspBit & operator= (const int pValue)

Friends

- bool operator== (const pspBit &pBit1, const pspBit &pBit2)
- pspBit operator| (const pspBit &pBit1, const pspBit &pBit2)
- pspBit operator& (const pspBit &pBit1, const pspBit &pBit2)
- pspBit operator~ (const pspBit &pBit)
- pspBit operator^ (const pspBit &pBit1, const pspBit &pBit2)

4.1.1 Detailed Description

Definition at line 517 of file PSpiceDigApiDefs.h.

4.1.2 Member Enumeration Documentation

4.1.2.1 enum pspBitLevels

Enumerator

LO

HI

UNKNOWN

RISE

FALL

HIZ

Definition at line 521 of file PSpiceDigApiDefs.h.

4.1.3 Constructor & Destructor Documentation

4.1.3.1 pspBit (char *pChar*) [inline]

Definition at line 530 of file PSpiceDigApiDefs.h.

4.1.3.2 pspBit (PSpiceState & *pState*) [inline]

Definition at line 534 of file PSpiceDigApiDefs.h.

4.1.3.3 pspBit (bool *pBool* = false) [inline]

Definition at line 538 of file PSpiceDigApiDefs.h.

4.1.4 Member Function Documentation

4.1.4.1 operator char ( ) const [inline]

Definition at line 542 of file PSpiceDigApiDefs.h.

4.1.4.2 operator int ( ) const [inline]

Definition at line 546 of file PSpiceDigApiDefs.h.

4.1.4.3 pspBit& operator= (const pspBit & *pBit*) [inline]

Definition at line 558 of file PSpiceDigApiDefs.h.

4.1.4.4 pspBit& operator= (const int *pValue*) [inline]

Definition at line 562 of file PSpiceDigApiDefs.h.

4.1.5 Friends And Related Function Documentation

4.1.5.1 pspBit operator& (const pspBit & *pBit1*, const pspBit & *pBit2*) [friend]

Definition at line 605 of file PSpiceDigApiDefs.h.

4.1.5.2 bool operator== (const pspBit & *pBit1*, const pspBit & *pBit2*) [friend]

Definition at line 596 of file PSpiceDigApiDefs.h.

4.1.5.3 pspBit operator^ (const pspBit & *pBit1*, const pspBit & *pBit2*) [friend]

Definition at line 600 of file PSpiceDigApiDefs.h.

4.1.5.4 pspBit operator| (const pspBit & *pBit1*, const pspBit & *pBit2*) [friend]

Definition at line 610 of file PSpiceDigApiDefs.h.

4.1.5.5 pspBit operator~ (const pspBit & *pBit*) [friend]

Definition at line 615 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.2 PSpiceAnyScalar Class Reference

PSpice Parameter Value subclass.

`#include <PSpiceCommonAPIDefs.h>`

Data Fields

- double Real
- std::string String
- void * exp

4.2.1 Detailed Description

PSpice Parameter Value subclass.

Definition at line 342 of file PSpiceCommonAPIDefs.h.

4.2.2 Field Documentation

4.2.2.1 void* exp

Expression value

Definition at line 346 of file PSpiceCommonAPIDefs.h.

4.2.2.2 double Real

Double value

Definition at line 344 of file PSpiceCommonAPIDefs.h.

4.2.2.3 std::string String

String value

Definition at line 345 of file PSpiceCommonAPIDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceCommonAPIDefs.h

4.3 PSpiceAnyValue Class Reference

PSpice Parameter Value Class.

`#include <PSpiceCommonAPIDefs.h>`

Data Fields

- PSpiceValueType Type
- PSpiceAnyScalar Scalar

4.3.1 Detailed Description

PSpice Parameter Value Class.

Definition at line 350 of file PSpiceCommonAPIDefs.h.

4.3.2 Field Documentation

4.3.2.1 PSpiceAnyScalar Scalar

Actual Value

Definition at line 353 of file PSpiceCommonAPIDefs.h.

4.3.2.2 PSpiceValueType Type

Value Type

Definition at line 352 of file PSpiceCommonAPIDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceCommonAPIDefs.h

4.4 PSpiceCMIParam Class Reference

PSpice Parameter Top-level class.

`#include <PSpiceCommonAPIDefs.h>`

Public Member Functions

- PSpiceCMIParam ()
- ~PSpiceCMIParam ()

Data Fields

- PSpiceParamDesc * Desc
- PSpiceAnyValue Value

4.4.1 Detailed Description

PSpice Parameter Top-level class.

Definition at line 357 of file PSpiceCommonAPIDefs.h.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 PSpiceCMIParam ( ) [inline]

Definition at line 362 of file PSpiceCommonAPIDefs.h.

4.4.2.2 ~PSpiceCMIParam ( ) [inline]

Definition at line 366 of file PSpiceCommonAPIDefs.h.

4.4.3 Field Documentation

4.4.3.1 PSpiceParamDesc* Desc

PSpice Parameter Descriptor

Definition at line 359 of file PSpiceCommonAPIDefs.h.

4.4.3.2 PSpiceAnyValue Value

PSpice Parameter Value

Definition at line 360 of file PSpiceCommonAPIDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceCommonAPIDefs.h

4.5 PSpiceConstraint Class Reference

Composite class for definition of all constraints.

`#include <PSpiceDigApiDefs.h>`

Data Fields

- PSpiceSetupHoldConstraint mSetupHold
- PSpiceWidthConstraint mWidth
- PSpiceFreqConstraint mFreq

4.5.1 Detailed Description

Composite class for definition of all constraints.

Definition at line 382 of file PSpiceDigApiDefs.h.

4.5.2 Field Documentation

4.5.2.1 PSpiceFreqConstraint mFreq

Definition at line 386 of file PSpiceDigApiDefs.h.

4.5.2.2 PSpiceSetupHoldConstraint mSetupHold

Definition at line 384 of file PSpiceDigApiDefs.h.

4.5.2.3 PSpiceWidthConstraint mWidth

Definition at line 385 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.6 PSpiceDelay Class Reference

```
#include <PSpiceDigApiDefs.h>
```

Data Fields

- double mMinDelay
- double mTypDelay
- double mMaxDelay

4.6.1 Detailed Description

Definition at line 389 of file PSpiceDigApiDefs.h.

4.6.2 Field Documentation

4.6.2.1 double mMaxDelay

Definition at line 393 of file PSpiceDigApiDefs.h.

4.6.2.2 double mMinDelay

Definition at line 391 of file PSpiceDigApiDefs.h.

4.6.2.3 double mTypDelay

Definition at line 392 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.7 PSpiceDeviceInst Class Reference

This class defines basic PSpice Device Instance which will be used to transmit data to and from CMI Models.

`#include <PSpiceCMIApiDefs.h>`

Data Fields

- char * mInstID
- void * mData
- PSpiceDeviceModel * mModelData
- void * mSignals
- void * mStencil
- void * mState

4.7.1 Detailed Description

This class defines basic PSpice Device Instance which will be used to transmit data to and from CMI Models.

Definition at line 204 of file PSpiceCMIApiDefs.h.

4.7.2 Field Documentation

4.7.2.1 void* mData

Definition at line 207 of file PSpiceCMIApiDefs.h.

4.7.2.2 char* mInstID

Definition at line 206 of file PSpiceCMIApiDefs.h.

4.7.2.3 PSpiceDeviceModel* mModelData

Definition at line 208 of file PSpiceCMIApiDefs.h.

4.7.2.4 void* mSignals

Definition at line 209 of file PSpiceCMIApiDefs.h.

4.7.2.5 void* mState

Definition at line 211 of file PSpiceCMIApiDefs.h.

4.7.2.6 void* mStencil

Definition at line 210 of file PSpiceCMIApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceCMIApiDefs.h

4.8 PSpiceDeviceMiscInfo Class Reference

`#include <PSpiceCMIApiDefs.h>`

Data Fields

- double Temperature
- double MeasurementTemperature
- double Vt
- double RelTol
- unsigned int CurrentAnalysisNumber
- double GMin
- bool SkipBP
- double CKTag [4]
- double CKTomega

4.8.1 Detailed Description

Definition at line 214 of file PSpiceCMIApiDefs.h.

4.8.2 Field Documentation

4.8.2.1 double CKTag[4]

Definition at line 223 of file PSpiceCMIApiDefs.h.

4.8.2.2 double CKTomega

Definition at line 224 of file PSpiceCMIApiDefs.h.

4.8.2.3 unsigned int CurrentAnalysisNumber

Definition at line 220 of file PSpiceCMIApiDefs.h.

4.8.2.4 double GMin

Definition at line 221 of file PSpiceCMIApiDefs.h.

4.8.2.5 double MeasurementTemperature

Definition at line 217 of file PSpiceCMIApiDefs.h.

4.8.2.6 double RelTol

Definition at line 219 of file PSpiceCMIApiDefs.h.

4.8.2.7 bool SkipBP

Definition at line 222 of file PSpiceCMIApiDefs.h.

4.8.2.8 double Temperature

Definition at line 216 of file PSpiceCMIApiDefs.h.

4.8.2.9 double Vt

Definition at line 218 of file PSpiceCMIApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceCMIApiDefs.h

4.9 PSpiceDeviceModel Class Reference

This class defines basic PSpice Device Model which will be used to transmit data to and from CMI Models.

`#include <PSpiceCMIApiDefs.h>`

Data Fields

- char * mModelID
- void * mData

4.9.1 Detailed Description

This class defines basic PSpice Device Model which will be used to transmit data to and from CMI Models.

Definition at line 195 of file PSpiceCMIApiDefs.h.

4.9.2 Field Documentation

4.9.2.1 void* mData

Definition at line 198 of file PSpiceCMIApiDefs.h.

4.9.2.2 char* mModelID

Definition at line 197 of file PSpiceCMIApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceCMIApiDefs.h

4.10 PSpiceFreqConstraint Class Reference

Class for Frequency Constraint Definition.

`#include <PSpiceDigApiDefs.h>`

Public Member Functions

- PSpiceFreqConstraint ()

Data Fields

- bool mFreqSpecified
- char * mInputNode

  Name of input being checked.
- float min_freq

  Minimum frequency (Hz)
- float max_freq

  Maximum frequency (Hz)
- char errorflags

4.10.1 Detailed Description

Class for Frequency Constraint Definition.

Definition at line 350 of file PSpiceDigApiDefs.h.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 PSpiceFreqConstraint ( ) [inline]

Definition at line 371 of file PSpiceDigApiDefs.h.

4.10.3 Field Documentation

4.10.3.1 char errorflags

Definition at line 369 of file PSpiceDigApiDefs.h.

4.10.3.2 float max_freq

Maximum frequency (Hz)

Definition at line 367 of file PSpiceDigApiDefs.h.

4.10.3.3 bool mFreqSpecified

Definition at line 352 of file PSpiceDigApiDefs.h.

4.10.3.4 float min_freq

Minimum frequency (Hz)

Definition at line 362 of file PSpiceDigApiDefs.h.

4.10.3.5 char* mInputNode

Name of input being checked.

Definition at line 357 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.11 PSpiceInputSpec Class Reference

Input Buffer Specification.

`#include <PSpiceDigApiDefs.h>`

Data Fields

- double load

  capacitive load
- double inR

  input Resistive load
- double Tstore

  minimum storage time for chrg ctrl net

4.11.1 Detailed Description

Input Buffer Specification.

Definition at line 423 of file PSpiceDigApiDefs.h.

4.11.2 Field Documentation

4.11.2.1 double inR

input Resistive load

Definition at line 426 of file PSpiceDigApiDefs.h.

4.11.2.2 double load

capacitive load

Definition at line 425 of file PSpiceDigApiDefs.h.

4.11.2.3 double Tstore

minimum storage time for chrg ctrl net

Definition at line 427 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.12 PSpiceNetsList Class Reference

`#include <PSpiceDigApiDefs.h>`

Public Member Functions

- PSpiceNetsList ()

Data Fields

- char mNetName [1024]
- PSpiceNetsList * mNext

4.12.1 Detailed Description

Definition at line 271 of file PSpiceDigApiDefs.h.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 PSpiceNetsList ( ) [inline]

Definition at line 276 of file PSpiceDigApiDefs.h.

4.12.3 Field Documentation

4.12.3.1 char mNetName[1024]

Definition at line 273 of file PSpiceDigApiDefs.h.

4.12.3.2 PSpiceNetsList* mNext

Definition at line 274 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.13 PSpiceOutputSpec Class Reference

Output Buffer Specification.

`#include <PSpiceDigApiDefs.h>`

Public Member Functions

- PSpiceOutputSpec ()

Data Fields

- double l_drive

  drive R at low
- double h_drive

  drive R at high
- double z_drive

  leakage R for Z state
- double load

  capacitive load
- double tswhl [MAXIOLEVEL]

  switching time - high to low
- double tswlh [MAXIOLEVEL]

  switching time - low to high
- double pwrt

  pulse width rejection threshold

4.13.1 Detailed Description

Output Buffer Specification.

Definition at line 399 of file PSpiceDigApiDefs.h.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 PSpiceOutputSpec ( ) [inline]

Definition at line 409 of file PSpiceDigApiDefs.h.

4.13.3 Field Documentation

4.13.3.1 double h_drive

drive R at high

Definition at line 402 of file PSpiceDigApiDefs.h.

4.13.3.2 double l_drive

drive R at low

Definition at line 401 of file PSpiceDigApiDefs.h.

4.13.3.3 double load

capacitive load

Definition at line 404 of file PSpiceDigApiDefs.h.

4.13.3.4 double pwrt

pulse width rejection threshold

Definition at line 407 of file PSpiceDigApiDefs.h.

4.13.3.5 double tswhl[MAXIOLEVEL]

switching time - high to low

Definition at line 405 of file PSpiceDigApiDefs.h.

4.13.3.6 double tswlh[MAXIOLEVEL]

switching time - low to high

Definition at line 406 of file PSpiceDigApiDefs.h.

4.13.3.7 double z_drive

leakage R for Z state

Definition at line 403 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.14 PSpiceParamDesc Class Reference

PSpice Parameter Descriptor.

`#include <PSpiceCommonAPIDefs.h>`

Data Fields

- char Name [1024]
- short mVersion

4.14.1 Detailed Description

PSpice Parameter Descriptor.

Definition at line 335 of file PSpiceCommonAPIDefs.h.

4.14.2 Field Documentation

4.14.2.1 short mVersion

Parameter Version (currently unused)

Definition at line 338 of file PSpiceCommonAPIDefs.h.

4.14.2.2 char Name[1024]

Parameter Name

Definition at line 337 of file PSpiceCommonAPIDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceCommonAPIDefs.h

4.15 PSpicePort Class Reference

PSpice Port.

`#include <PSpiceDigApiDefs.h>`

Data Fields

- const char * mName
- int mNumber
- int mType

4.15.1 Detailed Description

PSpice Port.

Definition at line 431 of file PSpiceDigApiDefs.h.

4.15.2 Field Documentation

4.15.2.1 const char* mName

Definition at line 433 of file PSpiceDigApiDefs.h.

4.15.2.2 int mNumber

Definition at line 435 of file PSpiceDigApiDefs.h.

4.15.2.3 int mType

Definition at line 436 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.16 PSpiceSetupHoldConstraint Class Reference

`#include <PSpiceDigApiDefs.h>`

Public Member Functions

- PSpiceSetupHoldConstraint ()
- ~PSpiceSetupHoldConstraint ()

Data Fields

- bool mSetupHoldSpecified
- char mClockName [1024]
- PSpiceNetsList * mNetsList
- int mCountData
- float setuptime_lo
- float setuptime_hi
- float holdtime_lo
- float holdtime_hi
- float releasetime_lh
- float releasetime_hl
- bool clk_assertion

4.16.1 Detailed Description

Definition at line 281 of file PSpiceDigApiDefs.h.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 PSpiceSetupHoldConstraint ( ) [inline]

Definition at line 299 of file PSpiceDigApiDefs.h.

4.16.2.2 ~PSpiceSetupHoldConstraint ( ) [inline]

Definition at line 314 of file PSpiceDigApiDefs.h.

4.16.3 Field Documentation

4.16.3.1 bool clk_assertion

Definition at line 296 of file PSpiceDigApiDefs.h.

4.16.3.2 float holdtime_hi

Definition at line 291 of file PSpiceDigApiDefs.h.

4.16.3.3 float holdtime_lo

Definition at line 290 of file PSpiceDigApiDefs.h.

4.16.3.4 char mClockName[1024]

Definition at line 285 of file PSpiceDigApiDefs.h.

4.16.3.5 int mCountData

Definition at line 287 of file PSpiceDigApiDefs.h.

4.16.3.6 PSpiceNetsList* mNetsList

Definition at line 286 of file PSpiceDigApiDefs.h.

4.16.3.7 bool mSetupHoldSpecified

Definition at line 283 of file PSpiceDigApiDefs.h.

4.16.3.8 float releasetime_hl

Definition at line 293 of file PSpiceDigApiDefs.h.

4.16.3.9 float releasetime_lh

Definition at line 292 of file PSpiceDigApiDefs.h.

4.16.3.10 float setuptime_hi

Definition at line 289 of file PSpiceDigApiDefs.h.

4.16.3.11 float setuptime_lo

Definition at line 288 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.17 PSpiceSignalNodeList Class Reference

`#include <PSpiceCMIApiDefs.h>`

Data Fields

- char ** mNodeNames
- int mNodeCount

4.17.1 Detailed Description

Definition at line 227 of file PSpiceCMIApiDefs.h.

4.17.2 Field Documentation

4.17.2.1 int mNodeCount

Definition at line 231 of file PSpiceCMIApiDefs.h.

4.17.2.2 char** mNodeNames

Definition at line 230 of file PSpiceCMIApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceCMIApiDefs.h

4.18 PSpiceState Class Reference

Digital State.

`#include <PSpiceDigApiDefs.h>`

Public Member Functions

- bool isZ () const
- int getLevel () const
- PSpiceState & operator= (const int pBit)

Data Fields

- union {
  struct {
  unsigned level: 3
  unsigned str0: 6
  unsigned str1: 6
  unsigned _filler: 1
  unsigned hazardtype: 3
  unsigned multiple: 1
  unsigned msgid: 10
  unsigned notposted: 1
  unsigned persistent: 1
  } fields
  unsigned long stateVal
  } val

4.18.1 Detailed Description

Digital State.

Definition at line 440 of file PSpiceDigApiDefs.h.

4.18.2 Member Function Documentation

4.18.2.1 int getLevel ( ) const [inline]

Definition at line 461 of file PSpiceDigApiDefs.h.

4.18.2.2 bool isZ ( ) const [inline]

Definition at line 458 of file PSpiceDigApiDefs.h.

4.18.2.3 PSpiceState& operator= (const int *pBit*) [inline]

Definition at line 464 of file PSpiceDigApiDefs.h.

4.18.3 Field Documentation

4.18.3.1 unsigned _filler

Definition at line 447 of file PSpiceDigApiDefs.h.

4.18.3.2 struct { ... } fields

4.18.3.3 unsigned hazardtype

Definition at line 450 of file PSpiceDigApiDefs.h.

4.18.3.4 unsigned level

Definition at line 444 of file PSpiceDigApiDefs.h.

4.18.3.5 unsigned msgid

Definition at line 452 of file PSpiceDigApiDefs.h.

4.18.3.6 unsigned multiple

Definition at line 451 of file PSpiceDigApiDefs.h.

4.18.3.7 unsigned notposted

Definition at line 453 of file PSpiceDigApiDefs.h.

4.18.3.8 unsigned persistent

Definition at line 454 of file PSpiceDigApiDefs.h.

4.18.3.9 unsigned long stateVal

Definition at line 456 of file PSpiceDigApiDefs.h.

4.18.3.10 unsigned str0

Definition at line 445 of file PSpiceDigApiDefs.h.

4.18.3.11 unsigned str1

Definition at line 446 of file PSpiceDigApiDefs.h.

4.18.3.12 union { ... } val

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

4.19 PSpiceWidthConstraint Class Reference

Class for Width Constraint Definition.

`#include <PSpiceDigApiDefs.h>`

Public Member Functions

- PSpiceWidthConstraint ()

Data Fields

- bool mWidthSpecified
- char mInputNode [1024]

  Name of input being checked.
- float min_high

  Minimum time that input must remain in H state.
- float min_low

  Minimum time that input must remain in L state.

4.19.1 Detailed Description

Class for Width Constraint Definition.

Definition at line 323 of file PSpiceDigApiDefs.h.

4.19.2 Constructor & Destructor Documentation

4.19.2.1 PSpiceWidthConstraint ( ) [inline]

Definition at line 342 of file PSpiceDigApiDefs.h.

4.19.3 Field Documentation

4.19.3.1 float min_high

Minimum time that input must remain in H state.

Definition at line 335 of file PSpiceDigApiDefs.h.

4.19.3.2 float min_low

Minimum time that input must remain in L state.

Definition at line 340 of file PSpiceDigApiDefs.h.

4.19.3.3 char mInputNode[1024]

Name of input being checked.

Definition at line 330 of file PSpiceDigApiDefs.h.

4.19.3.4 bool mWidthSpecified

Definition at line 325 of file PSpiceDigApiDefs.h.

The documentation for this class was generated from the following file:

- code_latest/PSpiceDigApiDefs.h

Chapter 5

File Documentation

5.1 code_latest/PSpiceCMIApiDefs.h File Reference

```
#include "PSpiceCommonAPIDefs.h"
```

Data Structures

- class PSpiceDeviceModel

  This class defines basic PSpice Device Model which will be used to transmit data to and from CMI Models.
- class PSpiceDeviceInst

  This class defines basic PSpice Device Instance which will be used to transmit data to and from CMI Models.
- class PSpiceDeviceMiscInfo
- class PSpiceSignalNodeList

Typedefs

- typedef double(∗ pPSpiceGetVoltageNodes_t) (const char ∗pNode1, const char ∗pNode2)

  Get Voltage calculated by PSpice simulator in the last interation between 2 nodes.
- typedef double(∗ pPSpiceGetVoltageNodesI_t) (const char ∗pNode1, const char ∗pNode2)

  Get imaginary part of Voltage calculated by PSpice simulator in the last interation between 2 nodes (valid for ac analysis)
- typedef int(∗ pPSpiceGetCurrentStateIndex_t) (void)

  Get index of current state vector that needs to be accessed during transient analysis.
- typedef double(∗ pPSpiceGetDelta_t) (void)

  Get Current Time Step Value for transient analysis.
- typedef double(∗ pPSpiceGetDeltaPrevious_t) (int pTimeStepIndex)

  Get Previous time step values for transient analysis.
- typedef double(∗ pPSpiceVoltageTolerance_t) (double pValue1, double pValue2)

  Get maximum acceptable value of error in voltage which will not cause convergence failure (depends on RELTOL and VNTOL)
- typedef double(∗ pPSpiceGetFrequency_t) (void)

  Get the current frequency value set by PSpice simulator (for ac analysis)
- typedef void(∗ pPSpiceIntegrate_t) (double &pConductance, double &pCurrent, double pCapacitance, qi_def &pSV0, qi_def &pSV1, qi_def &pSV2, int pInitFlag)

  Integrate the charge and voltage for current and previous states to get the output conductance and current values.
- typedef double(∗ pPSpiceCurrentTErr_t) (const qi_def ∗pQI0, const qi_def ∗pQI1, const qi_def ∗pQI2, const qi_def ∗pQI3, void ∗pDevice)

Get max integration error acceptable to the device.

- typedef void(* pPSpiceUpdateStateVector_t) (char **, char *, int)
- typedef void(* pPSpiceAddInternalNode_t) (double var, char *&inode, const char *xnode)

  Add internal nodes/branches for the device.
- typedef void(* pPSpiceAddInternalNodeByName_t) (double var, char *&inode, const char *xnode, const char *InternalNodeName)

  Add internal nodes/branches for the device, allows setting a default name for the new node/branch.
- typedef void(* pPSpiceGetMatrixPtr_t) (double **pMatrixValueItem, const char *pNode1, const char *p↩ Node2)

  Get pointer to Matrix value based on input node-pair.
- typedef void(* pPSpiceGetRHSPtr_t) (double **pRHSPtr, const char *pNode)

  Get pointer to Matrix RHS value based on input node.
- typedef void(* pPSpiceApplyValueItem_t) (double *pMatrixValueItem, double pValue)

  Sets matrix pointer to the device conductance value.
- typedef void(* pPSpiceApplyValueItemComplex_t) (double *pMatrixValueItem, double pRealValue, double pImagValue)

  Sets matrix pointer to the device conductance - real and imaginary.
- typedef void(* pPSpiceAdjustValueItem_t) (double *pMatrixValueItem, double pValue)

  Updates matrix location by adding the input device conductance value.
- typedef void(* pPSpiceSetPWLDataDbl_t) (void *pRef, double *pX, double *pY, int pSize)

  Only valid for PWL models.
- typedef void(* pPSpiceSetPWLDataStr_t) (void *pRef, const char *pStr, int pSize)

  Only valid for PWL models.
- typedef void * PrimitivePtr
- typedef const char *(* pPrintDescription_t) (PSpiceDeviceInst *pDeviceInst)

  Model Dll API called by PSpice To get any text printed in the PSpice out file.
- typedef void(* pInstallFunction_t) (void)

  Model Dll API called by PSpice To install Models.
- typedef void(* pDefaultModel_t) (PSpiceDeviceModel *pDeviceModel)

  Model Dll API called by PSpice To create and initialize a CMI Model.
- typedef void(* pDeleteModel_t) (PSpiceDeviceModel *pDeviceModel)

  Model Dll API called by PSpice To Delete a CMI Model object.
- typedef int(* pDefaultModelParams_t) (PSpiceDeviceModel *pDeviceModel, int ParamCount)

  Model Dll API called by PSpice To set default model parameters.
- typedef int(* pSetModelParams_t) (PSpiceDeviceModel *pDeviceModel, PSpiceCMIParam **pParamVector, int pParamCount)

  Model Dll API called by PSpice To set specific model parameters.
- typedef int(* pModChk_t) (PSpiceDeviceModel *pDeviceModel)

  Model Dll API called by PSpice To Check validity of Model Parameters.
- typedef void(* pDefaultInstance_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To Create and Initialize a Device Instance.
- typedef void(* pDeleteInstance_t) (PSpiceDeviceInst *pDeviceInst)

  Model Dll API called by PSpice To Delete an existing Device Instance.
- typedef int(* pDefaultSignals_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To Create and Initialize Signal (Port) Data for a Device Instance.
- typedef int(* pDeleteSignals_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To Delete Signal Data for an existing Device Instance.
- typedef int(* pDefaultStencil_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To Create and Initialize Stencil (Matrix) Data for a Device Instance.
- typedef int(* pDeleteStencil_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To Delete Stencil (Matrix) Data for an existing Device Instance.

- typedef int(* pDefaultState_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To Create and Initialize State Data for a Device Instance.
- typedef int(* pDeleteState_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To Delete State Data for an existing Device Instance.
- typedef double(* pSetDevicePinCurrent_t) (PSpiceDeviceInst *pDeviceInst, int pPin, int pMode)

  Model Dll API called by PSpice To Set Device pin current for an instance.
- typedef void(* pSetDevicePinCurrentComplex_t) (PSpiceDeviceInst *pDeviceInst, int pin, double omega, double &pCurrentReal, double &pCurrentImag)

  Model Dll API called by PSpice To set complex value of device pin current for an instance (called during ac analysis)
- typedef void(* pBindTerminals_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo, int TerminalCount, PSpiceSignalNodeList *pNodeList)

  Model Dll API called by PSpice To bind terminals to their signals.
- typedef int(* pReserveNodes_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo, PSpiceSignalNodeList *pNodeList)

  Model Dll API called by PSpice To reserve memory in matrix for all value items needed by the device.
- typedef void(* pGetMatrixPointers_t) (PSpiceDeviceInst *InstKnot)

  Model Dll API called by PSpice To get matrix pointers for all value items of a device.
- typedef void(* pAddInternalNodes_t) (PSpiceDeviceInst *InstKnot, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To add internal nodes for the device.
- typedef int(* pPreload_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceModel *pDeviceModel)

  Model Dll API called by PSpice To preLoad a device (called only once for a simulation)
- typedef int(* pAC_Load_t) (PSpiceDeviceInst *pDeviceInst, double Omega)

  Model Dll API called by PSpice To load ac (frequency-dependent) model of the device.
- typedef int(* pTranLoad_t) (int ModeFl, int InitFl, int LoadFl, double pTemperature, PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo)

  Model Dll API called by PSpice To load transient model of the device.
- typedef double(* pGetIntercept_t) (PSpiceDeviceInst *pDeviceInst)

  Model Dll API called by PSpice For PWL Analysis.
- typedef double(* pNoise_t) (PSpiceDeviceInst *pDeviceInst, double pTemp, double pOmega, double *comps, int pCompsLength)

  Model Dll API called by PSpice To calculate device noise for a particular frequency.
- typedef double(* pGetBreakPoint_t) (PSpiceDeviceInst *pDeviceInst)

  Model Dll API called by PSpice To get simulation break point for a device.
- typedef void(* pTmpModDevice_t) (PSpiceDeviceInst *pDeviceInst, double ToldK, double TnewK, double TnomK, PSpiceDeviceMiscInfo *mMiscInfo)

  Model Dll API called by PSpice To load temperature changes for a device instance.
- typedef void(* pTmpModModel_t) (PSpiceDeviceModel *pDeviceModel, double ToldK, double TnewK, double TnomK)

  Model Dll API called by PSpice To load temperature changes for a device model.
- typedef double(* pTrunc_t) (PSpiceDeviceInst *pDeviceInst)

  Model Dll API called by PSpice To get integration error for a device.
- typedef void(* pSaveCheckpoint_t) (void *pDeviceInst, unsigned int pVersion, void *pStr, size_t pSize)

  Model Dll API called by PSpice To save checkpoint for a device.
- typedef void(* pLoadCheckpoint_t) (void *pDeviceInst, unsigned int pVersion, void *pStr, size_t pSize)

  Model Dll API called by PSpice To load checkpoint for a device.
- typedef bool(* pCheckTopology_t) (void *pDeviceInst, unsigned int pVersion, void *pStr)

  Model Dll API called by PSpice To check topology for the device.
- typedef void(* pSaveTopology_t) (void *pDeviceInst, unsigned int pVersion, void *pStr)

  Model Dll API called by PSpice To save topology for the device.
- typedef size_t(* pSetTopologySize_t) (void *pDeviceInst, unsigned int pVersion)

  Model Dll API called by PSpice To set topology size.
- typedef void(* descSetTitle_t) (PrimitivePtr handle, char *Title)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

- typedef void(∗ descSetMinTerminalCount_t) (PrimitivePtr handle, int MinTerminalCount)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetMaxTerminalCount_t) (PrimitivePtr handle, int MaxTerminalCount)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetTerminalNames_t) (PrimitivePtr handle, char ∗∗TerminalNames)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetTerminalNameCount_t) (PrimitivePtr handle, int TerminalNameCount)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetModelDataStructSize_t) (PrimitivePtr handle, unsigned int ModelDataStructSize)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetInstDataStructSize_t) (PrimitivePtr handle, unsigned int InstDataStructSize)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetDefaultModel_t) (PrimitivePtr handle, pDefaultModel_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSetModelParams_t) (PrimitivePtr handle, pSetModelParams_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetCheckModel_t) (PrimitivePtr handle, pModChk_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetDefaultInstance_t) (PrimitivePtr handle, pDefaultInstance_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSetInstanceParams_t) (PrimitivePtr handle, pSetInstanceParams_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetBindTerminals_t) (PrimitivePtr handle, pBindTerminals_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetDefaultState_t) (PrimitivePtr handle, pDefaultState_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetReserveNodes_t) (PrimitivePtr handle, pReserveNodes_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetGetMatrixPointers_t) (PrimitivePtr handle, pGetMatrixPointers_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetAddInternalNodes_t) (PrimitivePtr handle, pAddInternalNodes_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetPreload_t) (PrimitivePtr handle, pPreload_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetAC_Load_t) (PrimitivePtr handle, pAC_Load_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetTranLoad_t) (PrimitivePtr handle, pTranLoad_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetTrunc_t) (PrimitivePtr handle, pTrunc_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetNoise_t) (PrimitivePtr handle, pNoise_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetTmpModModel_t) (PrimitivePtr handle, pTmpModModel_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetTmpModDevice_t) (PrimitivePtr handle, pTmpModDevice_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSignalsStructSize_t) (PrimitivePtr handle, unsigned int SignalDataStructSize)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetStencilStructSize_t) (PrimitivePtr handle, unsigned int StencilDataStructSize)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.

- typedef void(∗ descSetStateStructSize_t) (PrimitivePtr handle, unsigned int StateDataStructSize)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetCheckPointSize_t) (PrimitivePtr handle, size_t CheckpointSize)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSaveCheckpoint_t) (PrimitivePtr handle, pSaveCheckpoint_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSaveTopology_t) (PrimitivePtr handle, pSaveTopology_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSetTopologySize_t) (PrimitivePtr handle, pSetTopologySize_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetCheckTopology_t) (PrimitivePtr handle, pCheckTopology_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetLoadCheckpoint_t) (PrimitivePtr handle, pLoadCheckpoint_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetDeleteModel_t) (PrimitivePtr handle, pDeleteModel_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetDeleteInstance_t) (PrimitivePtr handle, pDeleteInstance_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSetDevicePinCurrent_t) (PrimitivePtr handle, pSetDevicePinCurrent_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSetDevicePinCurrentComplex_t) (PrimitivePtr handle, pSetDevicePinCurrent↩ Complex_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef double(∗ descSetGetIntercept_t) (PrimitivePtr handle, pGetIntercept_t procedure)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef bool(∗ pIsPWLModel_t) (void)
- typedef const char ∗(∗ pPWLModelType_t) (void)
- typedef int(∗ pGetPWLData_t) (void ∗pDeviceInst, double ∗pXVector, double ∗pYVector, int pSize)
- typedef void(∗ pGetPWLDataStr_t) (void ∗pDeviceInst, const char ∗pValueStr, int pSize)
- typedef double(∗ pGetLastVoltage_t) (void)

5.1.1 Typedef Documentation

5.1.1.1 typedef void(∗ descSetAC_Load_t) (PrimitivePtr handle, pAC_Load_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to run ac analysis on a device

Returns

None

Definition at line 738 of file PSpiceCMIApiDefs.h.

5.1.1.2 typedef void(∗ descSetAddInternalNodes_t) (PrimitivePtr handle, pAddInternalNodes_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to create internal nodes for the model

Returns

None

Definition at line 722 of file PSpiceCMIApiDefs.h.

5.1.1.3 typedef void(∗ descSetBindTerminals_t) (PrimitivePtr handle, pBindTerminals_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to bind terminals to their signals

Returns

None

Definition at line 690 of file PSpiceCMIApiDefs.h.

5.1.1.4 typedef void(∗ descSetCheckModel_t) (PrimitivePtr handle, pModChk_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to check model data for accuracy

Returns

None

Definition at line 666 of file PSpiceCMIApiDefs.h.

5.1.1.5 typedef void(∗ descSetCheckPointSize_t) (PrimitivePtr handle, size_t CheckpointSize)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*CheckpointSize*	size of checkpoint data

Returns

None

Definition at line 810 of file PSpiceCMIApiDefs.h.

5.1.1.6 typedef void(∗ descSetCheckTopology_t) (PrimitivePtr handle, pCheckTopology_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for checking topology (Used during checkpoint restart)

Returns

None

Definition at line 842 of file PSpiceCMIApiDefs.h.

5.1.1.7 typedef void(∗ descSetDefaultInstance_t) (PrimitivePtr handle, pDefaultInstance_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to reset device instance

Returns

None

Definition at line 674 of file PSpiceCMIApiDefs.h.

5.1.1.8 typedef void(∗ descSetDefaultModel_t) (PrimitivePtr handle, pDefaultModel_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to set the model to its default values

Returns

None

Definition at line 649 of file PSpiceCMIApiDefs.h.

5.1.1.9 typedef void(∗ descSetDefaultState_t) (PrimitivePtr handle, pDefaultState_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to create default state

Returns

None

Definition at line 698 of file PSpiceCMIApiDefs.h.

5.1.1.10 typedef void(∗ descSetDeleteInstance_t) (PrimitivePtr handle, pDeleteInstance_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for Deleting Instance object

Returns

None

Definition at line 874 of file PSpiceCMIApiDefs.h.

5.1.1.11 typedef void(∗ descSetDeleteModel_t) (PrimitivePtr handle, pDeleteModel_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for Deleting Model object

Returns

None

Definition at line 866 of file PSpiceCMIApiDefs.h.

5.1.1.12 typedef double(∗ descSetGetIntercept_t) (PrimitivePtr handle, pGetIntercept_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for getting intercept value (for PWL Analysis)

Returns

None

Definition at line 898 of file PSpiceCMIApiDefs.h.

5.1.1.13 typedef void(∗ descSetGetMatrixPointers_t) (PrimitivePtr handle, pGetMatrixPointers_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to get matrix locations for all matrix entries

Returns

None

Definition at line 714 of file PSpiceCMIApiDefs.h.

5.1.1.14 typedef void(∗ descSetInstDataStructSize_t) (PrimitivePtr handle, unsigned int InstDataStructSize)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Set size of Device Data

Parameters

in	*handle*	Model Reference
in	*InstDataStruct↩ Size*	Size of Instance Data

Returns

None

Definition at line 641 of file PSpiceCMIApiDefs.h.

5.1.1.15 typedef void(∗ descSetLoadCheckpoint_t) (PrimitivePtr handle, pLoadCheckpoint_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for loading checkpoint

Returns

None

Definition at line 850 of file PSpiceCMIApiDefs.h.

5.1.1.16 typedef void(∗ descSetMaxTerminalCount_t) (PrimitivePtr handle, int MaxTerminalCount)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

May be used in future - this information provided by this API is not used right now

Parameters

in	*handle*	Model Reference
in	*MaxTerminal↩ Count*	Maximum number of terminals required by the model

Returns

None

Definition at line 605 of file PSpiceCMIApiDefs.h.

5.1.1.17 typedef void(∗ descSetMinTerminalCount_t) (PrimitivePtr handle, int MinTerminalCount)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

May be used in future - this information provided by this API is not used right now

Parameters

in	*handle*	Model Reference
in	*MinTerminal↩ Count*	Minimum number of terminals required by the model

Returns

None

Definition at line 596 of file PSpiceCMIApiDefs.h.

5.1.1.18 typedef void(∗ descSetModelDataStructSize_t) (PrimitivePtr handle, unsigned int ModelDataStructSize)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Set size of Model Data

Parameters

in	*handle*	Model Reference
in	*ModelData↩ StructSize*	Size of Model Data

Returns

None

Definition at line 632 of file PSpiceCMIApiDefs.h.

5.1.1.19 typedef void(∗ descSetNoise_t) (PrimitivePtr handle, pNoise_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for running noise analysis of a device

Returns

None

Definition at line 762 of file PSpiceCMIApiDefs.h.

5.1.1.20 typedef void(∗ descSetPreload_t) (PrimitivePtr handle, pPreload_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to preload a device (called only once)

Returns

None

Definition at line 730 of file PSpiceCMIApiDefs.h.

5.1.1.21 typedef void(∗ descSetReserveNodes_t) (PrimitivePtr handle, pReserveNodes_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to reserve memory for node pointers in matrix

Returns

None

Definition at line 706 of file PSpiceCMIApiDefs.h.

5.1.1.22 typedef void(∗ descSetSaveCheckpoint_t) (PrimitivePtr handle, pSaveCheckpoint_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for saving checkpoint

Returns

None

Definition at line 818 of file PSpiceCMIApiDefs.h.

5.1.1.23 typedef void(∗ descSetSaveTopology_t) (PrimitivePtr handle, pSaveTopology_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for saving topology (used during checkpoint save)

Returns

None

Definition at line 826 of file PSpiceCMIApiDefs.h.

5.1.1.24 typedef void(∗ descSetSetDevicePinCurrent_t) (PrimitivePtr handle, pSetDevicePinCurrent_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for setting Current for each pin of a device

Returns

None

Definition at line 882 of file PSpiceCMIApiDefs.h.

5.1.1.25 typedef void(∗ descSetSetDevicePinCurrentComplex_t) (PrimitivePtr handle, pSetDevicePinCurrent↩ Complex_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for setting complex current value for each pin of a device (for ac analysis)

Returns

None

Definition at line 890 of file PSpiceCMIApiDefs.h.

5.1.1.26 typedef void(∗ descSetSetInstanceParams_t) (PrimitivePtr handle, pSetInstanceParams_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to set instance parameters

Returns

None

Definition at line 682 of file PSpiceCMIApiDefs.h.

5.1.1.27 typedef void(∗ descSetSetModelParams_t) (PrimitivePtr handle, pSetModelParams_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to set model parameters

Returns

None

Definition at line 658 of file PSpiceCMIApiDefs.h.

5.1.1.28 typedef void(∗ descSetSetTopologySize_t)(PrimitivePtr handle, pSetTopologySize_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for setting topology size (used during checkpoint save and restart)

Returns

None

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for Setting Topology Size (Used during Checkpoint save and restart)

Returns

None

Definition at line 834 of file PSpiceCMIApiDefs.h.

5.1.1.29 typedef void(∗ descSetSignalsStructSize_t) (PrimitivePtr handle, unsigned int SignalDataStructSize)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*SignalData↩ StructSize*	size of Signal Data

Returns

None

Definition at line 786 of file PSpiceCMIApiDefs.h.

5.1.1.30 typedef void(∗ descSetStateStructSize_t) (PrimitivePtr handle, unsigned int StateDataStructSize)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*StateData↩ StructSize*	size of State Data

Returns

None

Definition at line 802 of file PSpiceCMIApiDefs.h.

5.1.1.31 typedef void(∗ descSetStencilStructSize_t) (PrimitivePtr handle, unsigned int StencilDataStructSize)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*StencilData↩ StructSize*	size of Stencil (Matrix) Data

Returns

None

Definition at line 794 of file PSpiceCMIApiDefs.h.

5.1.1.32 typedef void(∗ descSetTerminalNameCount_t) (PrimitivePtr handle, int TerminalNameCount)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Used by PSpice Simulator to validate that the number of terminals set from the netlist is correct

Parameters

in	*handle*	Model Reference
in	*TerminalName↩ Count*	Minimum number of terminals required by the model

Returns

None

Definition at line 623 of file PSpiceCMIApiDefs.h.

5.1.1.33 typedef void(∗ descSetTerminalNames_t) (PrimitivePtr handle, char ∗∗TerminalNames)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

May be used for printing in future - this information provided by this API is not used right now

Parameters

in	*handle*	Model Reference
in	*TerminalNames*	List of all Internal Terminal Names of the Model

Returns

None

Definition at line 614 of file PSpiceCMIApiDefs.h.

5.1.1.34 typedef void(∗ descSetTitle_t) (PrimitivePtr handle, char ∗Title)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Definition at line 587 of file PSpiceCMIApiDefs.h.

5.1.1.35 typedef void(∗ descSetTmpModDevice_t) (PrimitivePtr handle, pTmpModDevice_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for running temperature analysis of a device

Returns

None

Definition at line 778 of file PSpiceCMIApiDefs.h.

5.1.1.36 typedef void(∗ descSetTmpModModel_t) (PrimitivePtr handle, pTmpModModel_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer for running temperature analysis of a model

Returns

None

Definition at line 770 of file PSpiceCMIApiDefs.h.

5.1.1.37 typedef void(∗ descSetTranLoad_t) (PrimitivePtr handle, pTranLoad_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to run transient analysis on a device

Returns

None

Definition at line 746 of file PSpiceCMIApiDefs.h.

5.1.1.38 typedef void(∗ descSetTrunc_t) (PrimitivePtr handle, pTrunc_t procedure)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*handle*	Model Reference
in	*procedure*	function pointer to get integration error for a device

Returns

None

Definition at line 754 of file PSpiceCMIApiDefs.h.

5.1.1.39 typedef int(∗ pAC_Load_t) (PSpiceDeviceInst ∗pDeviceInst, double Omega)

Model Dll API called by PSpice To load ac (frequency-dependent) model of the device.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*Omega*	AC Frequency

Returns

Status

Definition at line 446 of file PSpiceCMIApiDefs.h.

5.1.1.40 typedef void(∗ pAddInternalNodes_t) (PSpiceDeviceInst ∗InstKnot, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To add internal nodes for the device.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

None

Definition at line 428 of file PSpiceCMIApiDefs.h.

5.1.1.41 typedef void(∗ pBindTerminals_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo, int TerminalCount, PSpiceSignalNodeList ∗pNodeList)

Model Dll API called by PSpice To bind terminals to their signals.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info
in	*TerminalCount*	Number of terminals
in	*pNodeList*	List of Device Terminals

Returns

None

Definition at line 401 of file PSpiceCMIApiDefs.h.

5.1.1.42 typedef bool(∗ pCheckTopology_t) (void ∗pDeviceInst, unsigned int pVersion, void ∗pStr)

Model Dll API called by PSpice To check topology for the device.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pVersion*	checkpoint version
in	*pStr*	topology string

Returns

status

Definition at line 548 of file PSpiceCMIApiDefs.h.

5.1.1.43 typedef void(∗ pDefaultInstance_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To Create and Initialize a Device Instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

None

Definition at line 305 of file PSpiceCMIApiDefs.h.

5.1.1.44 typedef void(∗ pDefaultModel_t) (PSpiceDeviceModel ∗pDeviceModel)

Model Dll API called by PSpice To create and initialize a CMI Model.

Parameters

in	*pDeviceModel*	Device Model Pointer

Returns

None

Definition at line 261 of file PSpiceCMIApiDefs.h.

5.1.1.45 typedef int(* pDefaultModelParams_t) (PSpiceDeviceModel *pDeviceModel, int ParamCount)

Model Dll API called by PSpice To set default model parameters.

Parameters

in	*pDeviceModel*	Device Model Pointer
in	*ParamCount*	Number of Model Parameters

Returns

Status

Definition at line 278 of file PSpiceCMIApiDefs.h.

5.1.1.46 typedef int(∗ pDefaultSignals_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To Create and Initialize Signal (Port) Data for a Device Instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

Status

Definition at line 323 of file PSpiceCMIApiDefs.h.

5.1.1.47 typedef int(∗ pDefaultState_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To Create and Initialize State Data for a Device Instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

Status

Definition at line 359 of file PSpiceCMIApiDefs.h.

5.1.1.48 typedef int(∗ pDefaultStencil_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To Create and Initialize Stencil (Matrix) Data for a Device Instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

Status

Definition at line 341 of file PSpiceCMIApiDefs.h.

5.1.1.49 typedef void(∗ pDeleteInstance_t) (PSpiceDeviceInst ∗pDeviceInst)

Model Dll API called by PSpice To Delete an existing Device Instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer

Returns

None

Definition at line 313 of file PSpiceCMIApiDefs.h.

5.1.1.50 typedef void(∗ pDeleteModel_t) (PSpiceDeviceModel ∗pDeviceModel)

Model Dll API called by PSpice To Delete a CMI Model object.

Parameters

in	*pDeviceModel*	Device Model Pointer

Returns

None

Definition at line 269 of file PSpiceCMIApiDefs.h.

5.1.1.51 typedef int(∗ pDeleteSignals_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To Delete Signal Data for an existing Device Instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

None

Definition at line 332 of file PSpiceCMIApiDefs.h.

5.1.1.52 typedef int(∗ pDeleteState_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To Delete State Data for an existing Device Instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

None

Definition at line 368 of file PSpiceCMIApiDefs.h.

5.1.1.53 typedef int(∗ pDeleteStencil_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To Delete Stencil (Matrix) Data for an existing Device Instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

None

Definition at line 350 of file PSpiceCMIApiDefs.h.

5.1.1.54 typedef double(∗ pGetBreakPoint_t) (PSpiceDeviceInst ∗pDeviceInst)

Model Dll API called by PSpice To get simulation break point for a device.

Parameters

in	*pDeviceInst*	Device Instance Pointer

Returns

breakpoint value

Definition at line 486 of file PSpiceCMIApiDefs.h.

5.1.1.55 typedef double(∗ pGetIntercept_t) (PSpiceDeviceInst ∗pDeviceInst)

Model Dll API called by PSpice For PWL Analysis.

Evaluates distance of each model from the constant slope region boundary enum value = PSPICE_GET_INTER↩ CEPT

Parameters

in	*pDeviceInst*	Device Instance Pointer

Returns

Intercept value

Definition at line 468 of file PSpiceCMIApiDefs.h.

5.1.1.56 typedef double(∗ pGetLastVoltage_t) (void)

Definition at line 910 of file PSpiceCMIApiDefs.h.

5.1.1.57 typedef void(∗ pGetMatrixPointers_t) (PSpiceDeviceInst ∗InstKnot)

Model Dll API called by PSpice To get matrix pointers for all value items of a device.

Parameters

in	*pDeviceInst*	Device Instance Pointer

Returns

None

Definition at line 419 of file PSpiceCMIApiDefs.h.

5.1.1.58 typedef int(∗ pGetPWLData_t) (void ∗pDeviceInst, double ∗pXVector, double ∗pYVector, int pSize)

Definition at line 906 of file PSpiceCMIApiDefs.h.

5.1.1.59 typedef void(∗ pGetPWLDataStr_t) (void ∗pDeviceInst, const char ∗pValueStr, int pSize)

Definition at line 907 of file PSpiceCMIApiDefs.h.

5.1.1.60 typedef void(∗ pInstallFunction_t) (void)

Model Dll API called by PSpice To install Models.

Parameters

None	

Returns

None

Definition at line 253 of file PSpiceCMIApiDefs.h.

5.1.1.61 typedef bool(∗ pIsPWLModel_t) (void)

Definition at line 902 of file PSpiceCMIApiDefs.h.

5.1.1.62 typedef void(∗ pLoadCheckpoint_t) (void ∗pDeviceInst, unsigned int pVersion, void ∗pStr, size_t pSize)

Model Dll API called by PSpice To load checkpoint for a device.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pVersion*	checkpoint version
in	*pStr*	checkpoint data
in	*pSize*	checkpoint data size

Returns

None

Definition at line 538 of file PSpiceCMIApiDefs.h.

5.1.1.63 typedef int(∗ pModChk_t) (PSpiceDeviceModel ∗pDeviceModel)

Model Dll API called by PSpice To Check validity of Model Parameters.

Parameters

in	*pDeviceModel*	Device Model Pointer

Returns

String to be printed out

Definition at line 296 of file PSpiceCMIApiDefs.h.

5.1.1.64 typedef double(∗ pNoise_t) (PSpiceDeviceInst ∗pDeviceInst, double pTemp, double pOmega, double ∗comps, int pCompsLength)

Model Dll API called by PSpice To calculate device noise for a particular frequency.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pTemp*	Temperature
in	*pOmega*	Frequency

Returns

Device Noise value

Definition at line 478 of file PSpiceCMIApiDefs.h.

5.1.1.65 typedef int(∗ pPreload_t) (PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceModel ∗pDeviceModel)

Model Dll API called by PSpice To preLoad a device (called only once for a simulation)

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pDeviceModel*	Model Pointer

Returns

status

Definition at line 437 of file PSpiceCMIApiDefs.h.

5.1.1.66 typedef const char∗(∗ pPrintDescription_t) (PSpiceDeviceInst ∗pDeviceInst)

Model Dll API called by PSpice To get any text printed in the PSpice out file.

Definition at line 245 of file PSpiceCMIApiDefs.h.

5.1.1.67 typedef void(∗ pPSpiceAddInternalNode_t) (double var, char ∗&inode, const char ∗xnode)

Add internal nodes/branches for the device.

Parameters

in	*var*	Parameter on which internal node addition is dependent
out	*inode*	Internal node - internal node created if var != 0
in	*xnode*	Internal node shorted to existing node xnode if var == 0

Definition at line 116 of file PSpiceCMIApiDefs.h.

5.1.1.68 typedef void(∗ pPSpiceAddInternalNodeByName_t) (double var, char ∗&inode, const char ∗xnode, const char ∗InternalNodeName)

Add internal nodes/branches for the device, allows setting a default name for the new node/branch.

Parameters

in	*var*	Parameter on which internal node addition is dependent
out	*inode*	Internal node - internal node created if var != 0
in	*xnode*	Internal node shorted to existing node xnode if var == 0
in	*InternalNode↩ Name*	Default Name for the new node

Definition at line 125 of file PSpiceCMIApiDefs.h.

5.1.1.69 typedef void(∗ pPSpiceAdjustValueItem_t) (double ∗pMatrixValueItem, double pValue)

Updates matrix location by adding the input device conductance value.

Parameters

in	*pMatrixValue↩ Item*	Pointer to matrix value item
in	*pValue*	Real value to be added to an existing matrix value item

Returns

None

Definition at line 168 of file PSpiceCMIApiDefs.h.

5.1.1.70 typedef void(∗ pPSpiceApplyValueItem_t) (double ∗pMatrixValueItem, double pValue)

Sets matrix pointer to the device conductance value.

Parameters

in	*pMatrixValue↩ Item*	Pointer to real matrix value item
in	*pValue*	Real value to be written in matrix

Returns

None

Definition at line 150 of file PSpiceCMIApiDefs.h.

5.1.1.71 typedef void(∗ pPSpiceApplyValueItemComplex_t) (double ∗pMatrixValueItem, double pRealValue, double pImagValue)

Sets matrix pointer to the device conductance - real and imaginary.

Valud for ac analysis

Parameters

in	*pMatrixValue↩ Item*	Pointer to complex matrix value item
in	*pRealValue*	Real value to be written in matrix
in	*pImagValue*	Imaginary value to be written in matrix

Returns

None

Definition at line 159 of file PSpiceCMIApiDefs.h.

5.1.1.72 typedef double(∗ pPSpiceCurrentTErr_t) (const qi_def ∗pQI0, const qi_def ∗pQI1, const qi_def ∗pQI2, const qi_def ∗pQI3, void ∗pDevice)

Get max integration error acceptable to the device.

Parameters

in	*pQI0*	Current state vector
in	*pQI1*	Previous first state vector
in	*pQI2*	Previous second state vector
in	*pQI3*	Previous third state vector
in	*pDevice*	Device Pointer

Returns

Truncation error in terms of minimum time step recommended by the device

Definition at line 105 of file PSpiceCMIApiDefs.h.

5.1.1.73 typedef int(∗ pPSpiceGetCurrentStateIndex_t) (void)

Get index of current state vector that needs to be accessed during transient analysis.

Parameters

None	

Returns

State index currently being used by engine (valid for transient analysis)

Definition at line 52 of file PSpiceCMIApiDefs.h.

5.1.1.74 typedef double(∗ pPSpiceGetDelta_t) (void)

Get Current Time Step Value for transient analysis.

Parameters

None	

Returns

Get current Time Step value (valid for transient analysis)

Definition at line 59 of file PSpiceCMIApiDefs.h.

5.1.1.75 typedef double(∗ pPSpiceGetDeltaPrevious_t) (int pTimeStepIndex)

Get Previous time step values for transient analysis.

Parameters

in	*pTimeStepIndex*	Previous Time Step Index (valid values are 1|2|3

Returns

Get the respective Time Step value (valid for transient analysis)

Definition at line 66 of file PSpiceCMIApiDefs.h.

5.1.1.76 typedef double(∗ pPSpiceGetFrequency_t) (void)

Get the current frequency value set by PSpice simulator (for ac analysis)

Parameters

None	

Returns

Get current Frequency value (valid for ac analysis)

Definition at line 81 of file PSpiceCMIApiDefs.h.

5.1.1.77 typedef void(* pPSpiceGetMatrixPtr_t) (double **pMatrixValueItem, const char *pNode1, const char *pNode2)

Get pointer to Matrix value based on input node-pair.

Parameters

out	*pMatrixValue↩ Item*	Pointer to matrix value item - generated for the input node names
in	*pNode1*	Name of Node1
in	*pNode2*	Name of Node2

Returns

None

Definition at line 134 of file PSpiceCMIApiDefs.h.

5.1.1.78 typedef void(* pPSpiceGetRHSPtr_t) (double **pRHSPtr, const char *pNode)

Get pointer to Matrix RHS value based on input node.

Parameters

out	*pMatrixValue↩ Item*	Pointer to Matrix RHS - generated for the input node name
in	*pNode1*	Name of Node1

Returns

None

Definition at line 142 of file PSpiceCMIApiDefs.h.

5.1.1.79 typedef double(* pPSpiceGetVoltageNodes_t) (const char *pNode1, const char *pNode2)

Get Voltage calculated by PSpice simulator in the last interation between 2 nodes.

Definition at line 37 of file PSpiceCMIApiDefs.h.

5.1.1.80 typedef double(* pPSpiceGetVoltageNodesI_t) (const char *pNode1, const char *pNode2)

Get imaginary part of Voltage calculated by PSpice simulator in the last interation between 2 nodes (valid for ac analysis)

Parameters

in	*pNode1*	Name of Node1
in	*pNode2*	Name of Node2

Returns

Imaginary part of Voltage calculated between Node1 and Node2 (valid for ac analysis only)

Definition at line 45 of file PSpiceCMIApiDefs.h.

5.1.1.81 typedef void(* pPSpiceIntegrate_t) (double &pConductance, double &pCurrent, double pCapacitance, qi_def &pSV0, qi_def &pSV1, qi_def &pSV2, int pInitFlag)

Integrate the charge and voltage for current and previous states to get the output conductance and current values.

Parameters

out	*pConductance*	Equivalent conductance calculated based on integrated charges and currents
out	*pCurrent*	Equivalent current calculated based on integrated charges and currents
in	*pCapacitance*	Capacitance of branch
in	*pSV0*	Current state vector
in	*pSV1*	Previous first state vector
in	*pSV2*	Previous second state vector
in	*pInitFlag*	Init Flag

Returns

None

Definition at line 94 of file PSpiceCMIApiDefs.h.

5.1.1.82 typedef void(∗ pPSpiceSetPWLDataDbl_t) (void ∗pRef, double ∗pX, double ∗pY, int pSize)

Only valid for PWL models.

Parameters

in	*pRef*	Device Reference pointer
in	*pX*	array of X values
in	*pY*	array of corresponding Y-values
in	*pSize*	array size

Returns

None

Definition at line 179 of file PSpiceCMIApiDefs.h.

5.1.1.83 typedef void(∗ pPSpiceSetPWLDataStr_t) (void ∗pRef, const char ∗pStr, int pSize)

Only valid for PWL models.

Parameters

in	*pRef*	Device Reference pointer
in	*pStr*	comma-separated string of PWL x and y values, e.g. "1,.1,2,.5,3,.9"
in	*pSize*	array size

Returns

None

Definition at line 188 of file PSpiceCMIApiDefs.h.

5.1.1.84 typedef void(∗ pPSpiceUpdateStateVector_t) (char ∗∗, char ∗, int)

Definition at line 107 of file PSpiceCMIApiDefs.h.

5.1.1.85 typedef double(∗ pPSpiceVoltageTolerance_t) (double pValue1, double pValue2)

Get maximum acceptable value of error in voltage which will not cause convergence failure (depends on RELTOL and VNTOL)

Parameters

in	*pValue1*	First voltage value
in	*pValue2*	Second voltage value

Returns

Maximum Voltage tolerance for the voltage differential

Definition at line 74 of file PSpiceCMIApiDefs.h.

5.1.1.86 typedef const char*(* pPWLModelType_t) (void)

Definition at line 903 of file PSpiceCMIApiDefs.h.

5.1.1.87 typedef int(* pReserveNodes_t) (PSpiceDeviceInst *pDeviceInst, PSpiceDeviceMiscInfo *pMiscInfo, PSpiceSignalNodeList *pNodeList)

Model Dll API called by PSpice To reserve memory in matrix for all value items needed by the device.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info
in	*pNodeList*	List of Device Terminals

Returns

status

Definition at line 411 of file PSpiceCMIApiDefs.h.

5.1.1.88 typedef void* PrimitivePtr

Definition at line 190 of file PSpiceCMIApiDefs.h.

5.1.1.89 typedef void(* pSaveCheckpoint_t) (void *pDeviceInst, unsigned int pVersion, void *pStr, size_t pSize)

Model Dll API called by PSpice To save checkpoint for a device.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pVersion*	checkpoint version
out	*pStr*	checkpoint data
out	*pSize*	checkpoint data size

Returns

None

Definition at line 527 of file PSpiceCMIApiDefs.h.

5.1.1.90 typedef void(* pSaveTopology_t) (void *pDeviceInst, unsigned int pVersion, void *pStr)

Model Dll API called by PSpice To save topology for the device.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pVersion*	checkpoint version
out	*pStr*	topology string

Returns

None

Definition at line 558 of file PSpiceCMIApiDefs.h.

5.1.1.91 typedef double(∗ pSetDevicePinCurrent_t) (PSpiceDeviceInst ∗pDeviceInst, int pPin, int pMode)

Model Dll API called by PSpice To Set Device pin current for an instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pPin*	Pin Index
in	*pMode*	Simulation Mode

Returns

Current value

Definition at line 378 of file PSpiceCMIApiDefs.h.

5.1.1.92 typedef void(∗ pSetDevicePinCurrentComplex_t) (PSpiceDeviceInst ∗pDeviceInst, int pin, double omega, double &pCurrentReal, double &pCurrentImag)

Model Dll API called by PSpice To set complex value of device pin current for an instance (called during ac analysis)

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pPin*	Pin Index
in	*pOmega*	Current Frequency
out	*pCurrentReal*	Real part of Current
out	*pCurrentImag*	Imaginary part of Current

Returns

None

Definition at line 390 of file PSpiceCMIApiDefs.h.

5.1.1.93 typedef int(∗ pSetModelParams_t) (PSpiceDeviceModel ∗pDeviceModel, PSpiceCMIParam ∗∗pParamVector, int pParamCount)

Model Dll API called by PSpice To set specific model parameters.

Parameters

in	*pDeviceModel*	Device Instance Pointer
in	*pParamVector*	Vector of PSpiceCMIParam objects
in	*pParamCount*	Count of Parameters in pParamVector

Returns

Status

Definition at line 288 of file PSpiceCMIApiDefs.h.

5.1.1.94 typedef size_t(∗ pSetTopologySize_t) (void ∗pDeviceInst, unsigned int pVersion)

Model Dll API called by PSpice To set topology size.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*pVersion*	checkpoint version

Returns

topology size

Definition at line 567 of file PSpiceCMIApiDefs.h.

5.1.1.95 typedef void(∗ pTmpModDevice_t) (PSpiceDeviceInst ∗pDeviceInst, double ToldK, double TnewK, double TnomK, PSpiceDeviceMiscInfo ∗mMiscInfo)

Model Dll API called by PSpice To load temperature changes for a device instance.

Parameters

in	*pDeviceInst*	Device Instance Pointer
in	*ToldK*	Old Temperature
in	*TnewK*	New Temperature
in	*TnomK*	Nominal Temperature

Returns

None

Definition at line 497 of file PSpiceCMIApiDefs.h.

5.1.1.96 typedef void(∗ pTmpModModel_t) (PSpiceDeviceModel ∗pDeviceModel, double ToldK, double TnewK, double TnomK)

Model Dll API called by PSpice To load temperature changes for a device model.

Parameters

in	*pDeviceModel*	Device Model Pointer
in	*ToldK*	Old Temperature
in	*TnewK*	New Temperature
in	*TnomK*	Nominal Temperature

Returns

None

Definition at line 508 of file PSpiceCMIApiDefs.h.

5.1.1.97 typedef int(∗ pTranLoad_t) (int ModeFl, int InitFl, int LoadFl, double pTemperature, PSpiceDeviceInst ∗pDeviceInst, PSpiceDeviceMiscInfo ∗pMiscInfo)

Model Dll API called by PSpice To load transient model of the device.

Parameters

in	*ModeFl*	Mode flag
in	*InitFl*	Init flag
in	*LoadFl*	Load flag
in	*Temperature*	Simulation Temperature
in	*pDeviceInst*	Device Instance Pointer
in	*pMiscInfo*	Misc Simulation Info

Returns

Device Convergence status

Definition at line 459 of file PSpiceCMIApiDefs.h.

5.1.1.98 typedef double(∗ pTrunc_t) (PSpiceDeviceInst ∗pDeviceInst)

Model Dll API called by PSpice To get integration error for a device.

Parameters

in	*pDeviceInst*	Device Instance Pointer

Returns

Current maximum time step supported by device

Definition at line 516 of file PSpiceCMIApiDefs.h.

5.2 code_latest/PSpiceCommonAPIDefs.h File Reference

```
#include <time.h>
#include <float.h>
```

Data Structures

- class PSpiceParamDesc

  PSpice Parameter Descriptor.
- class PSpiceAnyScalar

  PSpice Parameter Value subclass.
- class PSpiceAnyValue

  PSpice Parameter Value Class.
- class PSpiceCMIParam

  PSpice Parameter Top-level class.

Macros

- #define PSP_CMI_EXPORT

  File containing API's required by PSpice for loading external C++ model dll's for analog and digital models.
- #define CDLL_FUNC extern "C" __declspec(dllexport)

Typedefs

- typedef void(∗ pFnPtr_t) (void ∗)

  PSpice Function templates - the templates starting with descSet∗ are used to set model dll function pointers into PSpice engine Model Dll must export a function " void PSpiceInstallFunction(); " Inside this function, function pointers for each model install function must be set.
- typedef void(∗ descSetInstallFunction_t) (pFnPtr_t)

  Set function pointer to Model Installation Function Function used to set pointer to model install function by model dll enum value = PSPICE_SET_INSTALLFUNC Usage Example: pdescSetInstallFunction_t fp_descSetInstallFunction; fp_descSetInstallFunction(&installCap);.
- typedef void(∗ pFnPtr1_t) (void ∗, void ∗∗pFunctionPointerList)

  Set function pointer to an alternate Model Installation Function Function used to set pointer to model install function by model dll enum value = PSPICE_SET_INSTALLFUNC1.
- typedef void(∗ descSetInstallFunction1_t) (pFnPtr1_t)

  Another type of install function - it returns the function pointers directly in a pre-defined order This will be used for PWL Models for now Expected return data - bool isPWLModel(void)
- typedef void(∗ descSetVersion_t) (void ∗handle, const char ∗pVersion)

  set DMI Version
- typedef void(∗ descSetName_t) (void ∗handle, const char ∗Name)

  set Model Name
- typedef const char ∗(∗ pPSpiceGetLicenseString_t) ()

  Following functions allow the model dll to set specific function pointers for each model.
- typedef const char ∗(∗ pPSPICEGetOptionsParams_t) ()

  Get name and value pairs of all PSpice options.
- typedef void(∗ pPSpiceSetSimulationTemperature_t) (double pTemperature)

  Set Simulation Temperature enum value = PSPICE_SET_SIMTEMP.
- typedef void(∗ pPSpiceSetProbeTitle_t) (const char ∗pTitle)

  Set Title of Probe Section.
- typedef double(∗ pPSpiceGetCurrentAnalogTime_t) (void)

  Get Current Analog TIME value as set by PSpice Simulator.
- typedef double(∗ pPSpiceGetCurrentDigitalTime_t) (void)

  Get Current Digital TIME value as set by PSpice Simulator.
- typedef void ∗(∗ pPSpiceGetDevice_t) (const char ∗)

  Get Device pointer for an instance name.
- typedef const char ∗(∗ pPSpiceGetParamValue_t) (const char ∗pParamName)

  Get value of global parameter from PSpice enum value = PSPICE_GET_PARAMVALUE.
- typedef double(∗ pPSpiceGetParamValueDbl_t) (const char ∗)

  Get value of global parameter from PSpice enum value = PSPICE_GET_PARAMVALUEDBL.
- typedef void(∗ pPSpiceWriteToOut_t) (const char ∗pText)

  Write text to PSpice Out File enum value = PSPICE_SET_WRITETOOUT.
- typedef int(∗ pDefaultInstanceParams_t) (void ∗pInstKnot, int ParamCount)

  Create default instance parameters (Not supported in current version)
- typedef int(∗ pSetInstanceParams_t) (void ∗pInstKnot, PSpiceCMIParam ∗∗, int pParamCount)

  Called by PSpice Simulator to set Instance Parameters enum value = PSPICE_SET_INSTPARAMS.

Enumerations

- enum modeFlags {
  MDBPDC, MDBPTR, MDDCSW, MDTRAN,
  MDAC, MDINITSMSIG, MDTRANOP, MDUIC,
  MDINITTRAN }

  Mode Flags.

- enum initFlags {
  INNORM, ININIT, INOFF, INSTV0,
  INTRAN, INPRDCT }

  Initialization Flags.
- enum PSpiceAPIs {
  PSPICE_SET_INSTALLFUNC = 1, PSPICE_SET_SIMTEMP, PSPICE_SET_PROBETITLE, PSPICE_GE↩
  T_ANALOGTIME,
  PSPICE_GET_DIGITALTIME, PSPICE_GET_PARAMVALUE, PSPICE_GET_PARAMVALUEDBL, PSPI↩
  CE_SET_DELAY,
  PSPICE_SET_CONSTRAINT, PSPICE_SET_INSPEC, PSPICE_SET_OUTSPEC, PSPICE_GET_INSPEC,
  PSPICE_GET_OUTSPEC, PSPICE_GET_TIMINGVALUE, PSPICE_ADD_INTERNALNODE, PSPICE_G↩
  ET_MATRIXPTR,
  PSPICE_APPLY_VALUEITEM, PSPICE_APPLY_VALUEITEMCMPLX, PSPICE_GET_V, PSPICE_GET_↩
  VI,
  PSPICE_GET_DELTA, PSPICE_GET_DELTAPREV, PSPICE_VOLTAGE_TOL, PSPICE_CURRENT_T↩
  OL,
  PSPICE_INTEGRATE, PSPICE_GET_STATE, PSPICE_GMINLOAD, PSPICE_GET_ACFREQUENCY,
  PSPICE_SET_INSTDATASTRUCTSIZE, PSPICE_SET_MODELDATASTRUCTSIZE, PSPICE_SET_SIG↩
  NALSTRUCTSIZE, PSPICE_SET_STATESTRUCTSIZE,
  PSPICE_SET_RESERVENODES, PSPICE_SET_ADDINTERNALNODES, PSPICE_SET_GETMATRIXP↩
  TR, PSPICE_SET_BINDTERMINALS,
  PSPICE_SET_DEFAULTSTATE, PSPICE_SET_MODELPARAMDESC, PSPICE_SET_STENCILSTRUC↩
  TSIZE, PSPICE_SET_PRELOAD,
  PSPICE_SET_TRANLOAD, PSPICE_SET_ACLOAD, PSPICE_SET_TRUNC, PSPICE_SET_NOISE,
  PSPICE_SET_GETBREAKPOINT, PSPICE_SET_TMPMODMODEL, PSPICE_SET_TMPMODDEVICE,
  PSPICE_SET_DEFMODEL,
  PSPICE_SET_MODELPARAMS, PSPICE_SET_MODCHK, PSPICE_SET_DEFINST, PSPICE_SET_INS↩
  TPARAMS,
  PSPICE_SET_VERSION, PSPICE_SET_NAME, PSPICE_UPDATE_SV, PSPICE_CURRENT_TERR,
  PSPICE_SET_PRINTDESC, PSPICE_SET_SAVECHECKPT, PSPICE_SET_LOADCHECKPT, PSPICE_↩
  SET_SAVETOPOLOGY,
  PSPICE_SET_CHECKTOPOLOGY, PSPICE_SET_DELETEINST, PSPICE_SET_DELETEMODEL, PSP↩
  ICE_SET_SETPINI,
  PSPICE_SET_SETPINICMPLX, PSPICE_DIG_CREATEDEVICE, PSPICE_DIG_DELETEDEVICE, PSPI↩
  CE_DIG_GETTERMCOUNT,
  PSPICE_DIG_SETTERMCOUNT, PSPICE_DIG_EVALDEVICE, PSPICE_DIG_SETTERM, PSPICE_DIG↩
  _GETTERM,
  PSPICE_DIG_GETTERMTYPE, PSPICE_DIG_SETPARAM, PSPICE_DIG_SETPRINTDESC, PSPICE_↩
  SET_WRITETOOUT,
  PSPICE_DIG_FNCHANGED, PSPICE_DIG_FNTRANSITION, PSPICE_DIG_INITDEV, PSPICE_DIG_S↩
  ETSTATE,
  PSPICE_DIG_GETPARAMVALUE, PSPICE_DIG_TICK_FROM_TIME, PSPICE_DIG_TIME_FROM_TICK,
  PSPICE_SET_GETTOPOLOGYSIZE,
  PSPICE_SET_CHECKPTSIZE, PSPICE_GET_LICENSE, PSPICE_GET_OPTIONPARAMS, PSPICE_A↩
  DJUST_VALUEITEM,
  PSPICE_GET_RHSPTR, PSPICE_ADD_INTERNALNODEBYNAME, PSPICE_INTERNAL0, PSPICE_G↩
  ET_RHSPTR1,
  PSPICE_SET_GETINTERCEPT, PSPICE_SET_INSTALLFUNC1, PSPICE_SET_PWLDATA_DBL, PSPI↩
  CE_SET_PWLDATA_STR }

  PSpice APIs Enumeration.
- enum PSpiceValueType { PSPICE_VALUE_REALTYPE, PSPICE_VALUE_STRINGTYPE, PSPICE_VAL↩
  UE_EXPRTYPE }

  PSpice Parameter Value Types.

Functions

- CDLL_FUNC void __cdecl pspiceSetFunctionList (void **pPtr)

Function used to set pointers to API's exposed from PSpice Engine.

- CDLL_FUNC void __cdecl PSpiceInstallFunction ()

  PSpice searches for this function by name - this needs to be exported by the model dll.

- CDLL_FUNC void __cdecl PSpiceInstallFunction1 ()

5.2.1 Macro Definition Documentation

5.2.1.1 #define CDLL_FUNC extern "C" __declspec(dllexport)

Definition at line 31 of file PSpiceCommonAPIDefs.h.

5.2.1.2 #define PSP_CMI_EXPORT

File containing API's required by PSpice for loading external C++ model dll's for analog and digital models.

Definition at line 28 of file PSpiceCommonAPIDefs.h.

5.2.2 Typedef Documentation

5.2.2.1 typedef void(∗ descSetInstallFunction1_t) (pFnPtr1_t)

Another type of install function - it returns the function pointers directly in a pre-defined order This will be used for PWL Models for now Expected return data - bool isPWLModel(void)

Parameters

in	*Model*	install function

Definition at line 115 of file PSpiceCommonAPIDefs.h.

5.2.2.2 typedef void(∗ descSetInstallFunction_t) (pFnPtr_t)

Set function pointer to Model Installation Function Function used to set pointer to model install function by model dll enum value = PSPICE_SET_INSTALLFUNC Usage Example: pdescSetInstallFunction_t fp_descSetInstall↩ Function; fp_descSetInstallFunction(&installCap);.

Parameters

in	*Model*	install function

Returns

None

Definition at line 98 of file PSpiceCommonAPIDefs.h.

5.2.2.3 typedef void(∗ descSetName_t) (void ∗handle, const char ∗Name)

set Model Name

Parameters

in	*handle*	Model Reference
in	*Name*	Model Name

Returns

None

Definition at line 132 of file PSpiceCommonAPIDefs.h.

5.2.2.4 typedef void(∗ descSetVersion_t) (void ∗handle, const char ∗pVersion)

set DMI Version

Parameters

in	*handle*	Model Reference (created by PSpice Engine - passed on to Model dll)
in	*DMIVersion*	DMI Version (Supported value = 1.0)

Returns

None

Definition at line 124 of file PSpiceCommonAPIDefs.h.

5.2.2.5 typedef int(∗ pDefaultInstanceParams_t) (void ∗pInstKnot, int ParamCount)

Create default instance parameters (Not supported in current version)

Definition at line 216 of file PSpiceCommonAPIDefs.h.

5.2.2.6 typedef void(∗ pFnPtr1_t) (void ∗, void ∗∗pFunctionPointerList)

Set function pointer to an alternate Model Installation Function Function used to set pointer to model install function by model dll enum value = PSPICE_SET_INSTALLFUNC1.

Parameters

in	*Fixed-size*	List of function pointers in a pre-defined order

Definition at line 106 of file PSpiceCommonAPIDefs.h.

5.2.2.7 typedef void(∗ pFnPtr_t) (void ∗)

PSpice Function templates - the templates starting with descSet∗ are used to set model dll function pointers into P↩ Spice engine Model Dll must export a function " void PSpiceInstallFunction(); " Inside this function, function pointers for each model install function must be set.

Function templates to be used by model dll to set and call functions in PSpice

Definition at line 86 of file PSpiceCommonAPIDefs.h.

5.2.2.8 typedef double(∗ pPSpiceGetCurrentAnalogTime_t) (void)

Get Current Analog TIME value as set by PSpice Simulator.

Returns

Current Analog TIME value

Definition at line 170 of file PSpiceCommonAPIDefs.h.

5.2.2.9 typedef double(∗ pPSpiceGetCurrentDigitalTime_t) (void)

Get Current Digital TIME value as set by PSpice Simulator.

Returns

Current Digital TIME value

Definition at line 176 of file PSpiceCommonAPIDefs.h.

5.2.2.10 typedef void∗(∗ pPSpiceGetDevice_t) (const char ∗)

Get Device pointer for an instance name.

Returns

Pointer to device

Definition at line 182 of file PSpiceCommonAPIDefs.h.

5.2.2.11 typedef const char∗(∗ pPSpiceGetLicenseString_t) ()

Following functions allow the model dll to set specific function pointers for each model.

Function Templates Exposed by PSpice Engine dll Get Name of License being used by PSpice Simulator

Returns

License string

Definition at line 143 of file PSpiceCommonAPIDefs.h.

5.2.2.12 typedef const char∗(∗ pPSPICEGetOptionsParams_t) ()

Get name and value pairs of all PSpice options.

Returns

string of the form "a=b\nc=d" etc.

Definition at line 149 of file PSpiceCommonAPIDefs.h.

5.2.2.13 typedef const char∗(∗ pPSpiceGetParamValue_t) (const char ∗pParamName)

Get value of global parameter from PSpice enum value = PSPICE_GET_PARAMVALUE.

Parameters

in	*pParamName*	Parameter Name

Returns

Value as string

Definition at line 190 of file PSpiceCommonAPIDefs.h.

5.2.2.14 typedef double(* pPSpiceGetParamValueDbl_t) (const char *)

Get value of global parameter from PSpice enum value = PSPICE_GET_PARAMVALUEDBL.

Returns

double value of parameter if it exists else MAXREAL

Definition at line 198 of file PSpiceCommonAPIDefs.h.

5.2.2.15 typedef void(* pPSpiceSetProbeTitle_t) (const char *pTitle)

Set Title of Probe Section.

Parameters

in	*pTitle*	Text Title for each section in dat file

Returns

None

Definition at line 164 of file PSpiceCommonAPIDefs.h.

5.2.2.16 typedef void(* pPSpiceSetSimulationTemperature_t) (double pTemperature)

Set Simulation Temperature enum value = PSPICE_SET_SIMTEMP.

Parameters

in	*pTemperature*	New Temperature at which simulation needs to be run (in Celsius)

Returns

None

Definition at line 157 of file PSpiceCommonAPIDefs.h.

5.2.2.17 typedef void(* pPSpiceWriteToOut_t) (const char *pText)

Write text to PSpice Out File enum value = PSPICE_SET_WRITETOOUT.

Parameters

in	*pText*	String to be written to out file

Returns

none

Definition at line 206 of file PSpiceCommonAPIDefs.h.

5.2.2.18 typedef int(* pSetInstanceParams_t) (void *pInstKnot, PSpiceCMIParam **, int pParamCount)

Called by PSpice Simulator to set Instance Parameters enum value = PSPICE_SET_INSTPARAMS.

Definition at line 222 of file PSpiceCommonAPIDefs.h.

5.2.3 Enumeration Type Documentation

5.2.3.1 enum initFlags

Initialization Flags.

Init Flag controls where the device evaluation code gets its terminal voltages from.

Enumerator

INNORM Use VltVct values from last iteration

ININIT Use IC= values in device table (transient bias point) or value calculated and stored in model table (reg. bias point)

INOFF Use VltVct unless device has OFF flag, then use 0

INSTV0 Use STVCT0 (= value used previous iteration)

INTRAN Use STVCT1 (= value used previous step)

INPRDCT Extrapolate using STVCT2 and STVCT1

Definition at line 51 of file PSpiceCommonAPIDefs.h.

5.2.3.2 enum modeFlags

Mode Flags.

Mode controls some options during the iterations.

Enumerator

MDBPDC Small signal bias point calculation

MDBPTR Bias point for transient analysis

MDDCSW DC Sweep which uses previous step

MDTRAN Transient analysis

MDAC AC analysis

MDINITSMSIG

MDTRANOP

MDUIC

MDINITTRAN

Definition at line 36 of file PSpiceCommonAPIDefs.h.

5.2.3.3 enum PSpiceAPIs

PSpice APIs Enumeration.

These are used to set the function pointers into the Model dll. are used by pspiceSetFunctionList

Enumerator

PSPICE_SET_INSTALLFUNC Both A&D: Install CMI Model

PSPICE_SET_SIMTEMP Both A&D: Set simulation temperature

PSPICE_SET_PROBETITLE Both A&D: Set probe section title

PSPICE_GET_ANALOGTIME Analog: Get Time

PSPICE_GET_DIGITALTIME Digital: Get Time

PSPICE_GET_PARAMVALUE Both A&D: Get parameter value as string

PSPICE_GET_PARAMVALUEDBL Both A&D: Get parameter value as double
PSPICE_SET_DELAY Digital: Set Delay
PSPICE_SET_CONSTRAINT Digital: Set constraint
PSPICE_SET_INSPEC Digital: Setup Input Buffer specification
PSPICE_SET_OUTSPEC Digital: Setup Output Buffer specification
PSPICE_GET_INSPEC Digital: Get Input Buffer specification
PSPICE_GET_OUTSPEC Digital: Get Output Buffer specification
PSPICE_GET_TIMINGVALUE Digital: get Timing specification
PSPICE_ADD_INTERNALNODE Analog: Add internal node
PSPICE_GET_MATRIXPTR Analog: Get Matrix pointer for a value item
PSPICE_APPLY_VALUEITEM Analog: Set value to a matrix pointer
PSPICE_APPLY_VALUEITEMCMPLX Analog: Set value to a complex matrix pointer
PSPICE_GET_V Analog: Get voltage for a node from simulator
PSPICE_GET_VI Analog
PSPICE_GET_DELTA Analog: Get current time step
PSPICE_GET_DELTAPREV Analog: Get previous time steps
PSPICE_VOLTAGE_TOL Analog: Get voltage tolerance
PSPICE_CURRENT_TOL Analog: Get current tolerance
PSPICE_INTEGRATE Analog: integrate
PSPICE_GET_STATE Analog: Get current State Index
PSPICE_GMINLOAD Analog: Load for Gmin Stepping
PSPICE_GET_ACFREQUENCY Analog: get frequency for ac analysis
PSPICE_SET_INSTDATASTRUCTSIZE Analog: Set inst data size (descSet)
PSPICE_SET_MODELDATASTRUCTSIZE Analog: set Model data size (descSet)
PSPICE_SET_SIGNALSTRUCTSIZE Analog: set signal data size (descSet)
PSPICE_SET_STATESTRUCTSIZE Analog: set state data size (descSet)
PSPICE_SET_RESERVENODES Analog: Reserve memory in matrix (descSet)
PSPICE_SET_ADDINTERNALNODES Analog: Add internal nodes for model (descSet)
PSPICE_SET_GETMATRIXPTR Analog: Get matrix pointers for all value items (descSet)
PSPICE_SET_BINDTERMINALS Analog: Bind terminals to matrix locations (descSet)
PSPICE_SET_DEFAULTSTATE Unused
PSPICE_SET_MODELPARAMDESC Analog: Set Model parameters (descSet)
PSPICE_SET_STENCILSTRUCTSIZE Analog: Set stencil (matrix) size (descSet)
PSPICE_SET_PRELOAD Analog: Set Preload for device (descSet)
PSPICE_SET_TRANLOAD Analog: Set Transient load for device (descSet)
PSPICE_SET_ACLOAD Analog: Set AC load for device (descSet)
PSPICE_SET_TRUNC Analog: Get integration error (descSet)
PSPICE_SET_NOISE Analog: Set noise load for device (descSet)
PSPICE_SET_GETBREAKPOINT Analog: Get breakpoint (descSet)
PSPICE_SET_TMPMODMODEL Analog: Set Temperature load for model (descSet)
PSPICE_SET_TMPMODDEVICE Analog: Set Temperature load for device (descSet)
PSPICE_SET_DEFMODEL Analog: Create Default model (descSet)
PSPICE_SET_MODELPARAMS Analog: Set Model Parameters (descSet)
PSPICE_SET_MODCHK Analog: Check Model (descSet)
PSPICE_SET_DEFINST Analog: Create default device (descSet)

PSPICE_SET_INSTPARAMS Analog: Set device parameters (descSet)
PSPICE_SET_VERSION Both A&D: Set CMI Version (descSet)
PSPICE_SET_NAME Both A&D: Set Name (descSet)
PSPICE_UPDATE_SV Analog: Update state vector
PSPICE_CURRENT_TERR Analog: Get current integration error
PSPICE_SET_PRINTDESC Both A&D: Print Description (descSet)
PSPICE_SET_SAVECHECKPT Both A&D: Save checkpoint (descSet)
PSPICE_SET_LOADCHECKPT Both A&D: Load checkpoint (descSet)
PSPICE_SET_SAVETOPOLOGY Both A&D: Save topology (for checkpoint) (descSet)
PSPICE_SET_CHECKTOPOLOGY Both A&D: Check topology (for checkpoint) (descSet)
PSPICE_SET_DELETEINST Analog: Delete device (descSet)
PSPICE_SET_DELETEMODEL Analog: Delete Model (descSet)
PSPICE_SET_SETPINI Analog: Set current value (descSet)
PSPICE_SET_SETPINICMPLX Analog: Set complex current value (for ac analysis) (descSet)
PSPICE_DIG_CREATEDEVICE Digital: Create device
PSPICE_DIG_DELETEDEVICE Digital: Delete Device
PSPICE_DIG_GETTERMCOUNT Digital: Get Terminal count
PSPICE_DIG_SETTERMCOUNT Digital: Set terminal count
PSPICE_DIG_EVALDEVICE Digital: Evaluate device
PSPICE_DIG_SETTERM Digital: Set Terminal value
PSPICE_DIG_GETTERM Digital: Get Terminal value
PSPICE_DIG_GETTERMTYPE Digital: Get Terminal Type
PSPICE_DIG_SETPARAM Digital: Unused
PSPICE_DIG_SETPRINTDESC Digital: Print Description of device to out file
PSPICE_SET_WRITETOOUT Both A&D: Write text to out file
PSPICE_DIG_FNCHANGED Digital: Get State change
PSPICE_DIG_FNTRANSITION Digital: get transition
PSPICE_DIG_INITDEV Digital: Initialize Device
PSPICE_DIG_SETSTATE Digital: set state
PSPICE_DIG_GETPARAMVALUE Both A&D: Get parameter value as string
PSPICE_DIG_TICK_FROM_TIME Digital: get digital ticks value
PSPICE_DIG_TIME_FROM_TICK Digital: get digital time value
PSPICE_SET_GETTOPOLOGYSIZE Both A&D: get topology size for checkpoint
PSPICE_SET_CHECKPTSIZE Both A&D: set checkpoint size
PSPICE_GET_LICENSE Both A&D: get name of license being used by PSpice
PSPICE_GET_OPTIONPARAMS Both A&D: get option value
PSPICE_ADJUST_VALUEITEM Analog: Add to matrix value item
PSPICE_GET_RHSPTR Analog: Get RHS pointer for the matrix
PSPICE_ADD_INTERNALNODEBYNAME Analog: Add internal node by name
PSPICE_INTERNAL0 For internal use only
PSPICE_GET_RHSPTR1 For PWL Simulation
PSPICE_SET_GETINTERCEPT For PWL Simulation - get intercept
PSPICE_SET_INSTALLFUNC1 New version of install - takes a list of function pointers as parameter
PSPICE_SET_PWLDATA_DBL Set PWL Data as vectors of x and y values
PSPICE_SET_PWLDATA_STR Set PWL Data as a single string value

Definition at line 228 of file PSpiceCommonAPIDefs.h.

5.2.3.4 enum PSpiceValueType

PSpice Parameter Value Types.

Enumerator

PSPICE_VALUE_REALTYPE Double value

PSPICE_VALUE_STRINGTYPE String value

PSPICE_VALUE_EXPRTYPE Expression - evaluated to a double value by PSpice engine

Definition at line 328 of file PSpiceCommonAPIDefs.h.

5.2.4 Function Documentation

5.2.4.1 CDLL_FUNC void __cdecl PSpiceInstallFunction ( )

PSpice searches for this function by name - this needs to be exported by the model dll.

Parameters

None	

Returns

None

5.2.4.2 CDLL_FUNC void __cdecl PSpiceInstallFunction1 ( )

5.2.4.3 CDLL_FUNC void __cdecl pspiceSetFunctionList (void ** *pPtr*)

Function used to set pointers to API's exposed from PSpice Engine.

Parameters

`in`	*List*	of function pointers to various functions - the order of pointers is given by enum PSpiceAPIs

Returns

None

5.3 code_latest/PSpiceDigApiDefs.h File Reference

```
#include "PSpiceCommonAPIDefs.h"
```

Data Structures

- class PSpiceNetsList
- class PSpiceSetupHoldConstraint
- class PSpiceWidthConstraint

  Class for Width Constraint Definition.
- class PSpiceFreqConstraint

  Class for Frequency Constraint Definition.

- class PSpiceConstraint

  Composite class for definition of all constraints.
- class PSpiceDelay
- class PSpiceOutputSpec

  Output Buffer Specification.
- class PSpiceInputSpec

  Input Buffer Specification.
- class PSpicePort

  PSpice Port.
- class PSpiceState

  Digital State.
- class pspBit

Macros

- #define UNSPEC 1e-33
- #define PSP_VALUE_NOT_DEFINED(x) (x>=MAXREAL?true:false)
- #define MAXIOLEVEL 4

Typedefs

- typedef void *(* pCreateDevice_t) (const char *pInstName, void *ref)

  *Templates for API's that need to be exposed by the Model Dll - their names follow the convention p*_t*
  Their pointers will be set by the Model Dll into PSpice Engine using the descSet Functions.*
- typedef void(* pInitDevice_t) (void *pRef)
- typedef bool(* pPSpiceEvaluateDevice_t) (void *pRef, double pTicks, PSpiceState *pVectorStates, int pSize)
- typedef void(* pPSpiceSetState_t) (void *pRef, int pPortIndex, PSpiceState *pState, const PSpiceDelay *p↩ Delay)
- typedef void(* pPSpiceGetParameterValue_t) (void *pRef, PSpiceCMIParam *pParam)
- typedef void(* pDeleteDevice_t) (void *pRef)
- typedef int(* pGetDeviceTermCount_t) (void *pRef)
- typedef bool(* pSetDeviceTermCount_t) (void *pRef, int pTermCount)
- typedef int(* pGetDeviceTermValue_t) (void *pRef, PSpiceState *pVectorStates, int pSize)

  Get Term values from Model Dll to PSpice.
- typedef int(* pSetDeviceTermValue_t) (void *pRef, PSpiceState *pVectorStates, int pSize)

  Set Term values from PSpice to Model Dll.
- typedef void(* pGetDeviceTermTypes_t) (void *pRef, void *pVectorPorts)
- typedef bool(* pSetParameter_t) (void *pRef, int pParamCount, void **pVectorParameter)
- typedef const char *(* pDigPrintDescription_t) (void *pRef)
- typedef double(* pGetTicksFromTime_t) (double pTime)
- typedef double(* pGetTimeFromTicks_t) (double pTicks)
- typedef bool(* pPSpiceChanged_t) (void *pRef, const char *pNetName, double pTime, int pChanged)
- typedef bool(* pPSpiceGetTransition_t) (void *pRef, const char *pOut, const char *pTransition)
- typedef void(* descSetCreateDevice_t) (void *pRef, void *pFuncPtr)

  Function templates to be used by model dll to set and call functions in PSpice.
- typedef void(* descSetDeleteDevice_t) (void *pRef, void *pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(* descSetGetDeviceTermCount_t) (void *pRef, void *pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(* descSetInitDevice_t) (void *pRef, void *pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.

- typedef void(∗ descSetSetDeviceTermCount_t) (void ∗pRef, void ∗pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetEvaluateDevice_t) (void ∗pRef, void ∗pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetGetDeviceTermValue_t) (void ∗pRef, void ∗pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSetDeviceTermValue_t) (void ∗pRef, void ∗pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetGetDeviceTerminals_t) (void ∗pRef, void ∗pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ descSetSetParameter_t) (void ∗pRef, void ∗pFuncPtr)

  PSpice API called by Model Dll to set function pointers into PSpice Engine.
- typedef void(∗ pPSpiceSetDelay_t) (void ∗pRef, const PSpiceDelay ∗pDelay)

  PSpice API called by Model Dll.
- typedef void(∗ pPSpiceSetConstraint_t) (void ∗pRef, const PSpiceConstraint ∗pConstraint)

  PSpice API called by Model Dll.
- typedef void(∗ pPSpiceSetInputSpec_t) (void ∗pRef, const PSpiceInputSpec ∗pInSpec)

  Following functions should be called before evaluate - they make changes on all terminals of the device.
- typedef void(∗ pPSpiceSetOutputSpec_t) (void ∗pRef, PSpiceOutputSpec ∗pOutSpec)

  Setup Output Buffer Specification enum value = PSPICE_SET_OUTSPEC.
- typedef void(∗ pPSpiceGetInputSpec_t) (void ∗pRef, PSpiceInputSpec &pInSpec)

  Get Input Buffer Specification enum value = PSPICE_GET_INSPEC.
- typedef void(∗ pPSpiceGetOutputSpec_t) (void ∗pRef, const PSpiceOutputSpec &pOutSpec)

  Get Output Buffer Specification enum value = PSPICE_GET_OUTSPEC.
- typedef bool(∗ pPSpiceGetTimingModelValue_t) (void ∗pRef, const char ∗pParamName, PSpiceDelay &p↩ Delay)

  Get Timing Model Value enum value = PSPICE_GET_TIMINGVALUE.

Enumerations

- enum PSPICE_PORT_TYPE { PSPICE_PORT_IO = 0, PSPICE_PORT_IN = 1 }

Functions

- PSP_CMI_EXPORT void PSpiceSetDelay (void ∗pRef, const PSpiceDelay ∗pDelay)
- PSP_CMI_EXPORT void PSpiceSetConstraint (void ∗pRef, const PSpiceConstraint ∗pConstraint)
- PSP_CMI_EXPORT void PSpiceSetOutputSpec (void ∗pRef, const PSpiceOutputSpec ∗pOutSpec)
- PSP_CMI_EXPORT void PSpiceSetInputSpec (void ∗pRef, const PSpiceInputSpec ∗pInSpec)
- PSP_CMI_EXPORT void PSpiceGetOutputSpec (void ∗pRef, PSpiceOutputSpec &pOutputSpec)
- PSP_CMI_EXPORT void PSpiceGetInputSpec (void ∗pRef, PSpiceInputSpec &pInputSpec)
- PSP_CMI_EXPORT bool PSpiceGetTimingModelValue (void ∗pRef, const char ∗pParamName, PSpiceDelay &pDelay)
- PSP_CMI_EXPORT void PSpiceSetState (void ∗pRef, int pPortIndex, PSpiceState ∗pState, const PSpice↩ Delay ∗pDelay)
- PSP_CMI_EXPORT void PSpiceGetParameterValue (void ∗pRef, PSpiceCMIParam ∗pParam)
- bool operator== (const pspBit &pBit1, const pspBit &pBit2)
- pspBit operator^ (const pspBit &pBit1, const pspBit &pBit2)
- pspBit operator& (const pspBit &pBit1, const pspBit &pBit2)
- pspBit operator| (const pspBit &pBit1, const pspBit &pBit2)
- pspBit operator~ (const pspBit &pBit)

5.3.1 Macro Definition Documentation

5.3.1.1 #define MAXIOLEVEL 4

Definition at line 396 of file PSpiceDigApiDefs.h.

5.3.1.2 #define PSP_VALUE_NOT_DEFINED(*x*) (x>=MAXREAL?true:false)

Definition at line 7 of file PSpiceDigApiDefs.h.

5.3.1.3 #define UNSPEC 1e-33

Definition at line 6 of file PSpiceDigApiDefs.h.

5.3.2 Typedef Documentation

5.3.2.1 typedef void(* descSetCreateDevice_t) (void *pRef, void *pFuncPtr)

Function templates to be used by model dll to set and call functions in PSpice.

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to create Device function

Definition at line 135 of file PSpiceDigApiDefs.h.

5.3.2.2 typedef void(* descSetDeleteDevice_t) (void *pRef, void *pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to delete Device function

Definition at line 142 of file PSpiceDigApiDefs.h.

5.3.2.3 typedef void(* descSetEvaluateDevice_t) (void *pRef, void *pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to evaluate Device function

Definition at line 171 of file PSpiceDigApiDefs.h.

5.3.2.4 typedef void(* descSetGetDeviceTermCount_t) (void *pRef, void *pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

This function call is optional

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to get device terminal count function

Definition at line 150 of file PSpiceDigApiDefs.h.

5.3.2.5 typedef void(∗ descSetGetDeviceTerminals_t) (void ∗pRef, void ∗pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to get device terminals function

Definition at line 192 of file PSpiceDigApiDefs.h.

5.3.2.6 typedef void(∗ descSetGetDeviceTermValue_t) (void ∗pRef, void ∗pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to get Device Terminal value function

Definition at line 178 of file PSpiceDigApiDefs.h.

5.3.2.7 typedef void(∗ descSetInitDevice_t) (void ∗pRef, void ∗pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to initialize Device function

Definition at line 157 of file PSpiceDigApiDefs.h.

5.3.2.8 typedef void(∗ descSetSetDeviceTermCount_t) (void ∗pRef, void ∗pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to set Device terminal count function

Definition at line 164 of file PSpiceDigApiDefs.h.

5.3.2.9 typedef void(∗ descSetSetDeviceTermValue_t) (void ∗pRef, void ∗pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to set device terminal value function

Definition at line 185 of file PSpiceDigApiDefs.h.

5.3.2.10 typedef void(∗ descSetSetParameter_t) (void ∗pRef, void ∗pFuncPtr)

PSpice API called by Model Dll to set function pointers into PSpice Engine.

Parameters

in	*pRef*	Instance Reference
in	*pFuncPtr*	Pointer to set Parameter function

Definition at line 199 of file PSpiceDigApiDefs.h.

5.3.2.11 typedef void∗(∗ pCreateDevice_t) (const char ∗pInstName, void ∗ref)

Templates for API's that need to be exposed by the Model Dll - their names follow the convention p∗_t
Their pointers will be set by the Model Dll into PSpice Engine using the descSet∗ Functions.

Parameters

in	*Instance*	Name

Returns

Digital Descriptor Reference

Definition at line 33 of file PSpiceDigApiDefs.h.

5.3.2.12 typedef void(∗ pDeleteDevice_t) (void ∗pRef)

Parameters

in	*pRef*	Instance Reference

Definition at line 65 of file PSpiceDigApiDefs.h.

5.3.2.13 typedef const char∗(∗ pDigPrintDescription_t) (void ∗pRef)

Definition at line 104 of file PSpiceDigApiDefs.h.

5.3.2.14 typedef int(∗ pGetDeviceTermCount_t) (void ∗pRef)

Parameters

in	*Instance*	Reference

Returns

Number of terminals

Definition at line 71 of file PSpiceDigApiDefs.h.

5.3.2.15 typedef void(∗ pGetDeviceTermTypes_t) (void ∗pRef, void ∗pVectorPorts)

Parameters

in	*pRef*	Instance Reference
in	*pVectorPorts*	Vector of values

Definition at line 95 of file PSpiceDigApiDefs.h.

5.3.2.16 typedef int(∗ pGetDeviceTermValue_t) (void ∗pRef, PSpiceState ∗pVectorStates, int pSize)

Get Term values from Model Dll to PSpice.

Parameters

in	*pRef*	Instance Reference
in	*pVectorStates*	Vector of values

Definition at line 82 of file PSpiceDigApiDefs.h.

5.3.2.17 typedef double(∗ pGetTicksFromTime_t) (double pTime)

Returns

Ticks

Definition at line 109 of file PSpiceDigApiDefs.h.

5.3.2.18 typedef double(∗ pGetTimeFromTicks_t) (double pTicks)

Returns

Time

Definition at line 114 of file PSpiceDigApiDefs.h.

5.3.2.19 typedef void(∗ pInitDevice_t) (void ∗pRef)

Parameters

in	*pRef*	Instance Reference

Returns

None

Definition at line 39 of file PSpiceDigApiDefs.h.

5.3.2.20 typedef bool(∗ pPSpiceChanged_t) (void ∗pRef, const char ∗pNetName, double pTime, int pChanged)

Definition at line 120 of file PSpiceDigApiDefs.h.

5.3.2.21 typedef bool(∗ pPSpiceEvaluateDevice_t) (void ∗pRef, double pTicks, PSpiceState ∗pVectorStates, int pSize)

Parameters

in	*pRef*	Instance Reference [in] pTicks TICKS

Returns

Status false=>Failure true=>Success

Definition at line 46 of file PSpiceDigApiDefs.h.

5.3.2.22 typedef void(* pPSpiceGetInputSpec_t) (void *pRef, PSpiceInputSpec &pInSpec)

Get Input Buffer Specification enum value = PSPICE_GET_INSPEC.

Parameters

in	*pRef*	Model reference
in	*pInSpec*	Input specification

Returns

None

Definition at line 250 of file PSpiceDigApiDefs.h.

5.3.2.23 typedef void(* pPSpiceGetOutputSpec_t) (void *pRef, const PSpiceOutputSpec &pOutSpec)

Get Output Buffer Specification enum value = PSPICE_GET_OUTSPEC.

Parameters

in	*pRef*	Model reference
in	*pOutSpec*	Output specification

Returns

None

Definition at line 259 of file PSpiceDigApiDefs.h.

5.3.2.24 typedef void(* pPSpiceGetParameterValue_t) (void *pRef, PSpiceCMIParam *pParam)

Parameters

in	*pRef*	Instance Reference
in	*pParam*	PSpice Parameter

Definition at line 60 of file PSpiceDigApiDefs.h.

5.3.2.25 typedef bool(* pPSpiceGetTimingModelValue_t) (void *pRef, const char *pParamName, PSpiceDelay &pDelay)

Get Timing Model Value enum value = PSPICE_GET_TIMINGVALUE.

Parameters

in	*pRef*	Model reference
in	*pParamName*	Parameter Name
in	*pDelay*	Delay spec

Returns

None

Definition at line 269 of file PSpiceDigApiDefs.h.

5.3.2.26 typedef bool(∗ pPSpiceGetTransition_t) (void ∗pRef, const char ∗pOut, const char ∗pTransition)

Definition at line 124 of file PSpiceDigApiDefs.h.

5.3.2.27 typedef void(∗ pPSpiceSetConstraint_t) (void ∗pRef, const PSpiceConstraint ∗pConstraint)

PSpice API called by Model Dll.

Set Constraint enum value = PSPICE_SET_CONSTRAINT

Parameters

in	*pRef*	Model reference
in	*pDelay*	Delay pointer

Returns

None

Definition at line 220 of file PSpiceDigApiDefs.h.

5.3.2.28 typedef void(∗ pPSpiceSetDelay_t) (void ∗pRef, const PSpiceDelay ∗pDelay)

PSpice API called by Model Dll.

Set Delay - optional function - the same functionality is also available in setState enum value = PSPICE_SET_D↩ ELAY

Parameters

in	*pRef*	Model reference
in	*pDelay*	Delay pointer

Returns

None

Definition at line 210 of file PSpiceDigApiDefs.h.

5.3.2.29 typedef void(∗ pPSpiceSetInputSpec_t) (void ∗pRef, const PSpiceInputSpec ∗pInSpec)

Following functions should be called before evaluate - they make changes on all terminals of the device.

Setup Input Buffer specification enum value = PSPICE_SET_INSPEC

Parameters

in	*pRef*	Model reference
in	*pInSpec*	Input specification

Returns

None

Definition at line 232 of file PSpiceDigApiDefs.h.

5.3.2.30 typedef void(∗ pPSpiceSetOutputSpec_t) (void ∗pRef, PSpiceOutputSpec ∗pOutSpec)

Setup Output Buffer Specification enum value = PSPICE_SET_OUTSPEC.

Parameters

in	*pRef*	Model reference
in	*pOutSpec*	Output specification

Returns

None

Definition at line 241 of file PSpiceDigApiDefs.h.

5.3.2.31 typedef void(∗ pPSpiceSetState_t) (void ∗pRef, int pPortIndex, PSpiceState ∗pState, const PSpiceDelay ∗pDelay)

Parameters

in	*pRef*	Instance Reference
in	*pPortIndex*	0-based port index on which to set state
in	*pState*	New State
in	*pDelay*	Delay to be used before setting the new state

Definition at line 54 of file PSpiceDigApiDefs.h.

5.3.2.32 typedef bool(∗ pSetDeviceTermCount_t) (void ∗pRef, int pTermCount)

Returns

true => Check PASS, else FAIL

Definition at line 76 of file PSpiceDigApiDefs.h.

5.3.2.33 typedef int(∗ pSetDeviceTermValue_t) (void ∗pRef, PSpiceState ∗pVectorStates, int pSize)

Set Term values from PSpice to Model Dll.

Parameters

in	*pRef*	Instance Reference
in	*pVectorStates*	Vector of values

Definition at line 89 of file PSpiceDigApiDefs.h.

5.3.2.34 typedef bool(∗ pSetParameter_t) (void ∗pRef, int pParamCount, void ∗∗pVectorParameter)

Parameters

in	*pRef*	Instance Reference
in	*pParamCount*	Length of vector
in	*pVector↩ Parameter*	Vector of values

Definition at line 102 of file PSpiceDigApiDefs.h.

5.3.3 Enumeration Type Documentation

5.3.3.1 enum PSPICE_PORT_TYPE

Enumerator

PSPICE_PORT_IO

PSPICE_PORT_IN

Definition at line 9 of file PSpiceDigApiDefs.h.

5.3.4 Function Documentation

5.3.4.1 pspBit operator& (const pspBit & *pBit1,* const pspBit & *pBit2*) [inline]

Definition at line 605 of file PSpiceDigApiDefs.h.

5.3.4.2 bool operator== (const pspBit & *pBit1,* const pspBit & *pBit2*) [inline]

Definition at line 596 of file PSpiceDigApiDefs.h.

5.3.4.3 pspBit operator^ (const pspBit & *pBit1,* const pspBit & *pBit2*) [inline]

Definition at line 600 of file PSpiceDigApiDefs.h.

5.3.4.4 pspBit operator| (const pspBit & *pBit1,* const pspBit & *pBit2*) [inline]

Definition at line 610 of file PSpiceDigApiDefs.h.

5.3.4.5 pspBit operator~ (const pspBit & *pBit*) [inline]

Definition at line 615 of file PSpiceDigApiDefs.h.

5.3.4.6 PSP_CMI_EXPORT void PSpiceGetInputSpec (void * *pRef,* PSpiceInputSpec & *pInputSpec*)

5.3.4.7 PSP_CMI_EXPORT void PSpiceGetOutputSpec (void * *pRef,* PSpiceOutputSpec & *pOutputSpec*)

5.3.4.8 PSP_CMI_EXPORT void PSpiceGetParameterValue (void * *pRef,* PSpiceCMIParam * *pParam*)

5.3.4.9 PSP_CMI_EXPORT bool PSpiceGetTimingModelValue (void * *pRef,* const char * *pParamName,* PSpiceDelay & *pDelay*)

5.3.4.10 PSP_CMI_EXPORT void PSpiceSetConstraint (void * *pRef,* const PSpiceConstraint * *pConstraint*)

5.3.4.11 **PSP_CMI_EXPORT** void PSpiceSetDelay (void * *pRef,* const **PSpiceDelay** * *pDelay*)

5.3.4.12 **PSP_CMI_EXPORT** void PSpiceSetInputSpec (void * *pRef,* const **PSpiceInputSpec** * *pInSpec*)

5.3.4.13 **PSP_CMI_EXPORT** void PSpiceSetOutputSpec (void * *pRef,* const **PSpiceOutputSpec** * *pOutSpec*)

5.3.4.14 **PSP_CMI_EXPORT** void PSpiceSetState (void * *pRef,* int *pPortIndex,* **PSpiceState** * *pState,* const **PSpiceDelay** * *pDelay*)

Index

_filler
PSpiceState, 25
~PSpiceCMIParam
PSpiceCMIParam, 11
~PSpiceSetupHoldConstraint
PSpiceSetupHoldConstraint, 22

CDLL_FUNC
PSpiceCommonAPIDefs.h, 62
CKTag
PSpiceDeviceMiscInfo, 14
CKTomega
PSpiceDeviceMiscInfo, 14
clk_assertion
PSpiceSetupHoldConstraint, 22
code_latest/PSpiceCMIApiDefs.h, 29
code_latest/PSpiceCommonAPIDefs.h, 60
code_latest/PSpiceDigApiDefs.h, 72
CurrentAnalysisNumber
PSpiceDeviceMiscInfo, 14

Desc
PSpiceCMIParam, 11
descSetAC_Load_t
PSpiceCMIApiDefs.h, 33
descSetAddInternalNodes_t
PSpiceCMIApiDefs.h, 33
descSetBindTerminals_t
PSpiceCMIApiDefs.h, 34
descSetCheckModel_t
PSpiceCMIApiDefs.h, 34
descSetCheckPointSize_t
PSpiceCMIApiDefs.h, 34
descSetCheckTopology_t
PSpiceCMIApiDefs.h, 34
descSetCreateDevice_t
PSpiceDigApiDefs.h, 74
descSetDefaultInstance_t
PSpiceCMIApiDefs.h, 35
descSetDefaultModel_t
PSpiceCMIApiDefs.h, 35
descSetDefaultState_t
PSpiceCMIApiDefs.h, 35
descSetDeleteDevice_t
PSpiceDigApiDefs.h, 75
descSetDeleteInstance_t
PSpiceCMIApiDefs.h, 35
descSetDeleteModel_t
PSpiceCMIApiDefs.h, 36
descSetEvaluateDevice_t
PSpiceDigApiDefs.h, 75
descSetGetDeviceTermCount_t
PSpiceDigApiDefs.h, 75
descSetGetDeviceTermValue_t
PSpiceDigApiDefs.h, 75
descSetGetDeviceTerminals_t
PSpiceDigApiDefs.h, 75
descSetGetIntercept_t
PSpiceCMIApiDefs.h, 36
descSetGetMatrixPointers_t
PSpiceCMIApiDefs.h, 36
descSetInitDevice_t
PSpiceDigApiDefs.h, 76
descSetInstDataStructSize_t
PSpiceCMIApiDefs.h, 36
descSetInstallFunction1_t
PSpiceCommonAPIDefs.h, 62
descSetInstallFunction_t
PSpiceCommonAPIDefs.h, 64
descSetLoadCheckpoint_t
PSpiceCMIApiDefs.h, 37
descSetMaxTerminalCount_t
PSpiceCMIApiDefs.h, 37
descSetMinTerminalCount_t
PSpiceCMIApiDefs.h, 37
descSetModelDataStructSize_t
PSpiceCMIApiDefs.h, 38
descSetName_t
PSpiceCommonAPIDefs.h, 64
descSetNoise_t
PSpiceCMIApiDefs.h, 38
descSetPreload_t
PSpiceCMIApiDefs.h, 38
descSetReserveNodes_t
PSpiceCMIApiDefs.h, 38
descSetSaveCheckpoint_t
PSpiceCMIApiDefs.h, 39
descSetSaveTopology_t
PSpiceCMIApiDefs.h, 39
descSetSetDevicePinCurrent_t
PSpiceCMIApiDefs.h, 39
descSetSetDevicePinCurrentComplex_t
PSpiceCMIApiDefs.h, 39
descSetSetDeviceTermCount_t
PSpiceDigApiDefs.h, 76
descSetSetDeviceTermValue_t
PSpiceDigApiDefs.h, 76
descSetSetInstanceParams_t
PSpiceCMIApiDefs.h, 40

descSetSetModelParams_t
PSpiceCMIApiDefs.h, 40
descSetSetParameter_t
PSpiceDigApiDefs.h, 76
descSetSetTopologySize_t
PSpiceCMIApiDefs.h, 40
descSetSignalsStructSize_t
PSpiceCMIApiDefs.h, 41
descSetStateStructSize_t
PSpiceCMIApiDefs.h, 41
descSetStencilStructSize_t
PSpiceCMIApiDefs.h, 41
descSetTerminalNameCount_t
PSpiceCMIApiDefs.h, 42
descSetTerminalNames_t
PSpiceCMIApiDefs.h, 42
descSetTitle_t
PSpiceCMIApiDefs.h, 42
descSetTmpModDevice_t
PSpiceCMIApiDefs.h, 42
descSetTmpModModel_t
PSpiceCMIApiDefs.h, 43
descSetTranLoad_t
PSpiceCMIApiDefs.h, 43
descSetTrunc_t
PSpiceCMIApiDefs.h, 43
descSetVersion_t
PSpiceCommonAPIDefs.h, 64

errorflags
PSpiceFreqConstraint, 16
exp
PSpiceAnyScalar, 9

FALL
pspBit, 8
fields
PSpiceState, 25

GMin
PSpiceDeviceMiscInfo, 14
getLevel
PSpiceState, 24

h_drive
PSpiceOutputSpec, 19
HI
pspBit, 8
HIZ
pspBit, 8
hazardtype
PSpiceState, 25
holdtime_hi
PSpiceSetupHoldConstraint, 22
holdtime_lo
PSpiceSetupHoldConstraint, 22

ININIT
PSpiceCommonAPIDefs.h, 68
INNORM
PSpiceCommonAPIDefs.h, 68
INOFF
PSpiceCommonAPIDefs.h, 68
INPRDCT
PSpiceCommonAPIDefs.h, 68
INSTV0
PSpiceCommonAPIDefs.h, 68
INTRAN
PSpiceCommonAPIDefs.h, 68
inR
PSpiceInputSpec, 17
initFlags
PSpiceCommonAPIDefs.h, 68
isZ
PSpiceState, 24

l_drive
PSpiceOutputSpec, 19
LO
pspBit, 8
level
PSpiceState, 25
load
PSpiceInputSpec, 17
PSpiceOutputSpec, 19

MAXIOLEVEL
PSpiceDigApiDefs.h, 74
mClockName
PSpiceSetupHoldConstraint, 22
mCountData
PSpiceSetupHoldConstraint, 22
MDAC
PSpiceCommonAPIDefs.h, 69
MDBPDC
PSpiceCommonAPIDefs.h, 68
MDBPTR
PSpiceCommonAPIDefs.h, 68
MDDCSW
PSpiceCommonAPIDefs.h, 69
MDINITSMSIG
PSpiceCommonAPIDefs.h, 69
MDINITTRAN
PSpiceCommonAPIDefs.h, 69
MDTRAN
PSpiceCommonAPIDefs.h, 69
MDTRANOP
PSpiceCommonAPIDefs.h, 69
MDUIC
PSpiceCommonAPIDefs.h, 69
mData
PSpiceDeviceInst, 13
PSpiceDeviceModel, 15
mFreq
PSpiceConstraint, 12
mFreqSpecified
PSpiceFreqConstraint, 16
mInputNode

PSpiceFreqConstraint, 17
PSpiceWidthConstraint, 26
mInstID
PSpiceDeviceInst, 13
mMaxDelay
PSpiceDelay, 12
mMinDelay
PSpiceDelay, 12
mModelData
PSpiceDeviceInst, 13
mModelID
PSpiceDeviceModel, 15
mName
PSpicePort, 21
mNetName
PSpiceNetsList, 18
mNetsList
PSpiceSetupHoldConstraint, 23
mNext
PSpiceNetsList, 18
mNodeCount
PSpiceSignalNodeList, 23
mNodeNames
PSpiceSignalNodeList, 23
mNumber
PSpicePort, 21
mSetupHold
PSpiceConstraint, 12
mSetupHoldSpecified
PSpiceSetupHoldConstraint, 23
mSignals
PSpiceDeviceInst, 13
mState
PSpiceDeviceInst, 13
mStencil
PSpiceDeviceInst, 13
mTypDelay
PSpiceDelay, 12
mType
PSpicePort, 21
mVersion
PSpiceParamDesc, 20
mWidth
PSpiceConstraint, 12
mWidthSpecified
PSpiceWidthConstraint, 27
max_freq
PSpiceFreqConstraint, 16
MeasurementTemperature
PSpiceDeviceMiscInfo, 14
min_freq
PSpiceFreqConstraint, 16
min_high
PSpiceWidthConstraint, 26
min_low
PSpiceWidthConstraint, 26
modeFlags
PSpiceCommonAPIDefs.h, 68
msgid
PSpiceState, 25
multiple
PSpiceState, 25

Name
PSpiceParamDesc, 20
notposted
PSpiceState, 25

operator char
pspBit, 8
operator int
pspBit, 8
operator^
PSpiceDigApiDefs.h, 82
pspBit, 9
operator=
PSpiceState, 24
pspBit, 8
operator==
PSpiceDigApiDefs.h, 82
pspBit, 8
operator&
PSpiceDigApiDefs.h, 82
pspBit, 8
operator|
PSpiceDigApiDefs.h, 83
pspBit, 9
operator~
PSpiceDigApiDefs.h, 83
pspBit, 9

pAC_Load_t
PSpiceCMIApiDefs.h, 43
pAddInternalNodes_t
PSpiceCMIApiDefs.h, 44
pBindTerminals_t
PSpiceCMIApiDefs.h, 44
pCheckTopology_t
PSpiceCMIApiDefs.h, 44
pCreateDevice_t
PSpiceDigApiDefs.h, 76
pDefaultInstance_t
PSpiceCMIApiDefs.h, 45
pDefaultInstanceParams_t
PSpiceCommonAPIDefs.h, 64
pDefaultModel_t
PSpiceCMIApiDefs.h, 45
pDefaultModelParams_t
PSpiceCMIApiDefs.h, 45
pDefaultSignals_t
PSpiceCMIApiDefs.h, 45
pDefaultState_t
PSpiceCMIApiDefs.h, 46
pDefaultStencil_t
PSpiceCMIApiDefs.h, 46
pDeleteDevice_t
PSpiceDigApiDefs.h, 77

pDeleteInstance_t
PSpiceCMIApiDefs.h, 46
pDeleteModel_t
PSpiceCMIApiDefs.h, 46
pDeleteSignals_t
PSpiceCMIApiDefs.h, 47
pDeleteState_t
PSpiceCMIApiDefs.h, 47
pDeleteStencil_t
PSpiceCMIApiDefs.h, 47
pDigPrintDescription_t
PSpiceDigApiDefs.h, 77
pFnPtr1_t
PSpiceCommonAPIDefs.h, 64
pFnPtr_t
PSpiceCommonAPIDefs.h, 66
pGetBreakPoint_t
PSpiceCMIApiDefs.h, 47
pGetDeviceTermCount_t
PSpiceDigApiDefs.h, 77
pGetDeviceTermTypes_t
PSpiceDigApiDefs.h, 77
pGetDeviceTermValue_t
PSpiceDigApiDefs.h, 77
pGetIntercept_t
PSpiceCMIApiDefs.h, 48
pGetLastVoltage_t
PSpiceCMIApiDefs.h, 48
pGetMatrixPointers_t
PSpiceCMIApiDefs.h, 48
pGetPWLData_t
PSpiceCMIApiDefs.h, 48
pGetPWLDataStr_t
PSpiceCMIApiDefs.h, 48
pGetTicksFromTime_t
PSpiceDigApiDefs.h, 77
pGetTimeFromTicks_t
PSpiceDigApiDefs.h, 78
pInitDevice_t
PSpiceDigApiDefs.h, 78
pInstallFunction_t
PSpiceCMIApiDefs.h, 48
pIsPWLModel_t
PSpiceCMIApiDefs.h, 49
pLoadCheckpoint_t
PSpiceCMIApiDefs.h, 49
pModChk_t
PSpiceCMIApiDefs.h, 49
pNoise_t
PSpiceCMIApiDefs.h, 49
pPSPICEGetOptionsParams_t
PSpiceCommonAPIDefs.h, 66
pPSpiceAddInternalNode_t
PSpiceCMIApiDefs.h, 50
pPSpiceAddInternalNodeByName_t
PSpiceCMIApiDefs.h, 50
pPSpiceAdjustValueItem_t
PSpiceCMIApiDefs.h, 51
pPSpiceApplyValueItem_t
PSpiceCMIApiDefs.h, 51
pPSpiceApplyValueItemComplex_t
PSpiceCMIApiDefs.h, 51
pPSpiceChanged_t
PSpiceDigApiDefs.h, 78
pPSpiceCurrentTErr_t
PSpiceCMIApiDefs.h, 51
pPSpiceEvaluateDevice_t
PSpiceDigApiDefs.h, 78
pPSpiceGetCurrentAnalogTime_t
PSpiceCommonAPIDefs.h, 66
pPSpiceGetCurrentDigitalTime_t
PSpiceCommonAPIDefs.h, 66
pPSpiceGetCurrentStateIndex_t
PSpiceCMIApiDefs.h, 52
pPSpiceGetDelta_t
PSpiceCMIApiDefs.h, 52
pPSpiceGetDeltaPrevious_t
PSpiceCMIApiDefs.h, 52
pPSpiceGetDevice_t
PSpiceCommonAPIDefs.h, 66
pPSpiceGetFrequency_t
PSpiceCMIApiDefs.h, 52
pPSpiceGetInputSpec_t
PSpiceDigApiDefs.h, 78
pPSpiceGetLicenseString_t
PSpiceCommonAPIDefs.h, 66
pPSpiceGetMatrixPtr_t
PSpiceCMIApiDefs.h, 53
pPSpiceGetOutputSpec_t
PSpiceDigApiDefs.h, 78
pPSpiceGetParamValue_t
PSpiceCommonAPIDefs.h, 67
pPSpiceGetParamValueDbl_t
PSpiceCommonAPIDefs.h, 67
pPSpiceGetParameterValue_t
PSpiceDigApiDefs.h, 80
pPSpiceGetRHSPtr_t
PSpiceCMIApiDefs.h, 53
pPSpiceGetTimingModelValue_t
PSpiceDigApiDefs.h, 80
pPSpiceGetTransition_t
PSpiceDigApiDefs.h, 80
pPSpiceGetVoltageNodes_t
PSpiceCMIApiDefs.h, 53
pPSpiceGetVoltageNodesI_t
PSpiceCMIApiDefs.h, 53
pPSpiceIntegrate_t
PSpiceCMIApiDefs.h, 54
pPSpiceSetConstraint_t
PSpiceDigApiDefs.h, 80
pPSpiceSetDelay_t
PSpiceDigApiDefs.h, 80
pPSpiceSetInputSpec_t
PSpiceDigApiDefs.h, 81
pPSpiceSetOutputSpec_t
PSpiceDigApiDefs.h, 81

pPSpiceSetPWLDataDbl_t
PSpiceCMIApiDefs.h, 54
pPSpiceSetPWLDataStr_t
PSpiceCMIApiDefs.h, 54
pPSpiceSetProbeTitle_t
PSpiceCommonAPIDefs.h, 67
pPSpiceSetSimulationTemperature_t
PSpiceCommonAPIDefs.h, 67
pPSpiceSetState_t
PSpiceDigApiDefs.h, 81
pPSpiceUpdateStateVector_t
PSpiceCMIApiDefs.h, 55
pPSpiceVoltageTolerance_t
PSpiceCMIApiDefs.h, 55
pPSpiceWriteToOut_t
PSpiceCommonAPIDefs.h, 68
pPWLModelType_t
PSpiceCMIApiDefs.h, 55
pPreload_t
PSpiceCMIApiDefs.h, 50
pPrintDescription_t
PSpiceCMIApiDefs.h, 50
pReserveNodes_t
PSpiceCMIApiDefs.h, 55
PSP_CMI_EXPORT
PSpiceCommonAPIDefs.h, 62
PSP_VALUE_NOT_DEFINED
PSpiceDigApiDefs.h, 74
PSPICE_ADD_INTERNALNODE
PSpiceCommonAPIDefs.h, 69
PSPICE_ADD_INTERNALNODEBYNAME
PSpiceCommonAPIDefs.h, 71
PSPICE_ADJUST_VALUEITEM
PSpiceCommonAPIDefs.h, 71
PSPICE_APPLY_VALUEITEM
PSpiceCommonAPIDefs.h, 69
PSPICE_APPLY_VALUEITEMCMPLX
PSpiceCommonAPIDefs.h, 69
PSPICE_CURRENT_TERR
PSpiceCommonAPIDefs.h, 70
PSPICE_CURRENT_TOL
PSpiceCommonAPIDefs.h, 69
PSPICE_DIG_CREATEDEVICE
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_DELETEDEVICE
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_EVALDEVICE
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_FNCHANGED
PSpiceCommonAPIDefs.h, 71
PSPICE_DIG_FNTRANSITION
PSpiceCommonAPIDefs.h, 71
PSPICE_DIG_GETPARAMVALUE
PSpiceCommonAPIDefs.h, 71
PSPICE_DIG_GETTERM
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_GETTERMCOUNT
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_GETTERMTYPE
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_INITDEV
PSpiceCommonAPIDefs.h, 71
PSPICE_DIG_SETPARAM
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_SETPRINTDESC
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_SETSTATE
PSpiceCommonAPIDefs.h, 71
PSPICE_DIG_SETTERM
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_SETTERMCOUNT
PSpiceCommonAPIDefs.h, 70
PSPICE_DIG_TICK_FROM_TIME
PSpiceCommonAPIDefs.h, 71
PSPICE_DIG_TIME_FROM_TICK
PSpiceCommonAPIDefs.h, 71
PSPICE_GET_ACFREQUENCY
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_ANALOGTIME
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_DELTA
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_DELTAPREV
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_DIGITALTIME
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_INSPEC
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_LICENSE
PSpiceCommonAPIDefs.h, 71
PSPICE_GET_MATRIXPTR
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_OPTIONPARAMS
PSpiceCommonAPIDefs.h, 71
PSPICE_GET_OUTSPEC
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_PARAMVALUE
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_PARAMVALUEDBL
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_RHSPTR
PSpiceCommonAPIDefs.h, 71
PSPICE_GET_RHSPTR1
PSpiceCommonAPIDefs.h, 71
PSPICE_GET_STATE
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_TIMINGVALUE
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_V
PSpiceCommonAPIDefs.h, 69
PSPICE_GET_VI
PSpiceCommonAPIDefs.h, 69
PSPICE_GMINLOAD
PSpiceCommonAPIDefs.h, 69
PSPICE_INTEGRATE
PSpiceCommonAPIDefs.h, 69

PSPICE_INTERNAL0
PSpiceCommonAPIDefs.h, 71
PSPICE_PORT_IN
PSpiceDigApiDefs.h, 82
PSPICE_PORT_IO
PSpiceDigApiDefs.h, 82
PSPICE_PORT_TYPE
PSpiceDigApiDefs.h, 82
PSPICE_SET_ACLOAD
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_ADDINTERNALNODES
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_BINDTERMINALS
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_CHECKPTSIZE
PSpiceCommonAPIDefs.h, 71
PSPICE_SET_CHECKTOPOLOGY
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_CONSTRAINT
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_DEFAULTSTATE
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_DEFINST
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_DEFMODEL
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_DELAY
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_DELETEINST
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_DELETEMODEL
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_GETBREAKPOINT
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_GETINTERCEPT
PSpiceCommonAPIDefs.h, 71
PSPICE_SET_GETMATRIXPTR
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_GETTOPOLOGYSIZE
PSpiceCommonAPIDefs.h, 71
PSPICE_SET_INSPEC
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_INSTALLFUNC
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_INSTALLFUNC1
PSpiceCommonAPIDefs.h, 71
PSPICE_SET_INSTDATASTRUCTSIZE
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_INSTPARAMS
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_LOADCHECKPT
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_MODCHK
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_MODELDATASTRUCTSIZE
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_MODELPARAMDESC
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_MODELPARAMS
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_NAME
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_NOISE
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_OUTSPEC
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_PRELOAD
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_PRINTDESC
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_PROBETITLE
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_PWLDATA_DBL
PSpiceCommonAPIDefs.h, 71
PSPICE_SET_PWLDATA_STR
PSpiceCommonAPIDefs.h, 71
PSPICE_SET_RESERVENODES
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_SAVECHECKPT
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_SAVETOPOLOGY
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_SETPINI
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_SETPINICMPLX
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_SIGNALSTRUCTSIZE
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_SIMTEMP
PSpiceCommonAPIDefs.h, 69
PSPICE_SET_STATESTRUCTSIZE
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_STENCILSTRUCTSIZE
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_TMPMODDEVICE
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_TMPMODMODEL
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_TRANLOAD
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_TRUNC
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_VERSION
PSpiceCommonAPIDefs.h, 70
PSPICE_SET_WRITETOOUT
PSpiceCommonAPIDefs.h, 71
PSPICE_UPDATE_SV
PSpiceCommonAPIDefs.h, 70
PSPICE_VALUE_EXPRTYPE
PSpiceCommonAPIDefs.h, 71
PSPICE_VALUE_REALTYPE
PSpiceCommonAPIDefs.h, 71
PSPICE_VALUE_STRINGTYPE
PSpiceCommonAPIDefs.h, 71
PSPICE_VOLTAGE_TOL
PSpiceCommonAPIDefs.h, 69

pSaveCheckpoint_t
PSpiceCMIApiDefs.h, 55
pSaveTopology_t
PSpiceCMIApiDefs.h, 57
pSetDevicePinCurrent_t
PSpiceCMIApiDefs.h, 57
pSetDevicePinCurrentComplex_t
PSpiceCMIApiDefs.h, 57
pSetDeviceTermCount_t
PSpiceDigApiDefs.h, 82
pSetDeviceTermValue_t
PSpiceDigApiDefs.h, 82
pSetInstanceParams_t
PSpiceCommonAPIDefs.h, 68
pSetModelParams_t
PSpiceCMIApiDefs.h, 58
pSetParameter_t
PSpiceDigApiDefs.h, 82
pSetTopologySize_t
PSpiceCMIApiDefs.h, 58
PSpiceAPIs
PSpiceCommonAPIDefs.h, 69
PSpiceAnyScalar, 9
exp, 9
Real, 9
String, 9
PSpiceAnyValue, 10
Scalar, 10
Type, 10
PSpiceCMIApiDefs.h
descSetAC_Load_t, 33
descSetAddInternalNodes_t, 33
descSetBindTerminals_t, 34
descSetCheckModel_t, 34
descSetCheckPointSize_t, 34
descSetCheckTopology_t, 34
descSetDefaultInstance_t, 35
descSetDefaultModel_t, 35
descSetDefaultState_t, 35
descSetDeleteInstance_t, 35
descSetDeleteModel_t, 36
descSetGetIntercept_t, 36
descSetGetMatrixPointers_t, 36
descSetInstDataStructSize_t, 36
descSetLoadCheckpoint_t, 37
descSetMaxTerminalCount_t, 37
descSetMinTerminalCount_t, 37
descSetModelDataStructSize_t, 38
descSetNoise_t, 38
descSetPreload_t, 38
descSetReserveNodes_t, 38
descSetSaveCheckpoint_t, 39
descSetSaveTopology_t, 39
descSetSetDevicePinCurrent_t, 39
descSetSetDevicePinCurrentComplex_t, 39
descSetSetInstanceParams_t, 40
descSetSetModelParams_t, 40
descSetSetTopologySize_t, 40
descSetSignalsStructSize_t, 41
descSetStateStructSize_t, 41
descSetStencilStructSize_t, 41
descSetTerminalNameCount_t, 42
descSetTerminalNames_t, 42
descSetTitle_t, 42
descSetTmpModDevice_t, 42
descSetTmpModModel_t, 43
descSetTranLoad_t, 43
descSetTrunc_t, 43
pAC_Load_t, 43
pAddInternalNodes_t, 44
pBindTerminals_t, 44
pCheckTopology_t, 44
pDefaultInstance_t, 45
pDefaultModel_t, 45
pDefaultModelParams_t, 45
pDefaultSignals_t, 45
pDefaultState_t, 46
pDefaultStencil_t, 46
pDeleteInstance_t, 46
pDeleteModel_t, 46
pDeleteSignals_t, 47
pDeleteState_t, 47
pDeleteStencil_t, 47
pGetBreakPoint_t, 47
pGetIntercept_t, 48
pGetLastVoltage_t, 48
pGetMatrixPointers_t, 48
pGetPWLData_t, 48
pGetPWLDataStr_t, 48
pInstallFunction_t, 48
pIsPWLModel_t, 49
pLoadCheckpoint_t, 49
pModChk_t, 49
pNoise_t, 49
pPSpiceAddInternalNode_t, 50
pPSpiceAddInternalNodeByName_t, 50
pPSpiceAdjustValueItem_t, 51
pPSpiceApplyValueItem_t, 51
pPSpiceApplyValueItemComplex_t, 51
pPSpiceCurrentTErr_t, 51
pPSpiceGetCurrentStateIndex_t, 52
pPSpiceGetDelta_t, 52
pPSpiceGetDeltaPrevious_t, 52
pPSpiceGetFrequency_t, 52
pPSpiceGetMatrixPtr_t, 53
pPSpiceGetRHSPtr_t, 53
pPSpiceGetVoltageNodes_t, 53
pPSpiceGetVoltageNodesI_t, 53
pPSpiceIntegrate_t, 54
pPSpiceSetPWLDataDbl_t, 54
pPSpiceSetPWLDataStr_t, 54
pPSpiceUpdateStateVector_t, 55
pPSpiceVoltageTolerance_t, 55
pPWLModelType_t, 55
pPreload_t, 50
pPrintDescription_t, 50

pReserveNodes_t, 55
pSaveCheckpoint_t, 55
pSaveTopology_t, 57
pSetDevicePinCurrent_t, 57
pSetDevicePinCurrentComplex_t, 57
pSetModelParams_t, 58
pSetTopologySize_t, 58
pTmpModDevice_t, 58
pTmpModModel_t, 59
pTranLoad_t, 59
pTrunc_t, 59
PrimitivePtr, 55
PSpiceCMIParam, 10
∼PSpiceCMIParam, 11
Desc, 11
PSpiceCMIParam, 11
Value, 11
PSpiceCommonAPIDefs.h
CDLL_FUNC, 62
descSetInstallFunction1_t, 62
descSetInstallFunction_t, 64
descSetName_t, 64
descSetVersion_t, 64
ININIT, 68
INNORM, 68
INOFF, 68
INPRDCT, 68
INSTV0, 68
INTRAN, 68
initFlags, 68
MDAC, 69
MDBPDC, 68
MDBPTR, 68
MDDCSW, 69
MDINITSMSIG, 69
MDINITTRAN, 69
MDTRAN, 69
MDTRANOP, 69
MDUIC, 69
modeFlags, 68
pDefaultInstanceParams_t, 64
pFnPtr1_t, 64
pFnPtr_t, 66
pPSPICEGetOptionsParams_t, 66
pPSpiceGetCurrentAnalogTime_t, 66
pPSpiceGetCurrentDigitalTime_t, 66
pPSpiceGetDevice_t, 66
pPSpiceGetLicenseString_t, 66
pPSpiceGetParamValue_t, 67
pPSpiceGetParamValueDbl_t, 67
pPSpiceSetProbeTitle_t, 67
pPSpiceSetSimulationTemperature_t, 67
pPSpiceWriteToOut_t, 68
PSP_CMI_EXPORT, 62
PSPICE_ADD_INTERNALNODE, 69
PSPICE_ADD_INTERNALNODEBYNAME, 71
PSPICE_ADJUST_VALUEITEM, 71
PSPICE_APPLY_VALUEITEM, 69
PSPICE_APPLY_VALUEITEMCMPLX, 69
PSPICE_CURRENT_TERR, 70
PSPICE_CURRENT_TOL, 69
PSPICE_DIG_CREATEDEVICE, 70
PSPICE_DIG_DELETEDEVICE, 70
PSPICE_DIG_EVALDEVICE, 70
PSPICE_DIG_FNCHANGED, 71
PSPICE_DIG_FNTRANSITION, 71
PSPICE_DIG_GETPARAMVALUE, 71
PSPICE_DIG_GETTERM, 70
PSPICE_DIG_GETTERMCOUNT, 70
PSPICE_DIG_GETTERMTYPE, 70
PSPICE_DIG_INITDEV, 71
PSPICE_DIG_SETPARAM, 70
PSPICE_DIG_SETPRINTDESC, 70
PSPICE_DIG_SETSTATE, 71
PSPICE_DIG_SETTERM, 70
PSPICE_DIG_SETTERMCOUNT, 70
PSPICE_DIG_TICK_FROM_TIME, 71
PSPICE_DIG_TIME_FROM_TICK, 71
PSPICE_GET_ACFREQUENCY, 69
PSPICE_GET_ANALOGTIME, 69
PSPICE_GET_DELTA, 69
PSPICE_GET_DELTAPREV, 69
PSPICE_GET_DIGITALTIME, 69
PSPICE_GET_INSPEC, 69
PSPICE_GET_LICENSE, 71
PSPICE_GET_MATRIXPTR, 69
PSPICE_GET_OPTIONPARAMS, 71
PSPICE_GET_OUTSPEC, 69
PSPICE_GET_PARAMVALUE, 69
PSPICE_GET_PARAMVALUEDBL, 69
PSPICE_GET_RHSPTR, 71
PSPICE_GET_RHSPTR1, 71
PSPICE_GET_STATE, 69
PSPICE_GET_TIMINGVALUE, 69
PSPICE_GET_V, 69
PSPICE_GET_VI, 69
PSPICE_GMINLOAD, 69
PSPICE_INTEGRATE, 69
PSPICE_INTERNAL0, 71
PSPICE_SET_ACLOAD, 70
PSPICE_SET_ADDINTERNALNODES, 70
PSPICE_SET_BINDTERMINALS, 70
PSPICE_SET_CHECKPTSIZE, 71
PSPICE_SET_CHECKTOPOLOGY, 70
PSPICE_SET_CONSTRAINT, 69
PSPICE_SET_DEFAULTSTATE, 70
PSPICE_SET_DEFINST, 70
PSPICE_SET_DEFMODEL, 70
PSPICE_SET_DELAY, 69
PSPICE_SET_DELETEINST, 70
PSPICE_SET_DELETEMODEL, 70
PSPICE_SET_GETBREAKPOINT, 70
PSPICE_SET_GETINTERCEPT, 71
PSPICE_SET_GETMATRIXPTR, 70
PSPICE_SET_GETTOPOLOGYSIZE, 71
PSPICE_SET_INSPEC, 69

PSPICE_SET_INSTALLFUNC, 69
PSPICE_SET_INSTALLFUNC1, 71
PSPICE_SET_INSTDATASTRUCTSIZE, 69
PSPICE_SET_INSTPARAMS, 70
PSPICE_SET_LOADCHECKPT, 70
PSPICE_SET_MODCHK, 70
PSPICE_SET_MODELDATASTRUCTSIZE, 69
PSPICE_SET_MODELPARAMDESC, 70
PSPICE_SET_MODELPARAMS, 70
PSPICE_SET_NAME, 70
PSPICE_SET_NOISE, 70
PSPICE_SET_OUTSPEC, 69
PSPICE_SET_PRELOAD, 70
PSPICE_SET_PRINTDESC, 70
PSPICE_SET_PROBETITLE, 69
PSPICE_SET_PWLDATA_DBL, 71
PSPICE_SET_PWLDATA_STR, 71
PSPICE_SET_RESERVENODES, 70
PSPICE_SET_SAVECHECKPT, 70
PSPICE_SET_SAVETOPOLOGY, 70
PSPICE_SET_SETPINI, 70
PSPICE_SET_SETPINICMPLX, 70
PSPICE_SET_SIGNALSTRUCTSIZE, 70
PSPICE_SET_SIMTEMP, 69
PSPICE_SET_STATESTRUCTSIZE, 70
PSPICE_SET_STENCILSTRUCTSIZE, 70
PSPICE_SET_TMPMODDEVICE, 70
PSPICE_SET_TMPMODMODEL, 70
PSPICE_SET_TRANLOAD, 70
PSPICE_SET_TRUNC, 70
PSPICE_SET_VERSION, 70
PSPICE_SET_WRITETOOUT, 71
PSPICE_UPDATE_SV, 70
PSPICE_VALUE_EXPRTYPE, 71
PSPICE_VALUE_REALTYPE, 71
PSPICE_VALUE_STRINGTYPE, 71
PSPICE_VOLTAGE_TOL, 69
pSetInstanceParams_t, 68
PSpiceAPIs, 69
PSpiceInstallFunction, 71
PSpiceInstallFunction1, 72
PSpiceValueType, 71
pspiceSetFunctionList, 72
PSpiceConstraint, 11
mFreq, 12
mSetupHold, 12
mWidth, 12
PSpiceDelay, 12
mMaxDelay, 12
mMinDelay, 12
mTypDelay, 12
PSpiceDeviceInst, 13
mData, 13
mInstID, 13
mModelData, 13
mSignals, 13
mState, 13
mStencil, 13
PSpiceDeviceMiscInfo, 14
CKTag, 14
CKTomega, 14
CurrentAnalysisNumber, 14
GMin, 14
MeasurementTemperature, 14
RelTol, 14
SkipBP, 15
Temperature, 15
Vt, 15
PSpiceDeviceModel, 15
mData, 15
mModelID, 15
PSpiceDigApiDefs.h
descSetCreateDevice_t, 74
descSetDeleteDevice_t, 75
descSetEvaluateDevice_t, 75
descSetGetDeviceTermCount_t, 75
descSetGetDeviceTermValue_t, 75
descSetGetDeviceTerminals_t, 75
descSetInitDevice_t, 76
descSetSetDeviceTermCount_t, 76
descSetSetDeviceTermValue_t, 76
descSetSetParameter_t, 76
MAXIOLEVEL, 74
operator^, 82
operator==, 82
operator&, 82
operator|, 83
operator~, 83
pCreateDevice_t, 76
pDeleteDevice_t, 77
pDigPrintDescription_t, 77
pGetDeviceTermCount_t, 77
pGetDeviceTermTypes_t, 77
pGetDeviceTermValue_t, 77
pGetTicksFromTime_t, 77
pGetTimeFromTicks_t, 78
pInitDevice_t, 78
pPSpiceChanged_t, 78
pPSpiceEvaluateDevice_t, 78
pPSpiceGetInputSpec_t, 78
pPSpiceGetOutputSpec_t, 78
pPSpiceGetParameterValue_t, 80
pPSpiceGetTimingModelValue_t, 80
pPSpiceGetTransition_t, 80
pPSpiceSetConstraint_t, 80
pPSpiceSetDelay_t, 80
pPSpiceSetInputSpec_t, 81
pPSpiceSetOutputSpec_t, 81
pPSpiceSetState_t, 81
PSP_VALUE_NOT_DEFINED, 74
PSPICE_PORT_IN, 82
PSPICE_PORT_IO, 82
PSPICE_PORT_TYPE, 82
pSetDeviceTermCount_t, 82
pSetDeviceTermValue_t, 82
pSetParameter_t, 82

PSpiceGetInputSpec, 83
PSpiceGetOutputSpec, 83
PSpiceGetParameterValue, 83
PSpiceGetTimingModelValue, 83
PSpiceSetConstraint, 83
PSpiceSetDelay, 83
PSpiceSetInputSpec, 83
PSpiceSetOutputSpec, 83
PSpiceSetState, 83
UNSPEC, 74
PSpiceFreqConstraint, 16
errorflags, 16
mFreqSpecified, 16
mInputNode, 17
max_freq, 16
min_freq, 16
PSpiceFreqConstraint, 16
PSpiceGetInputSpec
PSpiceDigApiDefs.h, 83
PSpiceGetOutputSpec
PSpiceDigApiDefs.h, 83
PSpiceGetParameterValue
PSpiceDigApiDefs.h, 83
PSpiceGetTimingModelValue
PSpiceDigApiDefs.h, 83
PSpiceInputSpec, 17
inR, 17
load, 17
Tstore, 17
PSpiceInstallFunction
PSpiceCommonAPIDefs.h, 71
PSpiceInstallFunction1
PSpiceCommonAPIDefs.h, 72
PSpiceNetsList, 18
mNetName, 18
mNext, 18
PSpiceNetsList, 18
PSpiceOutputSpec, 18
h_drive, 19
l_drive, 19
load, 19
PSpiceOutputSpec, 19
pwrt, 19
tswhl, 20
tswlh, 20
z_drive, 20
PSpiceParamDesc, 20
mVersion, 20
Name, 20
PSpicePort, 21
mName, 21
mNumber, 21
mType, 21
PSpiceSetConstraint
PSpiceDigApiDefs.h, 83
PSpiceSetDelay
PSpiceDigApiDefs.h, 83
PSpiceSetInputSpec
PSpiceDigApiDefs.h, 83
PSpiceSetOutputSpec
PSpiceDigApiDefs.h, 83
PSpiceSetState
PSpiceDigApiDefs.h, 83
PSpiceSetupHoldConstraint, 21
∼PSpiceSetupHoldConstraint, 22
clk_assertion, 22
holdtime_hi, 22
holdtime_lo, 22
mClockName, 22
mCountData, 22
mNetsList, 23
mSetupHoldSpecified, 23
PSpiceSetupHoldConstraint, 22
releasetime_hl, 23
releasetime_lh, 23
setuptime_hi, 23
setuptime_lo, 23
PSpiceSignalNodeList, 23
mNodeCount, 23
mNodeNames, 23
PSpiceState, 24
_filler, 25
fields, 25
getLevel, 24
hazardtype, 25
isZ, 24
level, 25
msgid, 25
multiple, 25
notposted, 25
operator=, 24
persistent, 25
stateVal, 25
str0, 25
str1, 25
val, 25
PSpiceValueType
PSpiceCommonAPIDefs.h, 71
PSpiceWidthConstraint, 26
mInputNode, 26
mWidthSpecified, 27
min_high, 26
min_low, 26
PSpiceWidthConstraint, 26
pTmpModDevice_t
PSpiceCMIApiDefs.h, 58
pTmpModModel_t
PSpiceCMIApiDefs.h, 59
pTranLoad_t
PSpiceCMIApiDefs.h, 59
pTrunc_t
PSpiceCMIApiDefs.h, 59
persistent
PSpiceState, 25
PrimitivePtr
PSpiceCMIApiDefs.h, 55

pspBit, 7
    FALL, 8
    HI, 8
    HIZ, 8
    LO, 8
    operator char, 8
    operator int, 8
    operator^, 9
    operator=, 8
    operator==, 8
    operator&, 8
    operator|, 9
    operator~, 9
    pspBit, 8
    pspBitLevels, 7
    RISE, 8
    UNKNOWN, 8
pspBitLevels
    pspBit, 7
pspiceSetFunctionList
    PSpiceCommonAPIDefs.h, 72
pwrt
    PSpiceOutputSpec, 19

RISE
    pspBit, 8
Real
    PSpiceAnyScalar, 9
RelTol
    PSpiceDeviceMiscInfo, 14
releasetime_hl
    PSpiceSetupHoldConstraint, 23
releasetime_lh
    PSpiceSetupHoldConstraint, 23

Scalar
    PSpiceAnyValue, 10
setuptime_hi
    PSpiceSetupHoldConstraint, 23
setuptime_lo
    PSpiceSetupHoldConstraint, 23
SkipBP
    PSpiceDeviceMiscInfo, 15
stateVal
    PSpiceState, 25
str0
    PSpiceState, 25
str1
    PSpiceState, 25
String
    PSpiceAnyScalar, 9

Temperature
    PSpiceDeviceMiscInfo, 15
Tstore
    PSpiceInputSpec, 17
tswhl
    PSpiceOutputSpec, 20
tswlh
    PSpiceOutputSpec, 20
Type
    PSpiceAnyValue, 10

UNKNOWN
    pspBit, 8
UNSPEC
    PSpiceDigApiDefs.h, 74

val
    PSpiceState, 25
Value
    PSpiceCMIParam, 11
Vt
    PSpiceDeviceMiscInfo, 15

z_drive
    PSpiceOutputSpec, 20