cadence

PSpice User's Guide, Product Version 16.3

15

Mixed analog/digital simulation

Chapter overview

This chapter describes how PSpice A/D runs mixed analog/digital simulations and includes the following sections:

- Interconnecting analog and digital parts
- Interface subcircuit selection by PSpice
- Specifying digital power supplies
- Interface generation and node names

Note: This entire chapter describes features that are not included in PSpice.

Interconnecting analog and digital parts

Prior to simulation, netlisting translates the part instances and nets defined in your schematic into parts connected by nodes. The standard simulation netlist contains a flat view of the circuit. You can also create hierarchical netlists. PSpice A/D extracts the definitions for all parts modeled as subcircuits, viewing parts as a collection of primitive parts and node connections.

The digital primitives that make up a digital part determine the way that PSpice A/D processes an analog/digital interface to that part. Specifically, the I/O model for each digital primitive connected at the interface gives PSpice A/D the necessary information.

PSpice A/D recognizes three types of nodes: analog nodes, digital nodes, and interface nodes. The

node type is determined by the types of parts connected to it. If all of the parts connected to a node are analog, then it is an analog node. If all of the parts are digital, then it is a digital node. If there is a combination of analog and digital parts, then it is an interface node.

PSpice A/D automatically breaks interface nodes into one purely analog and one or more digital nodes by inserting one or more analog/digital interface subcircuits.

PSpice A/D also automatically connects a power supply to the interface subcircuit to complete the generation of the interface.

To view simulation results at an analog/digital interface in your schematic using the graphical waveform analyzer:

- Place a marker on the appropriate interface net. The additional nodes created by PSpice A/D remain transparent.
- View results in PSpice A/D by selecting traces from the output variable list (from the Trace menu, choose Add Trace). If you use this approach, note the names PSpice A/D generates for the new nodes. To find out more, see Interface generation and node names.

Interface subcircuit selection by PSpice

Analog-to-digital (AtoD) and digital-to-analog (DtoA) interface subcircuits handle the translation between analog voltages/impedances and digital states, or vice-versa. The main component of an interface subcircuit is either a PSpice N part (digital input: digital-to-analog) or a PSpice O (that's the letter O, not the numeral zero) part (digital output: analog-to-digital).

PSpice N and O parts are neatly packaged into interface subcircuits in the model library. The standard model library shipped with your software installation includes interface subcircuits for each of the supported logic families: TTL, CD4000 series CMOS and high-speed CMOS (HC/HCT), ECL 10K, and ECL 100K. This frees you from ever having to define them yourself when using parts in the standard library. If you are creating custom digital parts in technologies other than those provided in the standard model library, you may need to create your own interface subcircuits.

Note: To search for particular parts in the standard PSpice libraries, see the online *PSpice Library List*.

Every digital primitive comprising the subcircuit description of a digital part has an I/O model describing its loading and driving characteristics. The name of the interface subcircuit actually inserted by PSpice A/D is specified by the I/O model of the digital primitive at the interface. The I/O model has parameters for up to four analog-to-digital (AtoD) and four digital-to-analog (DtoA) subcircuit names.

You can choose among four interface levels of subcircuit models, depending on the simulation accuracy you need. In some cases you may need more accurate simulations of the input/output stages of a digital part, while in other cases, a simpler, smaller model is enough.

Digital parts provided in the standard libraries only use interface levels 1 and 2. With the exception of the HC/HCT series (described below), levels 3 and 4 reference the same subcircuits as levels 1 and 2.

Table below summarizes the four interface levels.

The difference between levels 1 and 2 only occurs in the AtoD interfaces, described below. In all cases, the level 1 DtoA interface is the same as the level 2 DtoA interface, except that the level 2 DtoA interface does not generate intermediate R, F, and X levels.

Table 15-1 Interface subcircuit models

Level	Subcircuits	Definition
1	AtoD1/DtoA1	AtoD generates intermediate R, F, and X levels
2	AtoD2/DtoA2	AtoD does not generate intermediate R, F, and X levels
3	AtoD3/DtoA3	(same as level 1)
4	AtoD4/DtoA4	(same as level 2)

The OrCAD libraries provide two different DtoA models in the HC/HCT series: the *simple* model and the *elaborate* model. You can use the simple model by specifying level 1 or 2, the elaborate model by specifying level 3 or 4. The elaborate model is noticeably slower than the simple model, so you should only use it if you are using a power supply level other than 5.0 volts.

The HC/HCT level 1 and 2 DtoA models produce accurate I-V curves given a fixed power supply of 5.0 volts and a temperature of 25°C. The level 3 and 4 DtoA models produce accurate I-V curves over the acceptable range of power supply voltages (2-6 volts), and they include temperature derating.

Level 1 interface

The level 1 AtoD interface generates intermediate logic levels (R, F, X) between the voltage ranges VILMAX and VIHMIN (specific voltages depend on the technology you are using). A steadily rising voltage on the input of the AtoD will transition from 0 to R at VILMAX and from R to 1 at VIHMIN. The F level is output for steadily falling voltages in a similar manner. The X level is produced if the input voltage starts in the threshold region or doubles back into a previously crossed threshold.

Level 1 (the default) strictly maps logic levels onto the changing input voltage. The exact switching voltage is assumed to be anywhere between VILMAX and VIHMIN due to temperature or power supply variations. Thus, it provides more accurate, less optimistic results.

This behavior may not be appropriate when the input rise and fall times are long, or when the input voltage never leaves the threshold region. If this is the case, you may want to use the level 2 interface.

Level 2 interface

The level 2 AtoD interface transitions directly from 0 to 1 and 1 to 0 without passing through

intermediate R, F, or X levels. An exact switching voltage is assumed (again, the specific voltage depends on the technology you are using). It provides a more optimistic, and therefore less accurate, response than level 1. Level 2's behavior is appropriate when the input voltage oscillates around the threshold voltage.

Note: You can avoid simulations that get bogged down with the greater detail of R, F, and X states around these oscillations. You may want to specify level 2 on only those parts for which this behavior is critical to a successful simulation. This is described in Setting the default A/D interface below.

Setting the default A/D interface

For mixed-signal simulation, you can select the AtoD and DtoA interface level circuit-wide and on individual part instances.

- To select the default interface level circuit-wide, select one of the four Default A/D interfaces in the Simulation Settings dialog box by selecting the Gate-level simulation category under the Options tab. Part instances with the IO_LEVEL property set to 0 use this value.
- You can override the circuit-wide default on an individual part by specifying an IO_LEVEL property from 1 to 4, where:

1: AtoD1 and DtoA1 (default)

2: AtoD2 and DtoA2

3: AtoD3 and DtoA3

4: AtoD4 and DtoA4

For example, you can tell the simulator to use the level 2 interface subcircuits for a 7400 part by setting the IO_LEVEL property to 2. All other part instances continue to use the circuit-wide setting. By default, IO_LEVEL is set to 0, which tells the simulator to use the circuit-wide level defined in the Gate-level simulation category in the Simulation Settings dialog box.

Specifying digital power supplies

Digital power supplies are used to power interface subcircuits that are automatically created by PSpice A/D when simulating analog/digital interfaces. They are specified as follows:

- PSpice A/D can instantiate them automatically.
- You can create your own digital power supplies and place them in your design.

When using parts from the standard libraries in your design, you can usually have PSpice A/D

automatically create the necessary digital power supply. If you use custom digital parts created in technologies other than those provided in the standard model library, you may need to create your own digital power supplies.

Because digital power supplies are used only by analog/digital interface subcircuits, digital power supplies are not needed for digital-only designs. We recommend avoiding placing a power supply to a digital-only design because it may increase simulation time and memory usage.

Default power supply selection by PSpice A/D

When PSpice A/D encounters an analog/digital interface, it creates the appropriate interface subcircuit and power supply according to the I/O model referenced by the digital part. The I/O model is specific to the digital part's logic family. The power supply provides reference or drive voltage for the analog side of the interface.

By default, PSpice A/D inserts one power supply subcircuit for every logic family in which a digital primitive is involved with an analog/digital interface. These power supply subcircuits create the digital power and ground nodes that are the defaults for all parts in that family. If multiple digital primitives from the same logic family are involved with analog/digital interfaces, one instance of the power supply subcircuit is created with all primitives connected to the power supply nodes.

Table 15-2 summarizes the default node names and values. For instance, TTL power supplies have a default value of 5.0 volts at analog/digital interfaces.

Table 15-2 Default digital power/ground pin connections

Logic family	Digital power/ ground pin properties	Default digital power/ground nodes
TTL	PSPICEDEFAULTNET (PWR) PSPICEDEFAULTNET (GND)	$G_DPWR (5.0 volts) $G_DGND (0 volts)
CD4000	PSPICEDEFAULTNET (VDD) PSPICEDEFAULTNET (VSS)	$G_CD4000_VDD (5 volts) $G_CD4000_VSS (0 volts)
ECL 10K	PSPICEDEFAULTNET (VEE) PSPICEDEFAULTNET (VCC1) PSPICEDEFAULTNET (VCC2)	$G_ECL_10K_VEE (-5.2 volts) $G_ECL_10K_VCC1 (0 volts) $G_ECL_10K_VCC2 (0 volts)
ECL 100K	PSPICEDEFAULTNET (VEE) PSPICEDEFAULTNET (VCC1) PSPICEDEFAULTNET (VCC2)	$G_ECL_100K_VEE (-4.5 volts) $G_ECL_100K_VCC1 (0 volts) $G_ECL_100K_VCC2 (0 volts)

The PSPICEDEFAULTNET pin properties have the same default values as the digital power and ground nodes created by the default power supply. These node assignments are passed from the part instance to the digital primitives describing its behavior, connecting any digital primitive affected by an analog connection to the correct power supply.

The default I/O models and power supply subcircuits are found in DIG_IO.LIB. The four default power supplies provided in the model library are DIGIFPWR (TTL), CD4000_PWR (CD4000 series CMOS), ECL_10K_PWR (ECL 10K), and ECL_100K_PWR (ECL 100K).

Creating custom digital power supplies

Each digital part model has optional digital power and ground nodes that you can use to specify custom power supplies. To do this, use one of the digital power supplies listed in Table 15-3 below in your design and redefine the digital power supply nodes.

Table 15-3 Digital power supply parts in SPECIAL.OLB

Part type (PSpice A/D X model)	Part name
CD4000 power supply	CD4000_PWR
TTL power supply	DIGIFPWR
ECL 10K power supply	ECL_10K_PWR
ECL 100K power supply	ECL_100K_PWR

When creating custom power supplies, you can refer to the power supply definitions in DIG_IO.LIB for examples of power supply subcircuit definitions. The properties relevant to creating custom power supplies are shown in Table 15-4.

Table 15-4 Digital power supply properties

Part name	Property	Description
CD4000_PWR	VOLTAGE	CD4000 series CMOS power supply voltage
	PSPICEDEFAULTNET	CD4000 series CMOS hidden power supply pins for VDD and VSS
DIGIFPWR	VOLTAGE	TTL power supply voltage
	PSPICEDEFAULTNET	TTL hidden power (PWR) and ground (GND) pins

ECL_10K_PWR ECL_100K_PWR	VEE VCC1 VCC2	ECL power supply voltages
	PSPICEDEFAULTNET	ECL hidden power supply pins for VEE, VCC1 and VCC2

To create a custom digital power supply

Note: This procedure applies to all logic families.

- Place the appropriate power supply part listed in Table 15-3 in your design (by logic family).
- Rename the power supply power and ground pins (PSPICEDEFAULTNET properties).
- Reset the power supply power and ground voltages as required.
- For any digital part instance that uses the power supply, set its appropriate PSPICEDEFAULTNET pin properties to the power and ground pins created by the secondary power supply.

Overriding CD4000 power supply voltage throughout a design

Designs using CD4000 parts often require power supply voltages other than the default 5.0 volts supplied by the standard CD4000_PWR power supply part. If needed, you can override the power supply voltage for all CD4000 parts in a design.

The default power supply nodes used by CD4000 parts are named $G_CD4000_VDD and $G_CD4000_VSS as created by the power supply subcircuit CD4000_PWR. This supply defaults to 5.0 volts. You can override the voltage across these two nodes by defining values for the parameters named CD4000_VDD and CD4000_VSS that are referenced by the CD4000_PWR subcircuit definition.

To change the CD4000_PWR power supply to 12 volts, referenced to ground:

- Place an instance of the PARAM pseudo-part from `SPECIAL.OLB`.
- Create a new PARAM property as follows:

```
CD4000_VDD = 12.0V
```

DC4000_VSS is left at its default of 0 volts.

If the reference voltage also needs to be reset, the same method can be used to define the CD4000_VSS parameter by setting this property of the same PARAM instance. For example, if you want the supplies to go between -5 volts and +5 volts (a difference of 10 volts), set CD4000_VSS to -5V and CD4000_VDD to +10V; as a result, CD4000_VDD is 10 volts above CD4000_VSS, or +5 volts.

Creating a secondary CD4000, TTL, or ECL power supply

Designs using CD4000, TTL, or ECL parts may require power supply voltages in addition to the default 5.0 volts supplied by the standard CD4000_PWR power supply part.

To create a secondary power supply for any one of the CD4000, TTL, or ECL technologies, you must

place the appropriate power supply part and create user-defined nodes with a new voltage value.

Note: Designs with TTL and ECL parts rarely require secondary power supplies. If needed, however, you can use this procedure to add a secondary power supply for TTL and ECL parts.

To create and use a secondary CD4000 power supply with nodes MY_VDD and MY_VSS and a voltage of 3.5 volts:

Place the CD4000_PWR power supply and modify the appropriate pin properties as follows:

```
VOLTAGE = 3.5V
PSPICEDEFAULTNET = MY_VDD
PSPICEDEFAULTNET = MY_VSS
```

Select a CD4000 part in the schematic to which the new power supply should apply, then change the appropriate pin properties as follows:

```
PSPICEDEFAULTNET = MY_VDD
PSPICEDEFAULTNET = MY_VSS
```

Interface generation and node names

The majority of the interface generation process involves PSpice A/D determining whether analog and digital primitives are connected, and if so, inserting an interface subcircuit for each digital connection. This turns the interface node into a purely analog node, which now connects to the analog terminal of the interface subcircuit. To complete the original connection, PSpice A/D creates a new digital node between the digital terminal of the interface subcircuit and the digital primitive.

Because PSpice A/D must create new digital nodes, it must give them unique names. These node names are used in the output variables in the list of viewable traces when you choose Add Trace from the Trace menu. Name generation follows these rules:

The analog node retains the name of the original interface node--either the labeled wire name in the design, or the node name automatically generated for an unlabeled wire.

Each new digital node name consists of the labeled wire name in the design or the node name automatically generated for an unlabeled wire, appended with $AtoD or $DtoA. If the node is attached to more than one digital part, the second digital node is appended with $AtoD2 or $DtoA2, and so on.

Figure 15-1 below shows a fragment of a mixed analog/digital circuit before and after the interface subcircuits have been added. The wires labeled 1 and 2 in the schematic representation are the interface nets connecting analog and digital parts. These translate to interface nodes, which are processed by PSpice A/D to create the circuit fragment shown in the PSpice A/D representation.

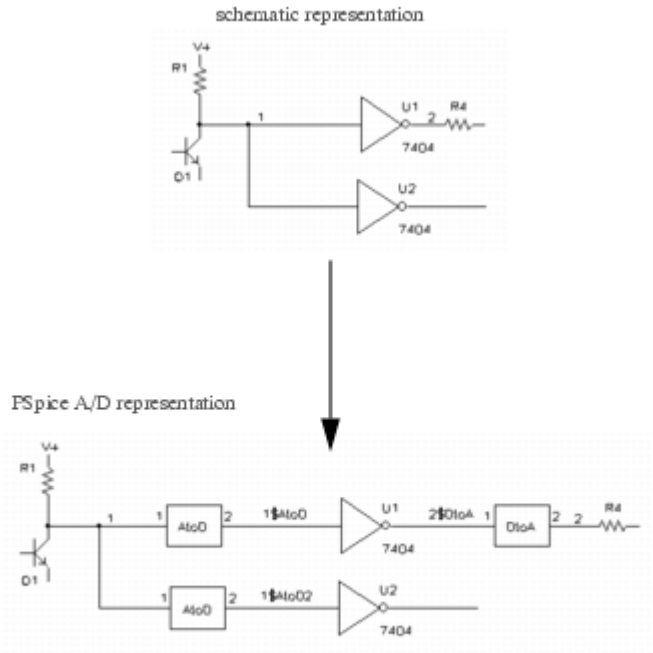

Figure 15-1 Mixed analog/digital circuit before and after interface generation.

After interface generation, node 1 is a purely analog node, connecting the resistor, transistor, and the analog inputs of both AtoD subcircuits. Node 2 is also a purely analog node, connecting the resistor and the analog output of the DtoA interface. You can see that PSpice A/D inserted two new digital nodes, 1$AtoD and 1$AtoD2, which connect the outputs of the AtoD interfaces to the inverter inputs. It also created one digital node, 2$DtoA, to connect the output of U1 to the digital input of the DtoA interface.

The interface subcircuits PSpice A/D automatically generates are listed in the simulation output file under the section named Generated AtoD and DtoA Interfaces. For the example in Figure 15-1, this section would appear in the simulation output file as shown in Figure 15-2 below.

```
**** Generated AtoD and DtoA Interfaces ****
*
* Analog/Digital interface for node 1
*
* Moving X1.U1:.A from analog node 1 to new digital node * 1$AtoD
X$1_AtoD1 1 1$AtoD $G_DPWR $G_DGND AtoD_STD
+     PARAMS: CAPACITANCE= 0
* Moving X2.U1:.A from analog node 1 to new digital node * 1$AtoD2
X$1_AtoD2 1 1$AtoD $G_DPWR $G_DGND AtoD_STD
+     PARAMS: CAPACITANCE= 0
*
* Analog/Digital interface for node 2
*
** Moving X1.U1.Y from analog node 2 to new digital node * 2$DtoA
X$2_DtoA1 2$DtoA 2 $G_DPWR $G_DGND DtoA_STD
+     PARAMS: DRVL=0 DRVH=0 CAPACITANCE=0
*
* Analog/Digital interface power supply subcircuit
*
X$DIGIFPWR 0 DIGIFPWR

.END ;(end of AtoD and DtoA interfaces)
```

Figure 15-2 Simulation output for mixed analog/digital circuit.

The lines that begin with "Moving...from analog node" indicate the new digital node names that were

generated. Below each of these are the interface subcircuit calls inserted by PSpice A/D.

In this example, the subcircuits named AtoD_STD and DtoA_STD are obtained from the I/O model that is referenced by the inverter primitive inside the subcircuit describing the 7404 part. The CAPACITANCE, DRVL (low-level driving resistance), and DRVH (high-level driving resistance) subcircuit parameter values come from the same I/O model.

After the interface subcircuit calls, PSpice A/D inserts one or more interface power supply subcircuits. The subcircuit name is specified in the I/O model for the digital primitive at the interface. In this example, PSpice A/D inserted DIGIFPWR, which is the power supply subcircuit used by all TTL models in the model library. DIGIFPWR creates the global nodes $G_DPWR and $G_DGND, which are the default nodes used by each TTL part.
Return to top of page

---

cadence