How to Use IBIS Models in PSpice

Author: Russell Carroll, PE Email: russell@emisleuth.com

Accurate device models are needed for signal integrity (SI) and electromagnetic compatibility (EMC) simulations of high-speed digital electronics. Device manufacturers usually provide IBIS, or "Input/Output Buffer Information Specification", models rather than actual SPICE models. This protects the manufacturer's intellectual property while giving accurate models for circuit analysis. Unfortunately, these IBIS models are not directly supported in the most popular versions of SPICE including PSpice. This article will cover the basic steps of converting and simulating a single ended IBIS model in PSpice.

I have developed an advanced IBIS to PSpice Converter tool along with an IBIS for PSpice symbol library. These tools will allow anyone to accurately simulate digital I/O with the manufacturer provided IBIS models is a PSpice environment including the free PSPICE-FOR-TI program. This is critical for engineers that need accurate modeling capabilities but don't have licenses for specialized SI tools like HyperLynx.

This article will simulate the serial clock (SCK) interface between a Microchip Flash Memory SST26VF064B (104 MHz max) and a Lattice FPGA iCE40 UltraPlus (185 MHz max clock).

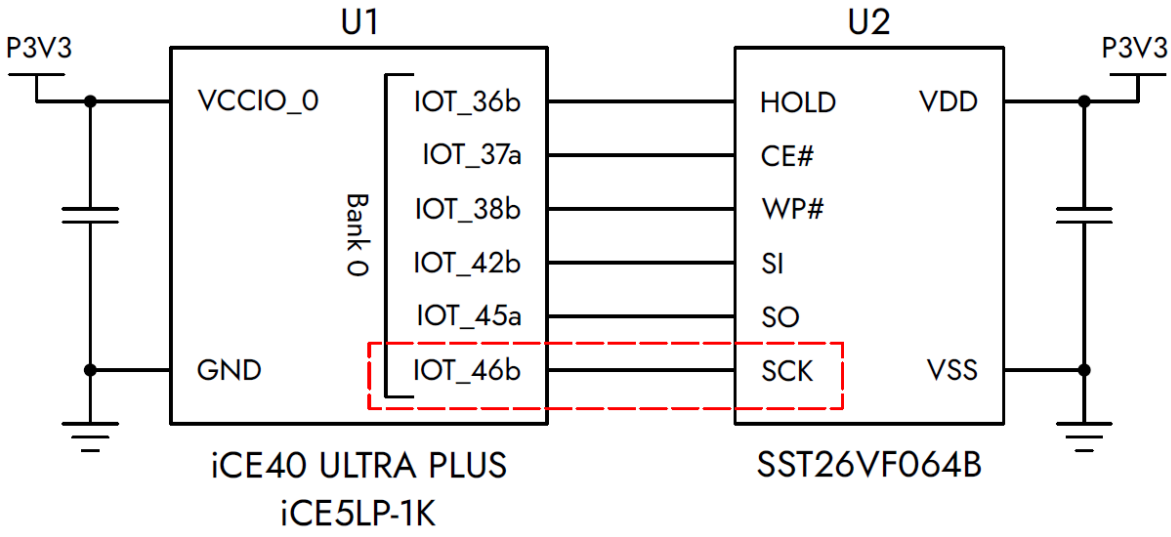


Figure 1. Schematic of High-Speed Flash Memory Interface to FPGA

Step 1. Gather and Review IBIS Models

Both IBIS models are available on the manufacturer's website for the iCE40 UltraPlus and the SST26VF064B chips. Key parameters were reviewed with a text editor with a summary shown in Table 1. The following conclusions were made:

1. The iCE40 UltraPlus FPGA LVCMOS 3.3V I/O pin IBIS model is well defined across the Typ, Min, and Max cases.
2. The SST26VF064B input pin IBIS model is only partially defined in the Min and Max cases with C_comp, GND_clamp, and POWER_clamp data missing. Also, the Typ rail voltage for the SST26VF064B is only 3.0V while it is being operated at 3.3V.

Table 1. Summary of Key Parameters in Manufacturer IBIS Models

Parameter	**SST26VF064B - Flash**			**iCE40 UltraPlus - FPGA**			Units
	Typ	*Min*	*Max*	*Typ*	*Min*	*Max*	
Model (Model_type)	IN1_SST (Input)			lvc330a080aaaaaaaaio (I/O)			-
R_pkg	110	55.17	121.24	181.50	34.879	763.86	mΩ
L_pkg	2.0	1.70	2.30	0.3362	0.0892	1.598	nH
C_pkg	0.53	0.50	0.56	0.3855	0.2457	0.543	pF
C_comp	6.0	NA	NA	3.30	3.17	3.43	pF
Voltage Range	3.0	2.7	3.6	3.30	3.14	3.47	V
GND_clamp	-0.437 to 0	NA	NA	-0.198 to -0.0001	-0.172 to -0.0001	-0.232 to -0.0001	A
POWER_clamp	0.0602 to 0	NA	NA	0.198 to 0.0001	0.172 to 0.0001	0.232 to 0.0001	A
Pulldown	-	-	-	-0.0671 to 0.0484	-0.0554 to 0.039	-0.085 to 0.0614	A
Pullup	-	-	-	0.285 to -0.0677	0.282 to -0.0569	0.294 to -0.0817	A
Rising Waveform	-	-	-	0.0473 to 2.116	0.0511 to 1.972	0.0424 to 2.481	V
Falling Waveform	-	-	-	3.256 to 1.242	3.094 to 1.386	3.430 to 1.067	V

Step 2. Convert IBIS Models to PSpice

The IBIS models are converted to a PSpice model library using the EMI Sleuth IBIS to PSpice Converter Tool web page as shown in Figure 2. The '.lib' files will automatically be downloaded. The standard PSpice symbol library can also be downloaded here.

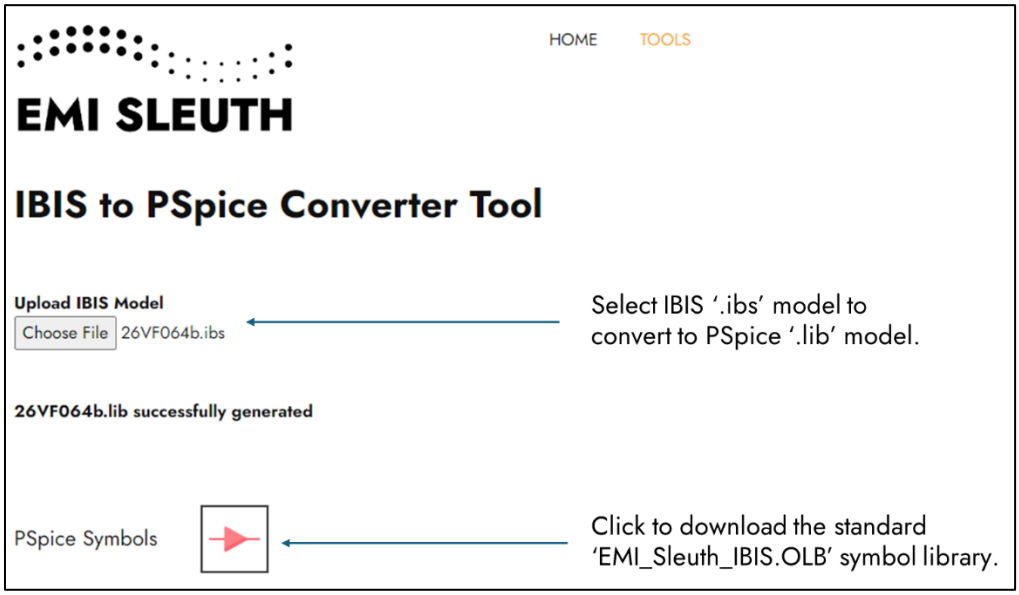


Figure 2. EMI Sleuth IBIS to PSpice Converter Tool Interface

Once the PSpice '.lib' files are generated they should be reviewed with a text editor. These PSpice library files have a header section with all the PSpice models listed for reference as shown in Table 2. Keep in mind that the same limitations to the Min and Max IBIS models identified in Step 1 will carry over to the generated PSpice models.

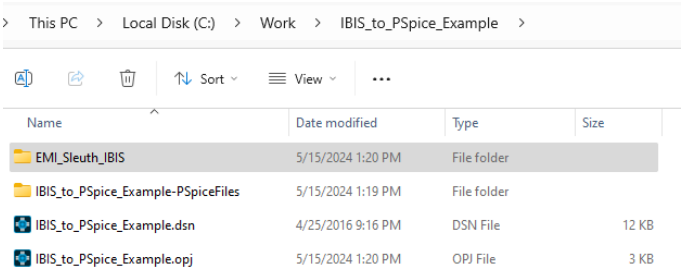
Table 2. Lists of Relevant PSpice Models from Generated '.lib' Model Files

SST26VF064B - Flash (26VF064b.lib)	**iCE40 UltraPlus - FPGA** (iCE40_UltraPlus.lib)
* -- List of PSpice Models -- * * ... * SST26VF064B_104I_SM_IN1_SST_TYP * SST26VF064B_104I_SM_IN1_SST_MIN * SST26VF064B_104I_SM_IN1_SST_MAX * ...	* -- List of PSpice Models -- * * ... * ICE40_ULTRA_PLUS_LVC330A080AAAAAAAAIO_TYP * ICE40_ULTRA_PLUS_LVC330A080AAAAAAAAIO_MIN * ICE40_ULTRA_PLUS_LVC330A080AAAAAAAAIO_MAX * ...

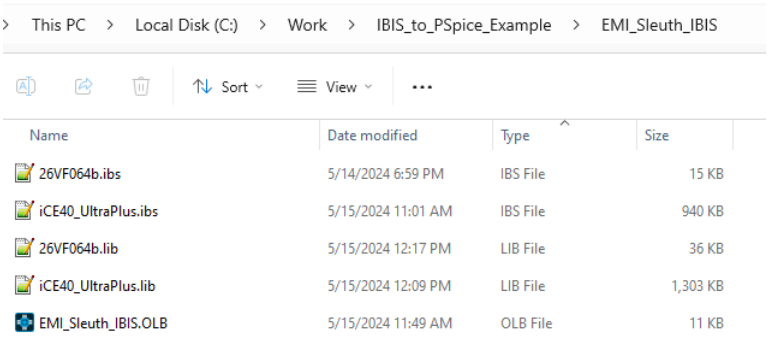
Step 3. Setup PSpice Project

For this example, a PSpice project was created called "IBIS_to_PSpice_Example". The following steps should be taken to integrate the downloaded PSpice model and symbol libraries:

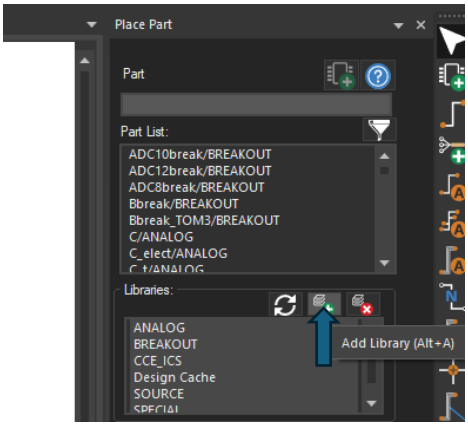
1. Create a new folder inside the project called "EMI_Sleuth_IBIS".

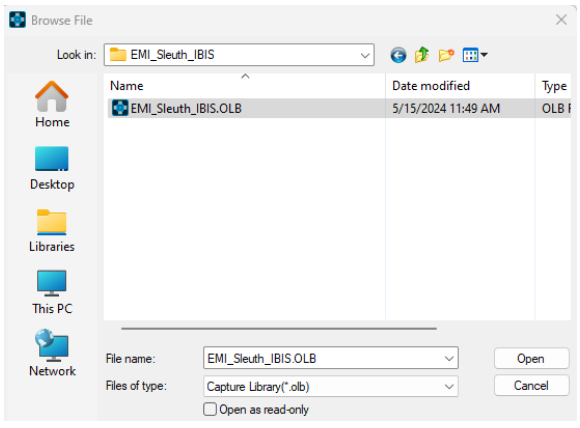


2. Add the '.ibs', '.lib' models and the '.OLB' symbol library to the new folder.



3. Add the "EMI_Sleuth_IBIS.OLB" symbol library to the part place libraries.

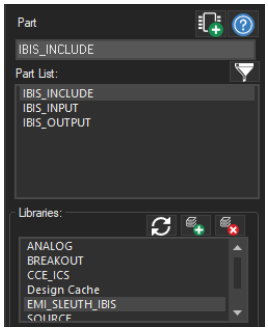




Step 4. PSpice Schematic Creation and Simulation

The schematic is created with the following steps:

1. Place two copies of the IBIS_INCLUDE models into the schematic. Add the names of the converter '.lib' files added in Step 3.



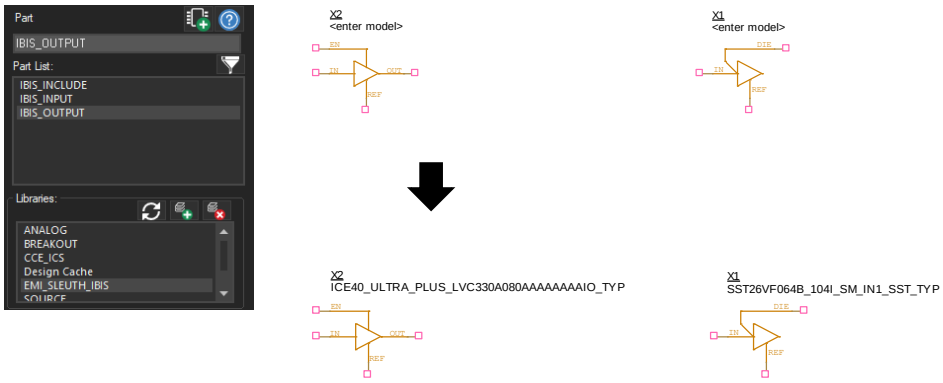
INCLUDE: ./../../../EMI_Sleuth_IBIS/<enter library>.lib

INCLUDE: ./../../../EMI_Sleuth_IBIS/<enter library>.lib

INCLUDE: ./../../../EMI_Sleuth_IBIS/26VF064b.lib

INCLUDE: ./../../../EMI_Sleuth_IBIS/iCE40_UltraPlus.lib

2. Place an IBIS_OUTPUT and IBIS_INPUT symbol into the schematic. Update the model names to match the input and output PSpice models identified in Step 2.



3. Add an ideal square wave on the logic input to the IBIS_OUTPUT driver. This example will use a 90 MHz clock to simulate the maximum achievable SCK speed on the FPGA.

4. Add a transmission line to simulate the propagation effects of a PCB trace.

5. If desired, include impedance mismatch effects such as matching resistors, vias or transmission line discontinuities.

A completed schematic is shown below in Figure 3. This schematic includes a 50 Ω transmission line with a 500 ps propagation delay and a zero-ohm jumper resistor at the driver output. Voltage probe points are added to the ideal LOGIC clock source, the driver OUTPUT node, and the INPUT_DIE node at the receiver. The simulation in Figure 4 shows significant overshoot and undershoot at the receiver due to the impedance mismatch at the driver.

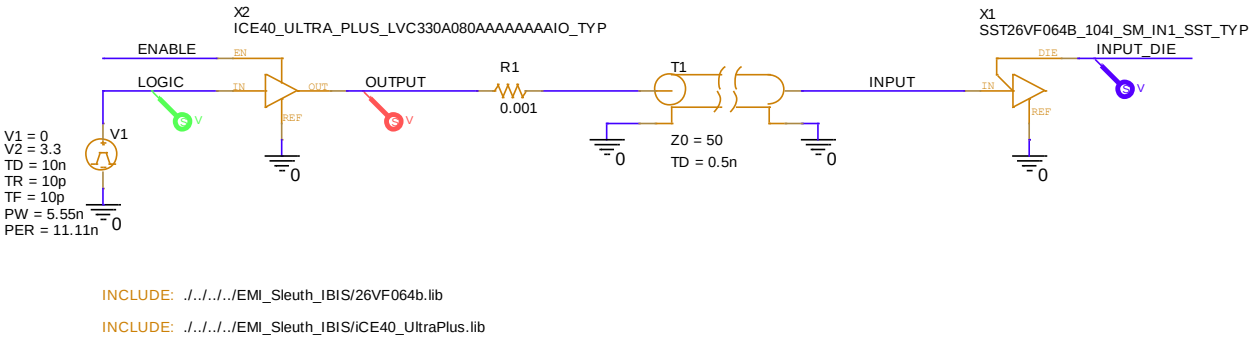


Figure 3. PSpice Schematic of SCK Channel without Matching

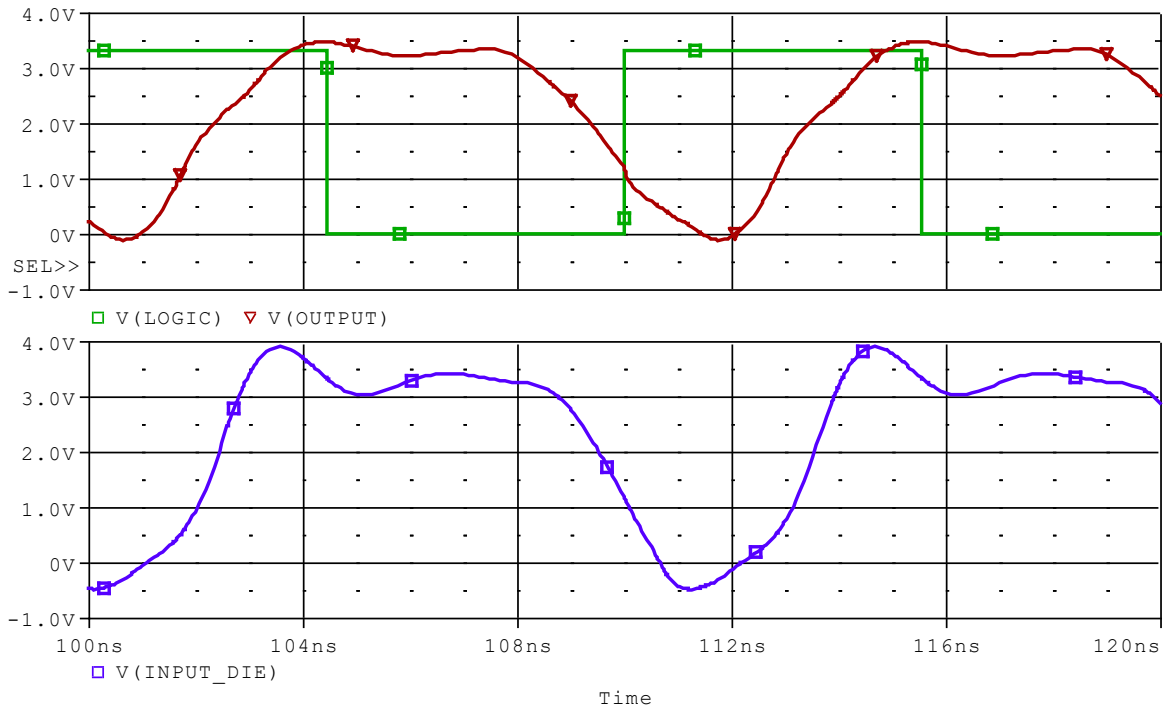


Figure 4. Simulated Output and Input Waveforms of SCK Channel without Matching

The matching resistor was increased to 22 Ω and shown in Figure 5. The simulation in Figure 6 shows a significant reduction to overshoot and undershoot at the receiver due to the improved impedance match at the line driver. However, closer inspection of the INPUT_DIE signal is required to ensure compliance to the waveform requirements of the receiver.

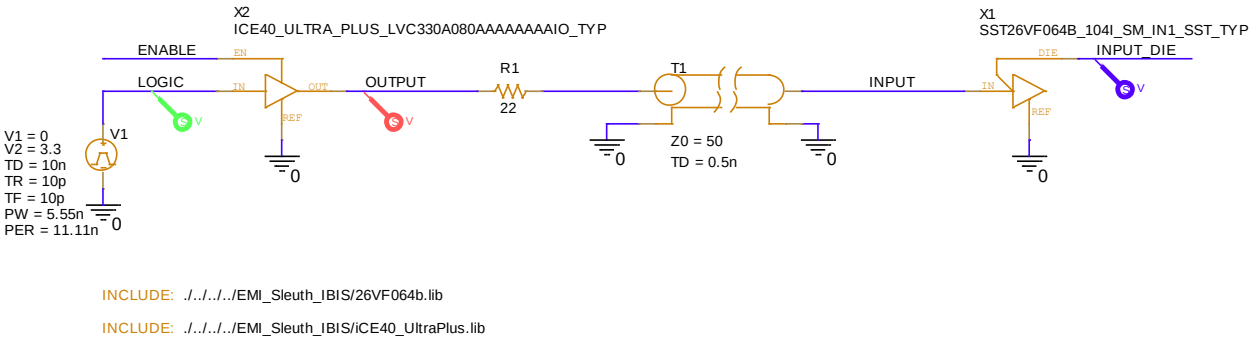


Figure 5. PSpice Schematic of SCK Channel with Matching

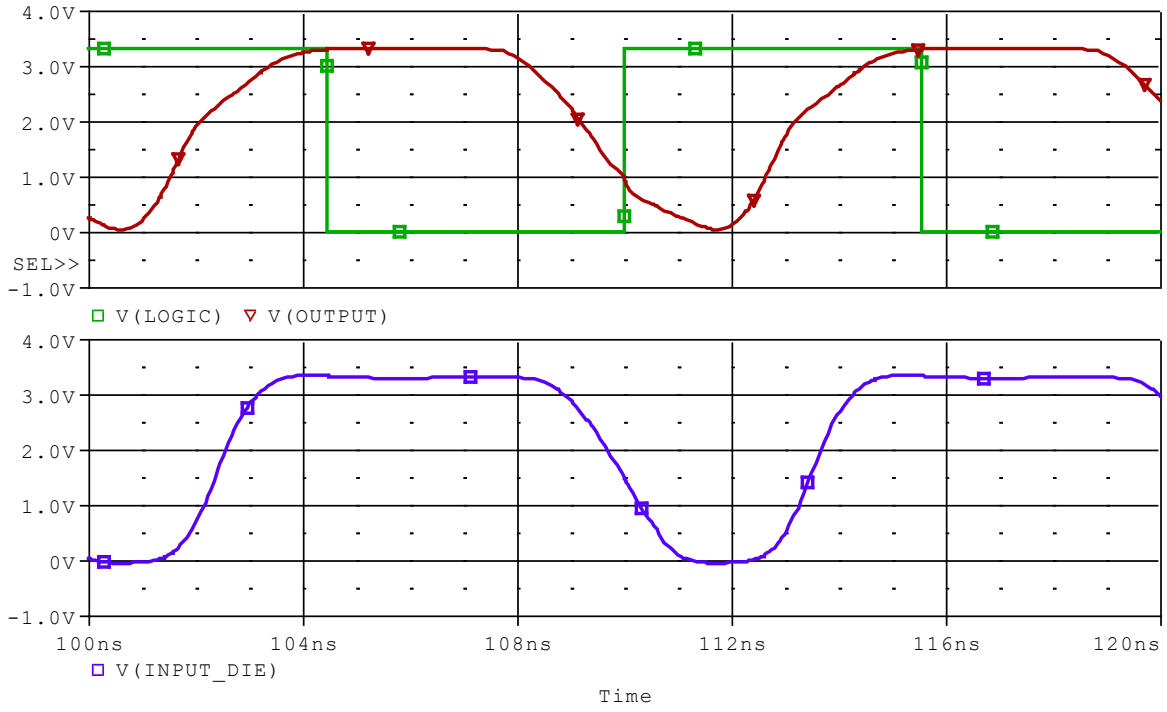


Figure 6. Simulated Output and Input Waveforms of SCK Channel with Matching

The AC timing characteristics of the flash memory SCK input pin is shown below in Figure 7. The receiver requires a minimum "Serial Clock Low Time" of 4.5 to 5.5 ns for a 104 or 80 MHz clock respectively which can be linearly interpolated to approximately 5 ns for the 90 MHz clock used in this simulation. The simulations above predict a typical low time of approximately 3.5 ns at the INPUT_DIE signal and therefore the timing requirements will not be met at the maximum FPGA frequency and a slower clock speed is required.

SST26VF064B/SST26VF064BA

8.0 AC CHARACTERISTICS

TABLE 8-1: AC OPERATING CHARACTERISTICS (V_{DD}[1] = 2.3V - 3.6V)

Symbol	Parameter	Limits - 40 MHz		Limits - 80 MHz		Limits - 104 MHz		Units
		Min	Max	Min	Max	Min	Max	
F_{CLK}	Serial Clock Frequency		40		80		104	MHz
T_{CLK}	Serial Clock Period		25		12.5		9.6	ns
T_{SCKH}	Serial Clock High Time	11		5.5		4.5		ns
T_{SCKL}	Serial Clock Low Time	11		5.5		4.5		ns
T_{SCKR}[2]	Serial Clock Rise Time (slew rate)	0.1		0.1		0.1		V/ns
T_{SCKF}[2]	Serial Clock Fall Time (slew rate)	0.1		0.1		0.1		V/ns

Figure 7. Selected Timing Requirements of the SST26VF064B SCK Input

Conclusions

This article demonstrated that using accurate IBIS models in PSpice is possible with the EMI Sleuth IBIS to PSpice Converter Tool. A PSpice simulation of a 90 MHz serial clock channel interface was simulated between an FPGA and a flash memory device. These IBIS simulation models were shown to predict significant overshoot and undershoot at the receiver and was able to identify the appropriate driver matching resistor to minimize the impedance mismatch. This simulation also showed that the AC timing characteristics of the receiver could not be met with this FPGA at the maximum rated frequency and a lower clock speed is required for compliance. Therefore, a PSpice simulation with accurate IBIS driver and receiver models was able to predict a significant timing compatibility issue between the FPGA and the selected flash memory device at the desired clock rate.

