Mixed-Signal Systems-on-Chip Design Verification With Automatic Real-Number Abstraction

Nancy Qiu,[1] Frank Yang,[1] and Himadri De[1]

[1]Analog Devices Incorporation, {nancy.qiu,frank.yang,himadri.de}@analog.com

Introduction: All systems-on-chip (SoC) are mixed-signal. While comprehensive verification methodologies have been developed for digital circuits, no effective methodology exists for analog blocks. Mixed-signal simulation (either co-simulation or the same kernel) is generally hard to use, and less effective due to nonlinear equation solving involved. Modeling analog blocks using real values in System-Verilog has been demonstrated to be effective for high throughput verification. However existing manual model creation is error prone, tedious, and requires the expert knowledge of analog circuits, digital languages, simulators, and system test benches. In this paper, we present ***for the first time*** a methodology based on ***automated real number abstraction***.

Mixed Signal Verification Challenges: Efficient generation of behavioral models of analog circuits and model validation are two major challenges in our present mixed-signal verification flow. Silicon bugs remain uncovered during verification due to improperly represented and correlated analog behavioral models; often resulting in first silicon failures and delayed time to market of the product. This paper describes a new automated real number abstraction methodology which can greatly resolve these problems and thereby minimize errors during the verification phase.

As is well known, behavioral modeling of analog circuits and proper validation between models and schematics are two big challenges in mixed signal verification flow in industry as well as in Analog Devices., improper modeling or out of date models sometimes resulting in failures which make the design not work after silicon back from the foundries. Methodologies used in ADI to generate behavioral models include ADICE schematic2, Cadence Schematic Model Generator (SMG), or hand-writing .am/.va/.vams/RNM models; all of which need a large amount of time to compose, and difficult to maintain during the design cycle. The flow of ADICE schematic2 model, which is very close to Cadence Schematic Model Generator, requires users to build up model view with some primitively behavioral elements, which thus induce model accuracy loss, speed balance, as well as model validation and maintenance. Those basic behavioral primitives are written by some internal developed language which is also evolved from C++ and event-driven based calculation and execution algorithm. With this methodology, users don't need to have programming skill but the model re-use, maintenance, and validation are several big challenges. The other popular methodology is hand-writing behavioral modeling which requires users to have good programming skill that appears to be challenging for circuit designers, it's also bad on reusability. Automated real number abstraction methodology raise up, perfectly fit into the application requests and effectively resolve above mentioned problems. It provides overhead advantages with simply pin type definition and can easily generate integrated models at any time and guarantee that they match with real schematics.

Methodology & Flow: The key innovation is the introduction of signal type driven abstraction. We have observed that all electrical nets in mixed-signal circuits can be classified into four groups:

- **Source, reference and biasing. Enable the circuit to function**
- **Data signal: the input and output signals represent information provided and processed**
- **Control, configuration and clocks: control the input and output data signal processing mode**
- **Feedthrough and loading: no functional purpose**

Furthermore, the data signals can be classified into four types only:

- **Digital data signal**
- **Large-analog data signal**
- **Small-analog data signal**
- **Periodic-analog data signal**

These types are captured by a pin table, which consists a specification provided by designers. These pin types can be represented in System-Verilog. All the circuit blocks that operate on these real value types can be automatically

abstracted from schematics using topological tracing and model order reduction. The tool Arana has been developed to generate System-Verilog, Wreal, or Verilog-A models from the schematic and pin table.

A. *S2R(schematic to RTL) Methodology*

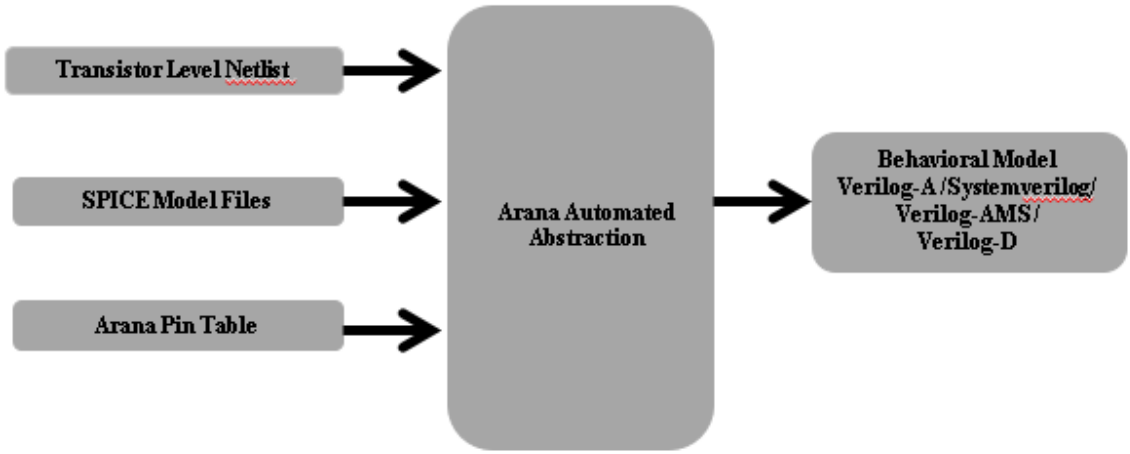


Figure.1 Schematic To RTL Methodology & Flow

In above Figure1, in S2R Methodolgy, automated abstraction is enabled through designers' transistor level netlist along with user specified pin specification input to Arana PinTable. Thus, it's a very systematic way of abstracting a particular function of a circuit to get behavior model which matchs with circuit's function.This Methodology facilitates reducing error prone model development time for any desired abstraction as pin matched model is very much consistent with schematic. In this methodology, user have choice to automatically generate System Verilog, WREAL and Verilog-A models from circuit schematic.

B. *SvR(Spice vs System Verilog) Methodology*

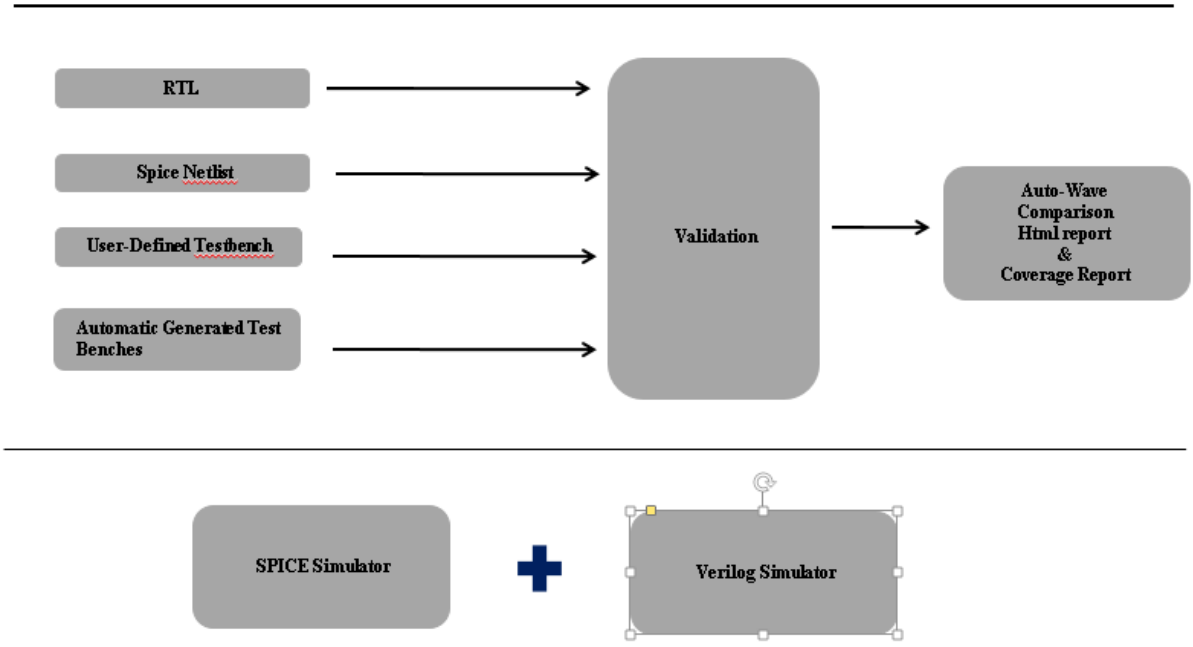
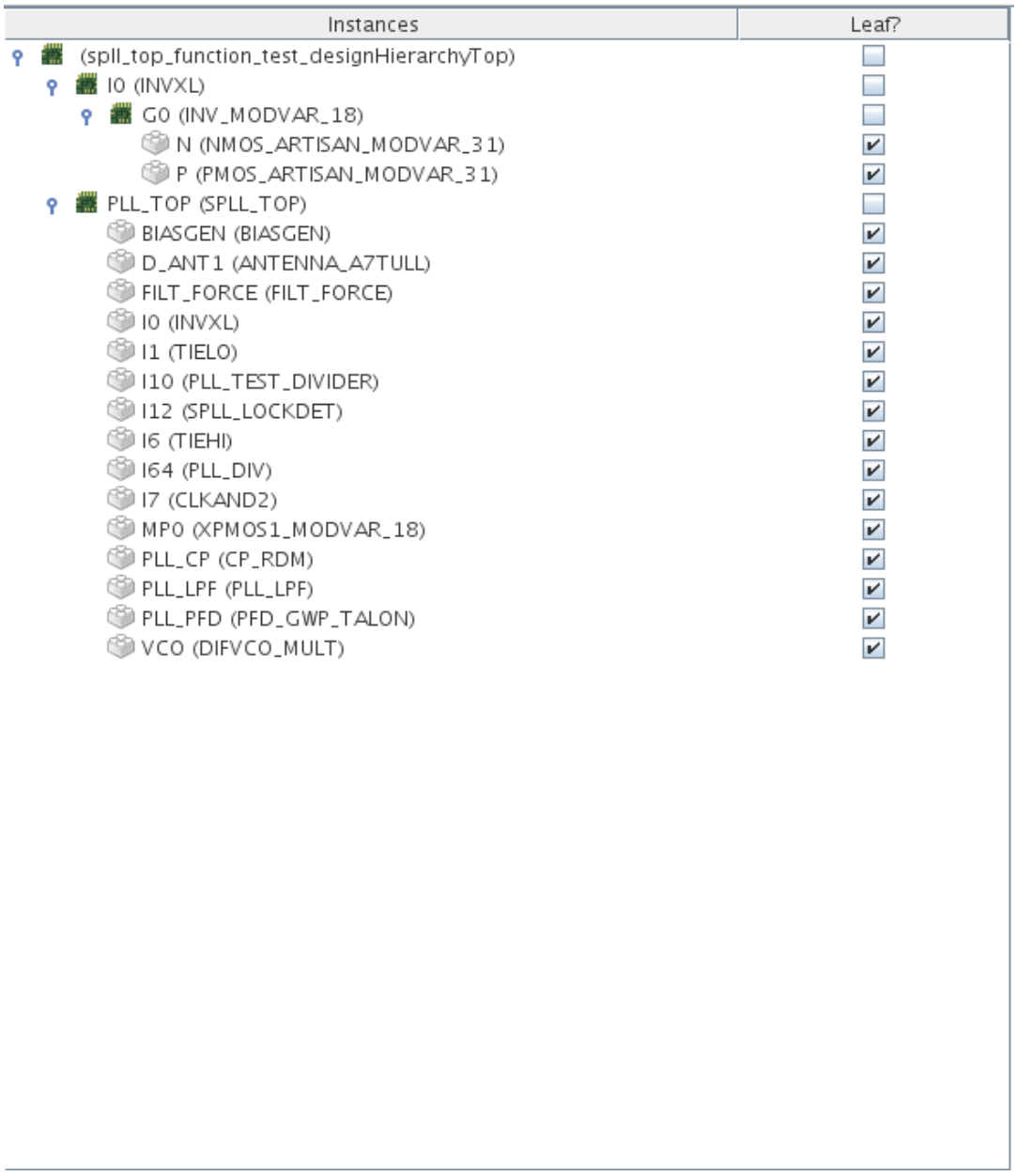


Figure 2. SvR Methodology & Flow

Application Of S2R Methodology Results on Analog Devices Circuits:

Figure2 describes SvR Methodology which facilitates automatic creation of Spice as well as System Verilog/Verilog-A testbench for validation of Arana Auto Abstracted models. It automatically does the waveform comparison of Spice vs. System Verilog/Verilog-A results and shows the results in html report.

C. SPLL_TOP: Netlist Hierarchy



D. PLL_TOP Pin Specification for Arana Automated Abstraction

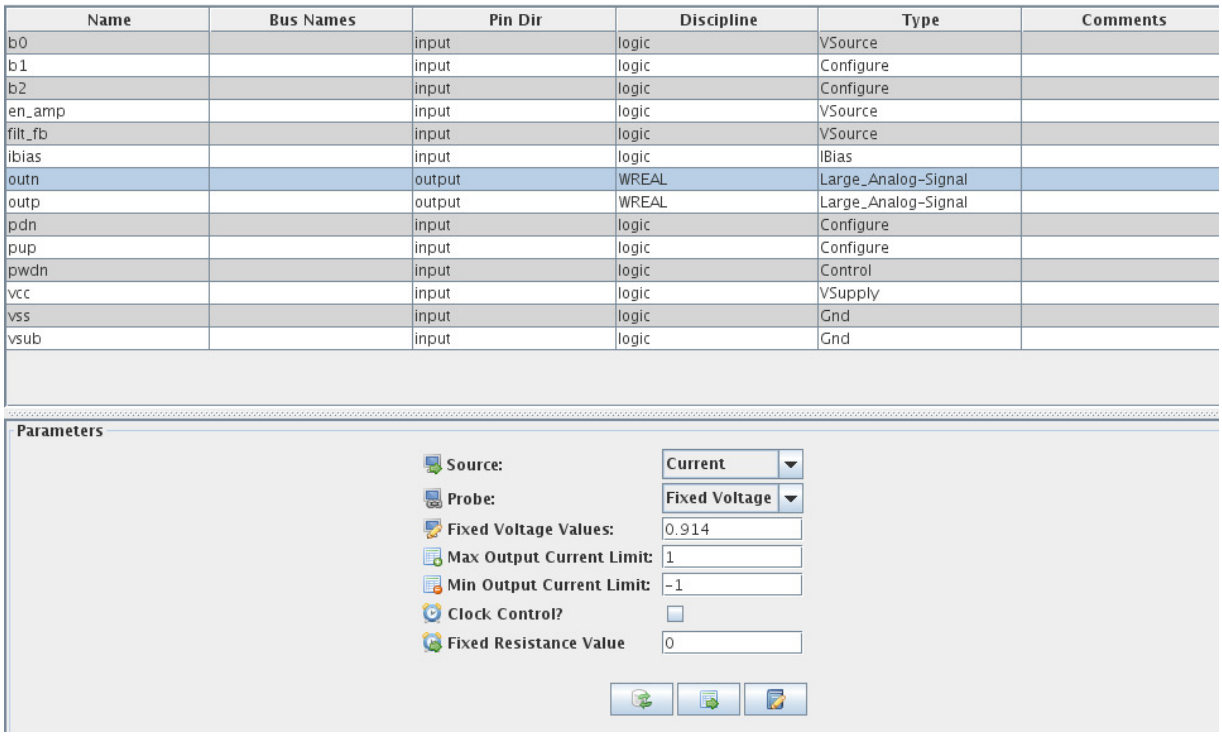


Figure 3. Large Analog Signal Type in AranaBMP Pin Table

The Large_Analog_Signal type is for signal pins of an analog circuit that take large signals, which may lead to the circuit working in the nonlinear region. Large_Analog_Signal is the opposite to Small_Analog_Signal, which takes small signals to make the circuit to work in the linearized region. Pins of type Small_Analog_Signal also have saturation limiting and slew-rate limiting too, but do not as accurately capture nonlinearity as pins of type Large_Analog_Signal. Figure3 is giving to display how Large_Analog_Signal generally works.

Theoretically user can always set signal pins as Large_Analog_Signal for both large and small signal circuits. However, to handle the specialty of the large signal circuit and fully utilize the advantage of linearity of the small signal circuit, Arana generates behavioral models with different methodologies for circuits with Large_Analog_Signal and Small_Analog_Signal pins. User should choose proper types as well as parameters for each type for corresponding large or small-signal circuits to generate the models with the best accuracy and efficiency tradeoff that meet the design, verification and application requirements.

To achieve the most abstraction and to generate the most efficient behavioral models, the tool Arana-BMP provides the following six combinations of probe and source settings as shown in Table 1.

TABLE 1

Large Analog Signal Pin Type

Combination	Source	Probe	Direction
1	voltage	Current	inout
2	voltage	zero current	output
3	current	Voltage	inout
4	current	Fixed voltage	output

5	none	Voltage	input
6	none	Current	input

Here, the combination of source current and probe fixed voltage indicates that the current is the source (target of a contribution statement) and the voltage is fixed at a particular level as a constant. Usually this is for the case that only the current property is needed to accurately model the signal on this pin. Its impedance to the ground is infinite. In practice, this implies this pin current is independent of its voltage level.

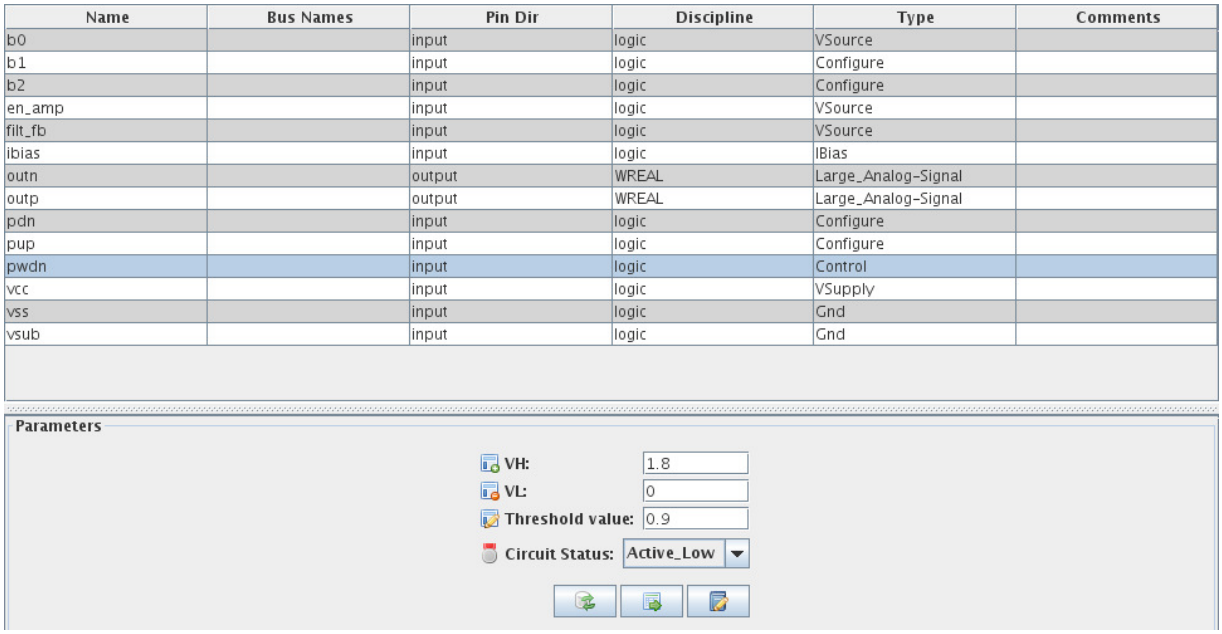


Figure 4. Control Pin Type In AranaBMP Pin Table

The control pins (figure 4) are pins having only two states: active and inactive. When a control pin is active, the circuit functions normally, and when this pin is inactive, the circuit works as set or reset.

The primary use of the control type is to model pins such as power up and power down. If signals on certain pins are continuous analog signals, you can sometimes simplify the signals to two states to get simplified behavioral models.

The data fields associated with the Control type are Circuit Status, VH, VL and Threshold value. The Circuit Status field is used to select the active state of the control pins. Active High option means the circuit works normally when the voltage on the control pin is set as the VH value. Active Low option means the circuit works normally when the voltage on the control pin is set as the VL value. The Threshold value is used to determine the two states of the signal.

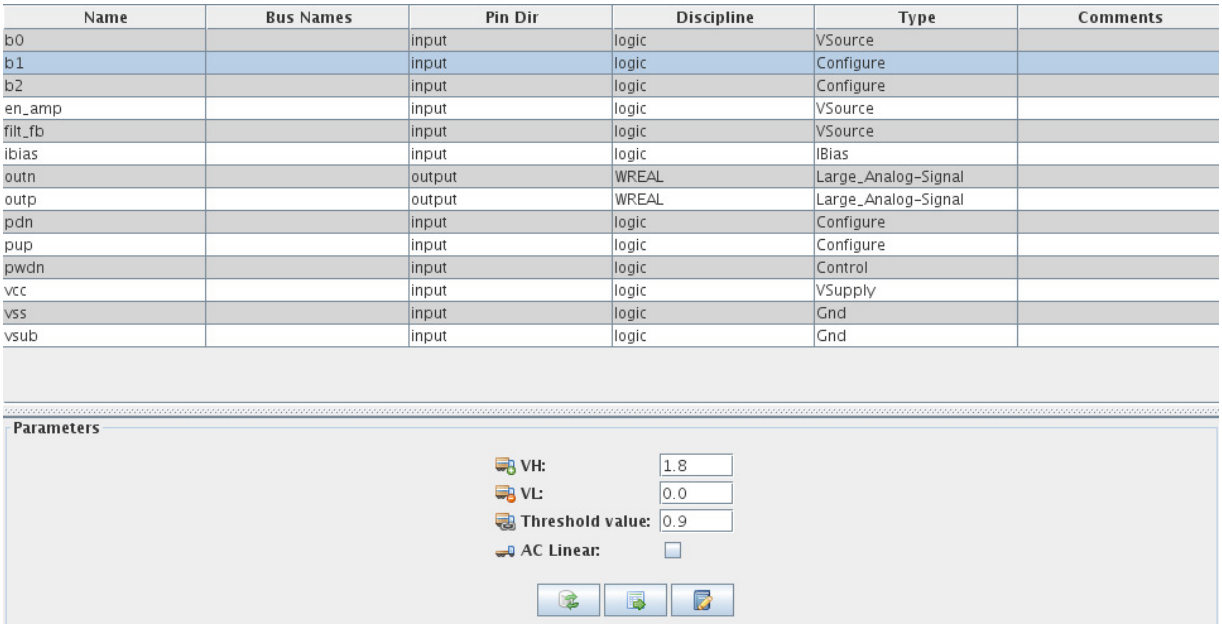


Figure 5. Configure Pin Type In AranaBMP Pin Table

The configure pins(figure 5) are pins with low and high states. The circuit exhibits different functionality in two different states. The difference from Control pins is that now both two states are normal working states of the circuit. Usually the circuit may have multiple Configure pins and the circuit's function is different for different combinations of Configure pin states, for example, for a circuit with N Configure pins, there can be up to 2^N total different functional states.

The data fields associated with Configure type are VH, VL and Threshold value. The VH is the maximum voltage value of this pin and the VL is the minimum voltage value of this pin. The Threshold value stands for the voltage value of the threshold used to determine the states of the signal.

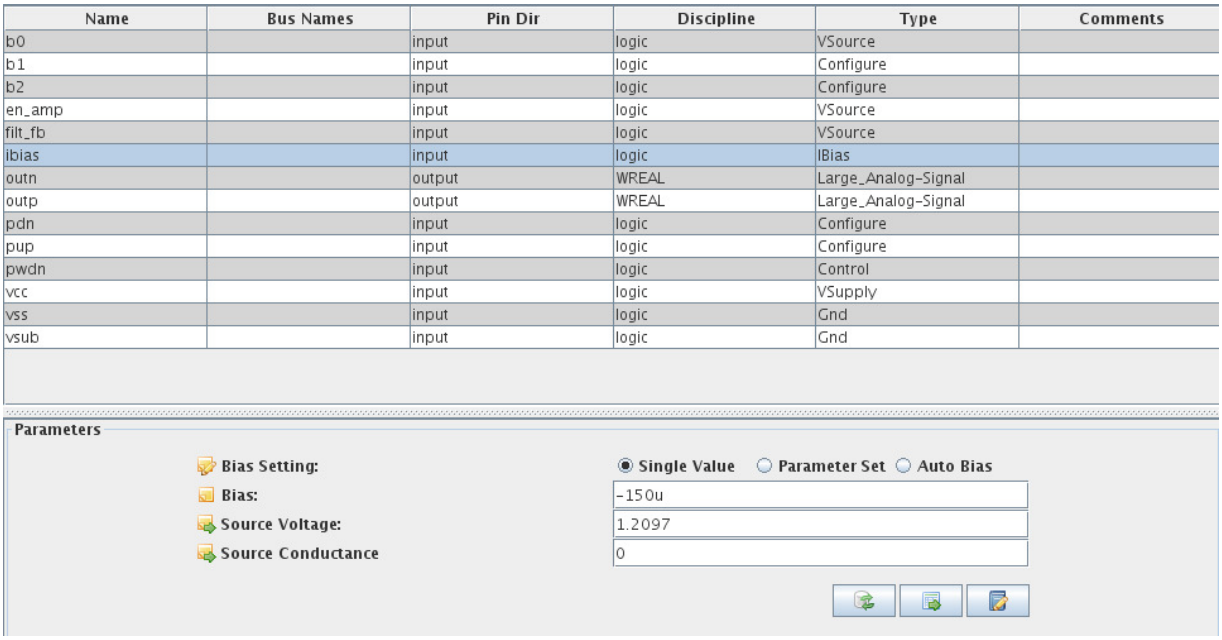


Figure 6. Ibias Pin Type In AranaBMP Pin Table

The **IBias** type(figure 6) is used for the current bias pin. When Gs=0, Ibias is an ideal current bias. In practice, when Gs is small compared to the input conductance of the circuit, the bias pin is applied, and it can be viewed as an ideal current bias. For example, a current mirror with high output resistance can be modeled as an ideal current source.

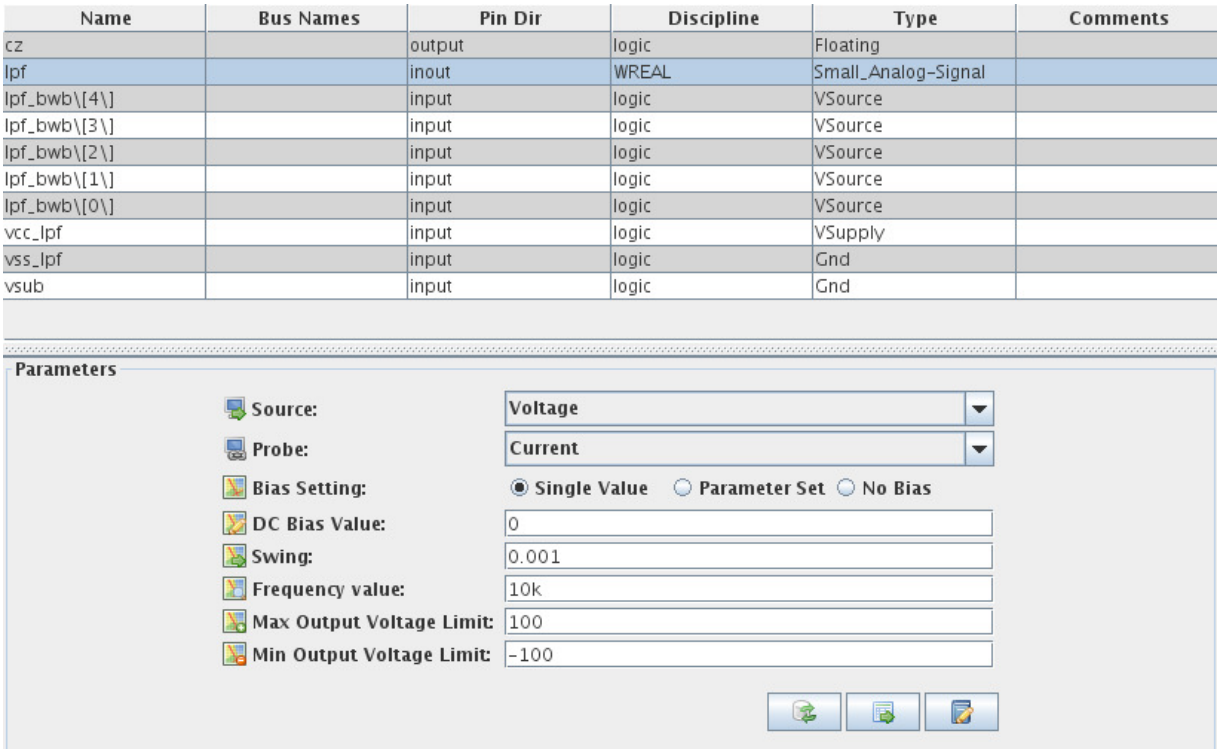


Figure 7. Small Analog Signal Type in AranaBMP Pin Table

The settings for Small_Analog_Signal pins (figure 7) include two parts. The probe and source setting part is similar to the Large_Analog_Signal type and the difference is that small analog signal pins do not need breaking points but need the values of the dc bias point and the signal swing.

Consider one of the most common analog circuits, an amplifier, as an example: if the amplifier works within the linear region, user should set signal pins to Small_Analog_Signal to generate the behavioral model. But if the amplifier works out of the linear region and the nonlinearity is important, you should set signal pins to Large_Analog_Signal.

In additional to the bias value, there are several parameters, Swing, Frequency, Maximum/minimum output values (if source voltage or source current is selected) associated with pin type Small_Analog_Signal, These parameters will be used in generating the assertions in the model for design verification. More specifically, we have

(1) Swing: the maximum swing, peak-to-peak, of the signal
(2) Frequency: the bandwidth of the small-signal

Maximum/Minimum (current/voltage) output values: the maximum value of the output current or voltage value; which are to be used for output limiting.

E. SPLL_TOP: Transistor vs. RNM Model Result

TABLE 2
Transistor vs. RNM Model Result

Region	Transistor Result	RNM Result
1	924mV	921mV
2	804mV	807mV
3	861mV	859mV
4	859mV	860mV

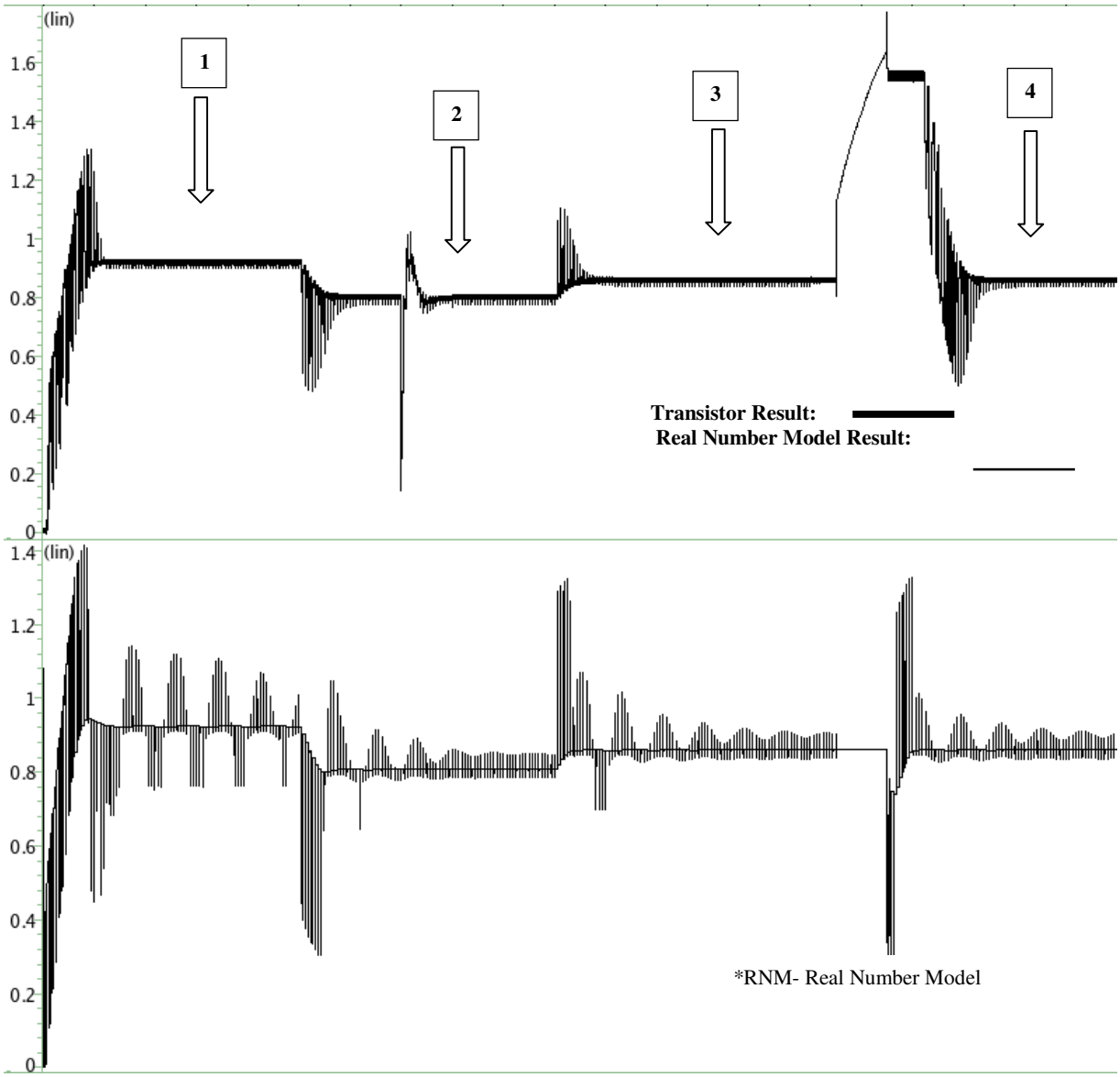
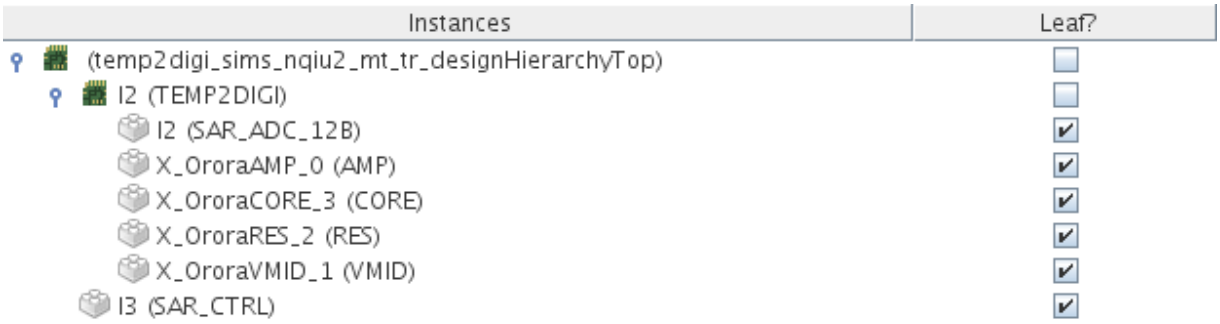


Figure 8. PLL Control Voltage Output- Transistor vs. Arana Abstracted RNM Model

F. SAR_ADC_12B Netlist Hierarchy



G. SAR_ADC_12B: DAC Pin Specification

Name	Bus Names	Pin Dir	Discipline	Type	Comments
[adc_sar\[11\]+adc_sa...	adc_sar[11:0]	input	logic	Configure	
adc_vin		input	WREAL	Large_Analog-Signal	
advdd		input	WREAL	VSupply	
advss		input	WREAL	Gnd	
agnd		input	WREAL	Gnd	
avdd		input	WREAL	VSupply	
sar_ref_hi		input	WREAL	VBias	
sar_ref_lo		input	logic	VBias	
tst_sar		input	logic	VBias	
tst_vin		input	logic	VBias	
samp		input	logic	Clock	
en		input	logic	Clock	
p2b		input	logic	Clock	
p2		input	logic	Clock	
p1b		input	logic	Clock	
p1		input	logic	Clock	
vip_comp		output	WREAL	Large_Analog-Signal	
vin_comp		output	WREAL	Large_Analog-Signal	

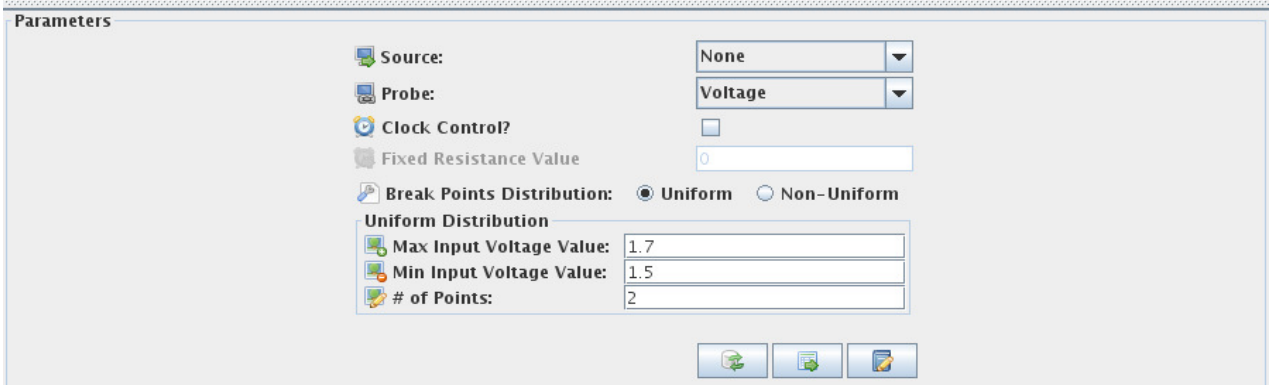


Figure 9. Large Analog Signal Pin Type for AranaBMP Pin Table

The combination of source voltage and probe zero current indicates that the voltage is the source (target of a contribution statement) and there is no probe. Usually this is for the case that only voltage property is needed to accurately model the signal on this pin. Its impedance to the ground is ideally zero. In practice, this implies this pin voltage is independent of its loading

Clock signals are well-defined and used by designers. They are signals that synchronize the circuit operation. Circuits with clock signals include synchronized digital and analog circuits, dynamic logic circuits, and enabling periodic discrete signal processing circuits such as those in switched-capacitor circuits.

For a clock signal, user need to provide the high, low and threshold values of the clock signal.

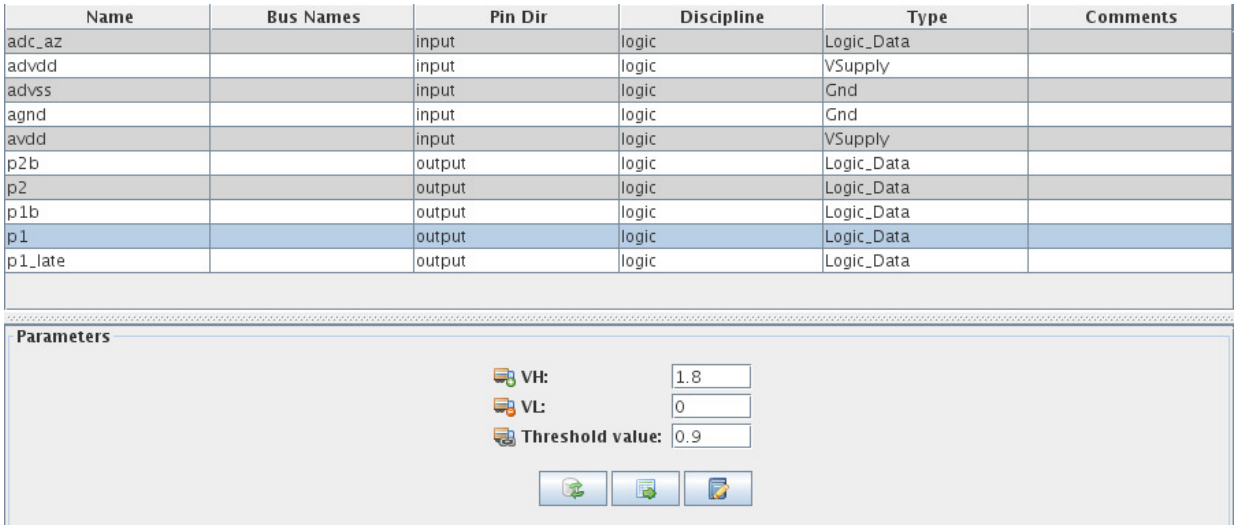


Figure 10. Logic Data Pin Type In AranaBMP Pin Table

H. *SAR_ADC_12B: Transistor vs. RNM Model Result*

TABLE 3
SAR_ADC: Transistor vs. RNM Model Result

DAC Bits	Transistor Result	RNM Result
Bit- 11	946mV	946mV
Bit- 10	806mV	807mV
Bit- 9	876mV	876mV
Bit- 8	911mV	911mV
Bit- 7	893mV	894mV
Bit- 6	902mV	903mV
Bit- 5	898mV	898mV
Bit- 4	902mV	902mV
Bit- 3	901mV	901mV
Bit- 2	900mV	900mV
Bit- 1	900mV	900mV

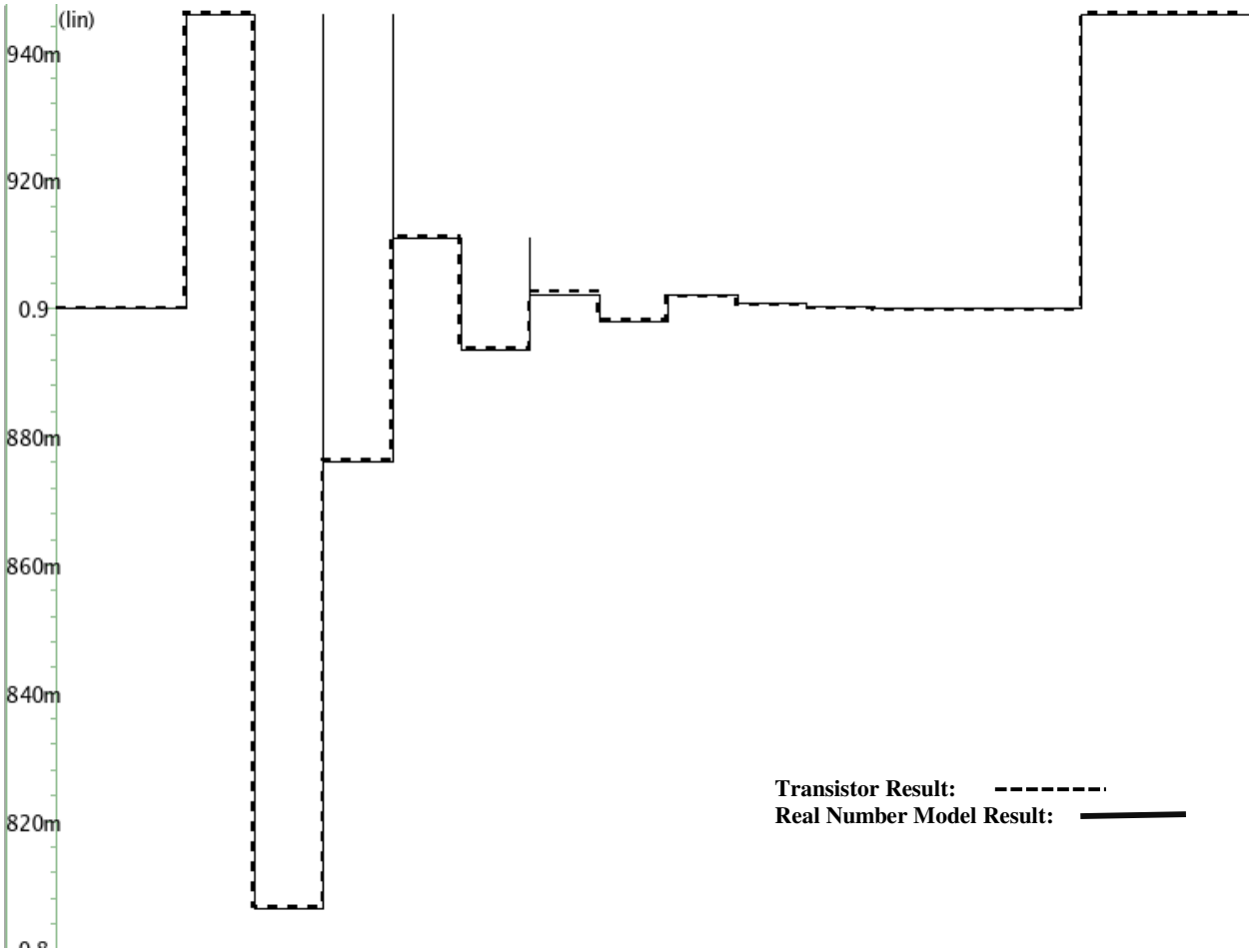


Figure 11. DAC output result- Transistor vs. Arana Abstracted RNM Model

I. MSDRV_V Netlist Hierarchy (Slew Control Circuit)

Instances	Leaf?
(msdrv_v_tb3_designHierarchyTop)	☐
I0 (MSDRV_V)	☐
I0 (MSDRV_V_3V_NATIVE_REGULATOR)	☑
IAMPDRV (MSDRV_V_AMP_DRV)	☑
IREFMUX (MSDRV_V_REF_MUX)	☑
IREG (MSDRV_V_DIG)	☑
ISLEW (MSDRV_VOLTAGE_SLEW_CNTRL)	☐
I0 (SLEW_SEL_DEC)	☑
ISLEWDIG (VSLEW_DIG)	☑
X_Ororacomp_1 (comp)	☑
X_Ororaremb_ch_2 (remb_ch)	☑
X_Ororatime_rcckt_0 (time_rcckt)	☑

J. MSDRV_V: Transistor vs. RNM Model Result

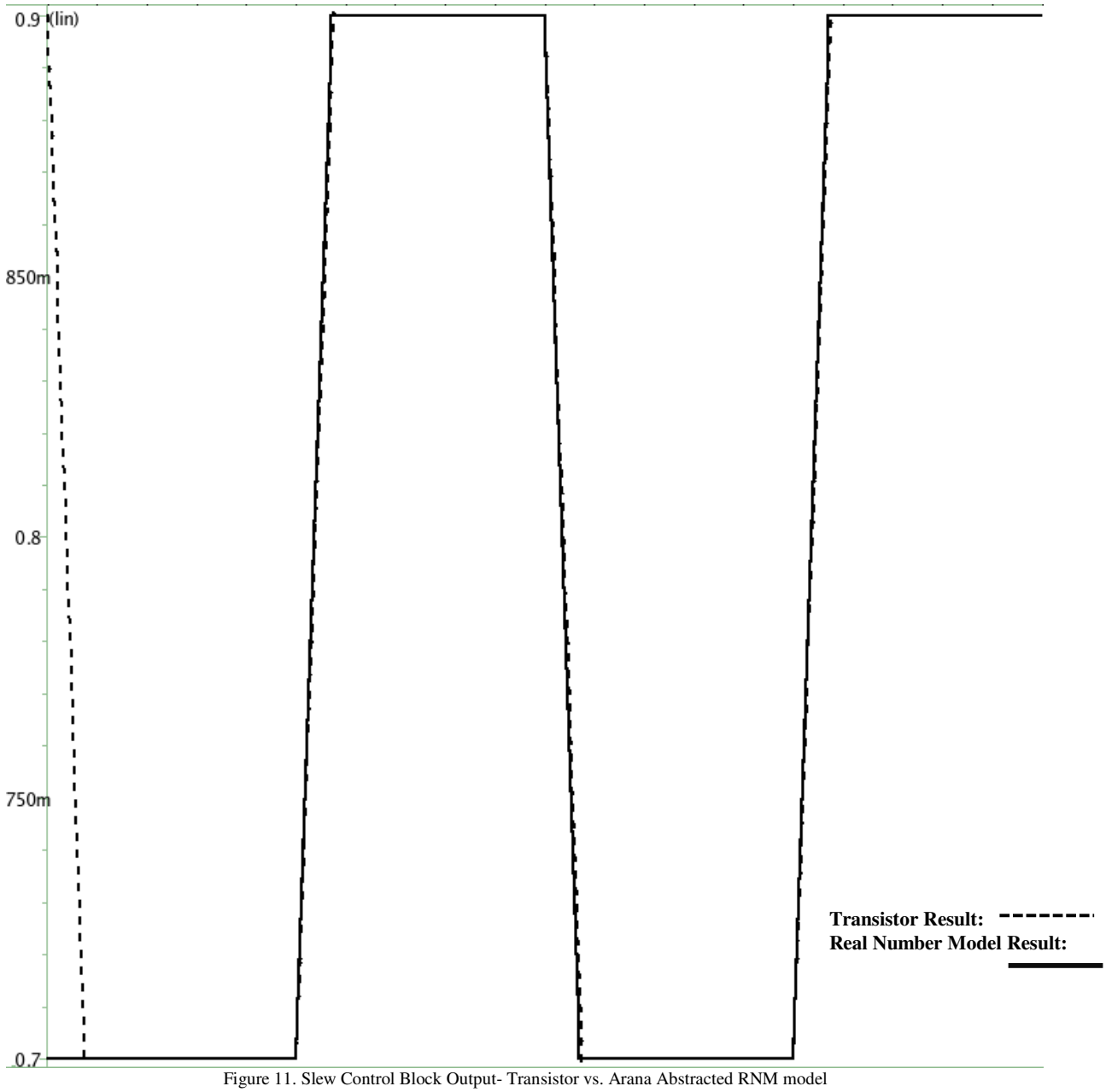


Figure 11. Slew Control Block Output- Transistor vs. Arana Abstracted RNM model

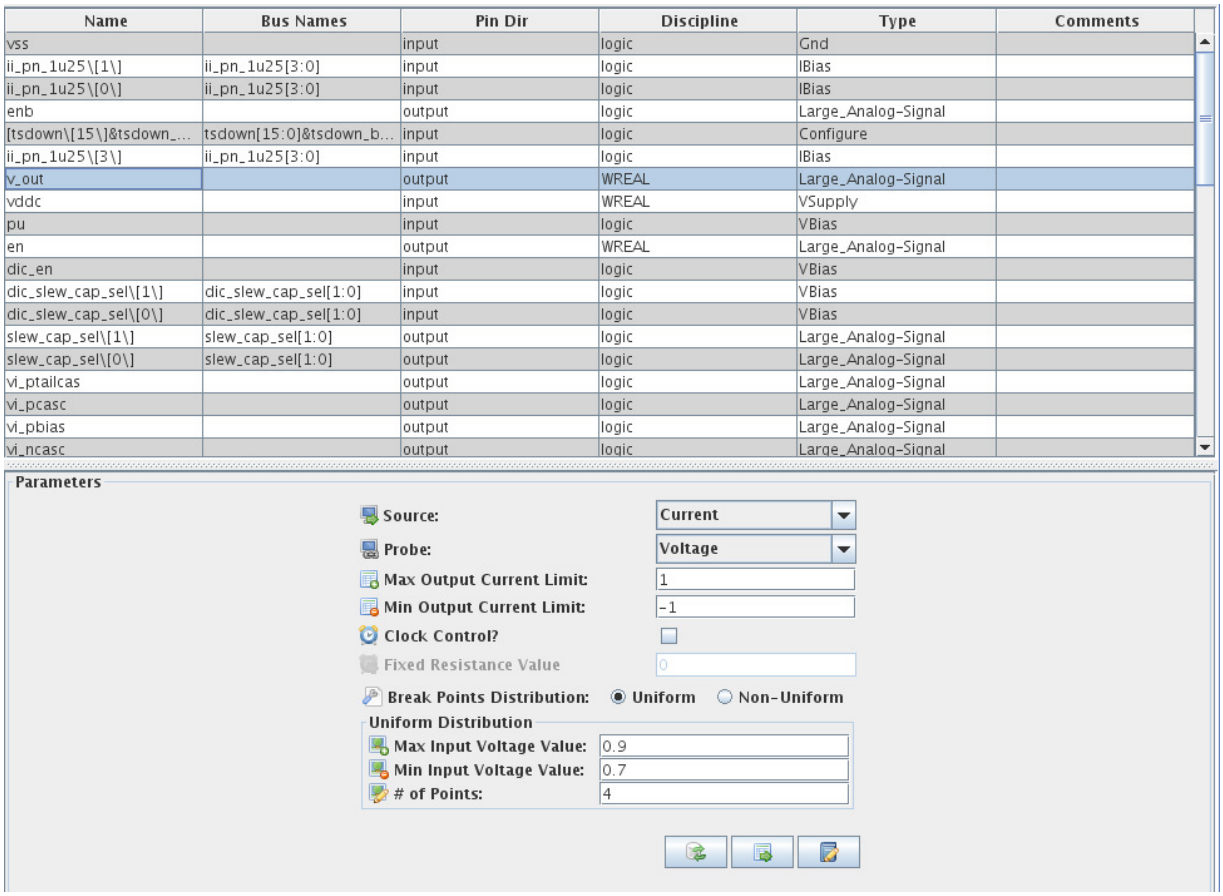
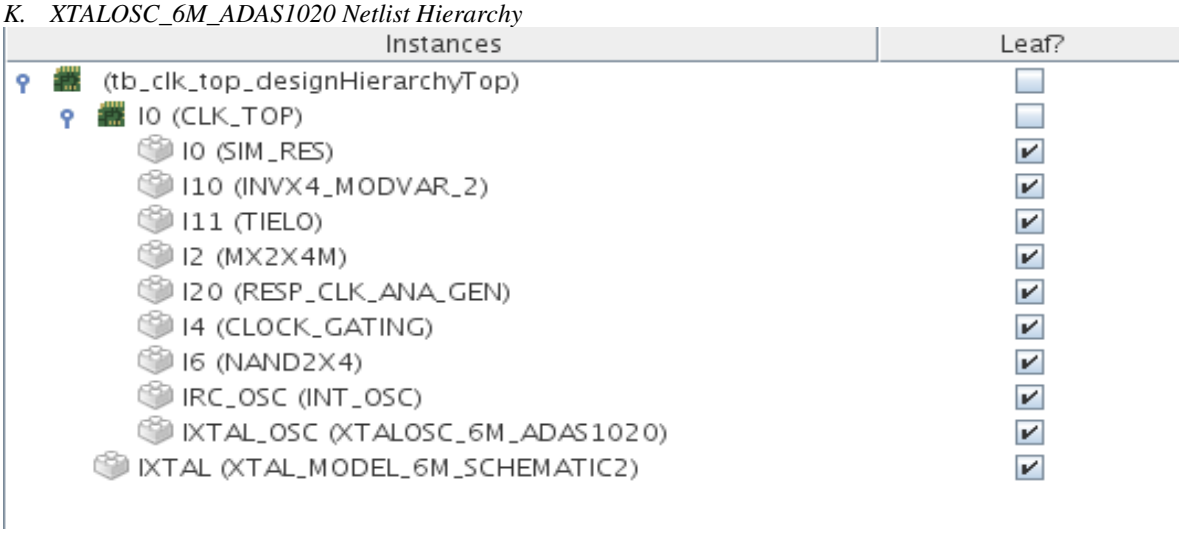
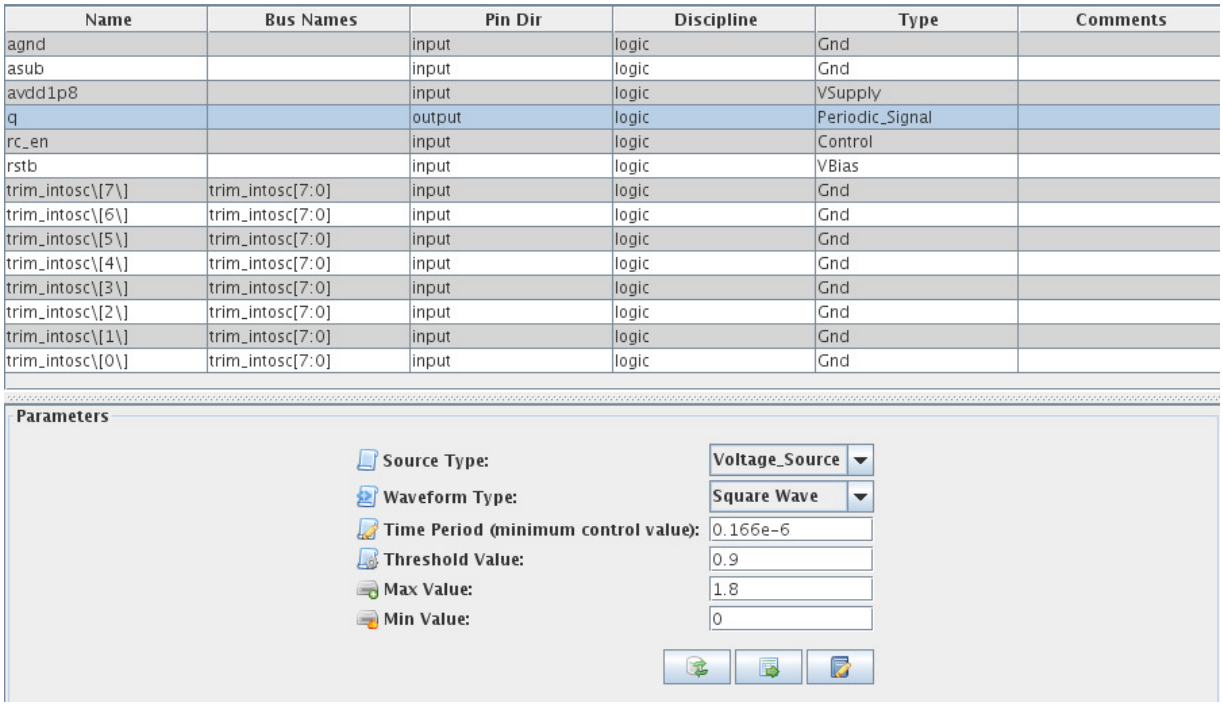


Figure 12. Large Analog Signal Pin Type In AranaBMP Pin Table

The combination of source current and probe voltage indicates that the current is the source (target of a contribution statement) and voltage is the probe (used in the expressions for the contribution statement). This is for the case that both voltage and current properties are needed to accurately model the signal on this pin and the pin voltage is used as an independent variable (for example, driven by a voltage source).

K. XTALOSC_6M_ADAS1020 Netlist Hierarchy





Name	Bus Names	Pin Dir	Discipline	Type	Comments
agnd		input	logic	Gnd	
asub		input	logic	Gnd	
avdd1p8		input	logic	VSupply	
q		output	logic	Periodic_Signal	
rc_en		input	logic	Control	
rstb		input	logic	VBias	
trim_intosc\[7\]	trim_intosc[7:0]	input	logic	Gnd	
trim_intosc\[6\]	trim_intosc[7:0]	input	logic	Gnd	
trim_intosc\[5\]	trim_intosc[7:0]	input	logic	Gnd	
trim_intosc\[4\]	trim_intosc[7:0]	input	logic	Gnd	
trim_intosc\[3\]	trim_intosc[7:0]	input	logic	Gnd	
trim_intosc\[2\]	trim_intosc[7:0]	input	logic	Gnd	
trim_intosc\[1\]	trim_intosc[7:0]	input	logic	Gnd	
trim_intosc\[0\]	trim_intosc[7:0]	input	logic	Gnd	

Figure 13. Periodic Signal Pin Table In AranaBMP Pin Table

The Periodic_Signal type (figure 13) is only for the output pins, which have periodic waveforms, for example, the VCO circuit or the sine waveform generator circuit. There are Voltage_Source and Current_Source options for the Source Type field. The Voltage_Source will generate a periodic voltage source and Current_Source will generate a periodic current source.

Arana supports three types of waveforms: Sine, Square Wave and Custom in order to handle different types of waveforms efficiently. The Custom option allows you to generate any shape of periodic waveforms. However, if you already know the waveforms are sinusoidal or square, the Sine or Square Wave option can generate the behavioral models considering the specialty of sinusoidal and square wave.

The Output Average (Threshold) Value field requires you to provide the average value of the output signal. For example, if the output waveform varies from 0 to 3, then the average value is 1.5.

Max Value is the maximum value of the waveform. Min Value is the minimum value of the waveform. These two fields are mainly used in sinusoidal waveform generation. In the settings, the option User Specify Output Min and Max Value controls if the specified Max Value and Min Value are used as the maximum and minimum values for the periodic square waveform. If this option is set as ‘no’, Arana will automatically extract the minimum and maximum values for the square waveform.

Note that the periodic signal type is used mainly for modeling the periodic-signal generator. The only input pins allowed in this case are Large_Analog_Signal, which are used in the same way as multiple point bias. All the output pins must be of the same type Periodic-Signal.

L. *XTALOSC_6M_ADAS1020 Transistor vs. RNM Model Result*

TABLE 4

XTALOSC_6M_ADAS1020: Transisotr vs. RNM Model Result

Output Ports	**Transistor Result**	**RNM Result**
Clkout	6MHz	6MHz
Clkout_1meg	1MHz	1MHz
Clkout_500k	500kHz	500kHz

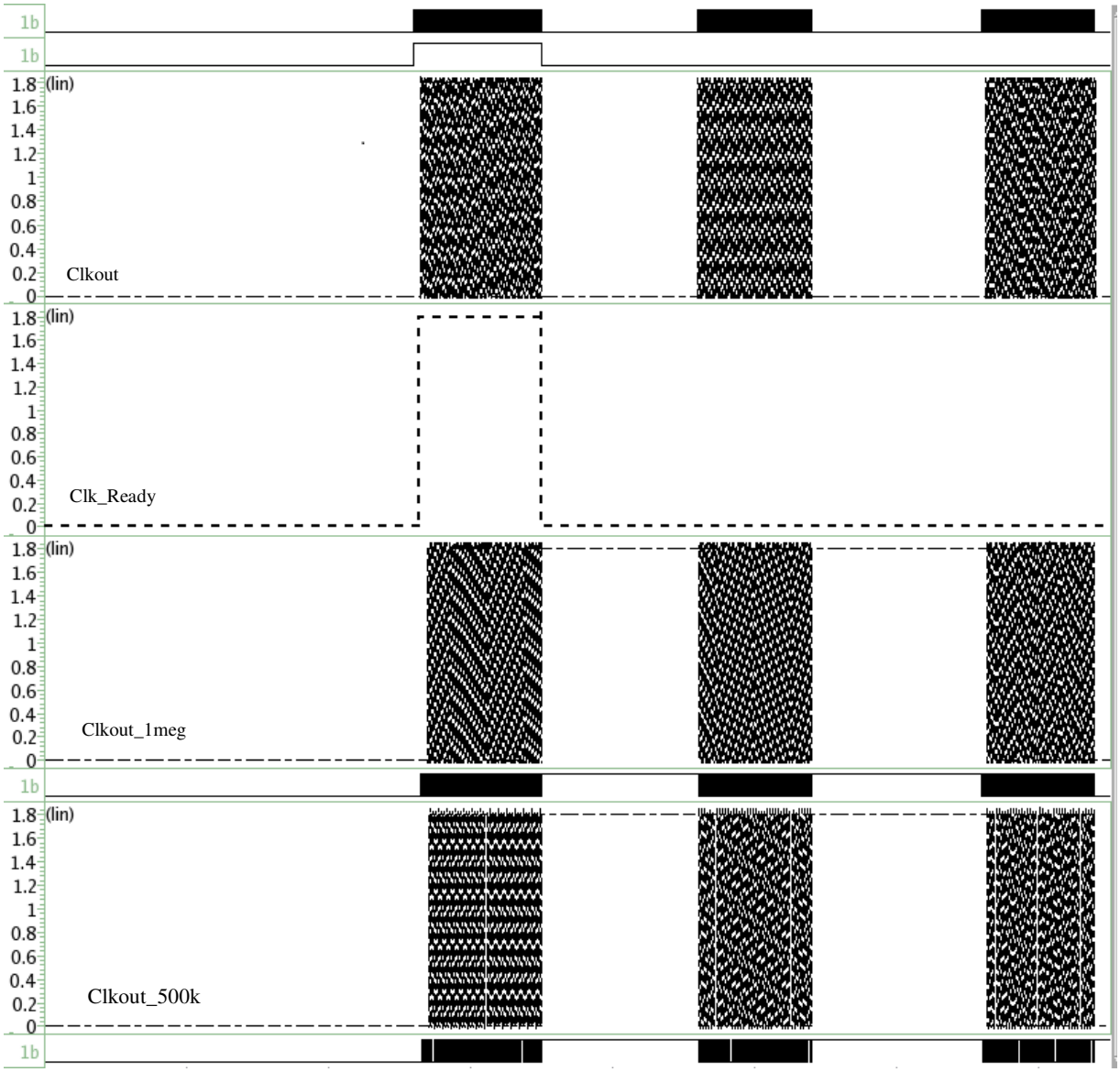
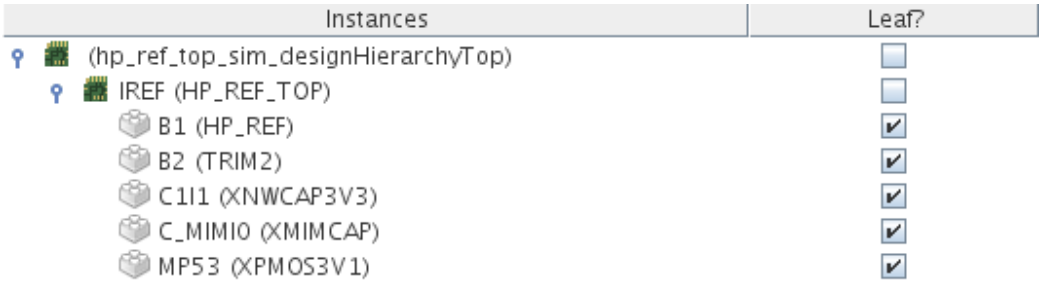


Figure 4. Oscillator Circuit Output- Transistor vs. Arana Abstracted RNM model

Transistor Result: - - - - - - -

Real Number Model Result: ———

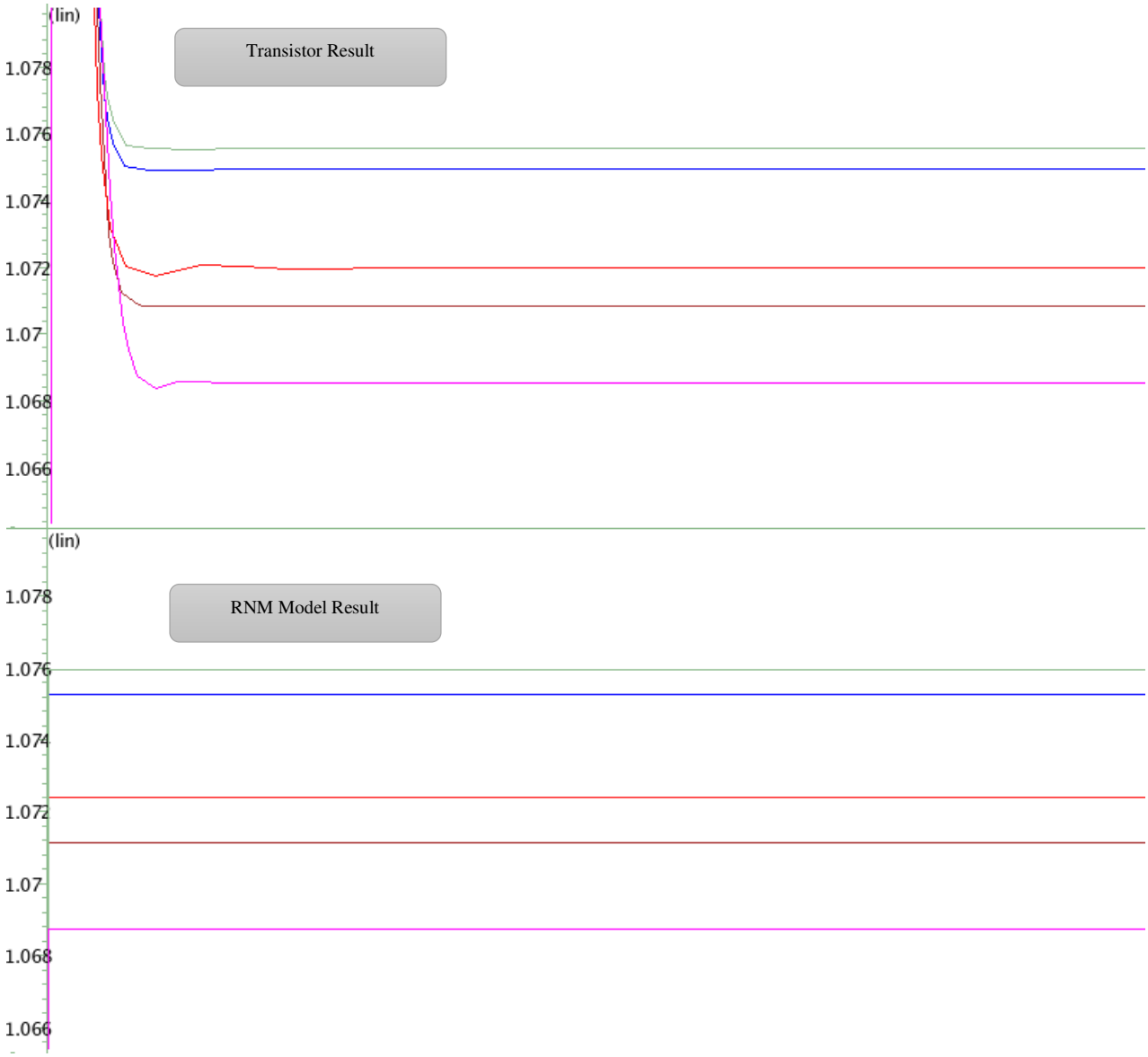
M. HP_REF_TOP Netlist Hierarchy



N. HP_REF_TOP: Transistor vs. RNM Model Results

TABLE 4
HP_REF_TOP: Transistor vs. RNM Model

Output	**Transistor Result**	**RNM Result**
Output1	1.0709v	1.0711v
Output2	1.0749v	1.0753v
Output3	1.072v	1.0724v
Output4	1.0756v	1.0759v
Output5	1.0685v	1.0687v



O. CPU Statistics:

ADI Circuits	Simulation Time Transistor	Simulation Time (NCverilog) Verilog-AMS Models	Simulation Time (VCS) Verilog-AMS Models	Simulation Speedup
SPLL_TOP	12h 10m 16s	5.3s	13.8s	~>8000x
SAR_ADC_12B	2h 25m 50s	1.6s	0.53s	~>14,400x
MSDRV_V	2m 28.4s	1.03s	1.03s	~>120x
XTALOSC_6M_ADAS1020	2h 46m 56s	126s	680s	~>60x
HP_REF_TOP	5.8s	0.3s	0.45s	~>20x

Contribution of this Work: The key innovation is the introduction of signal type driven abstraction. We have observed that all electrical nets in mixed-signal circuits can be classified to four groups:

- Soure, reference and biasing. Enable the circuit to function
- Data signal: the input and output signals represent information provided and processed
- Control, configuration and clocks: control the input and output data signal processing mode
- Feedthrough and loading: no functional purpose

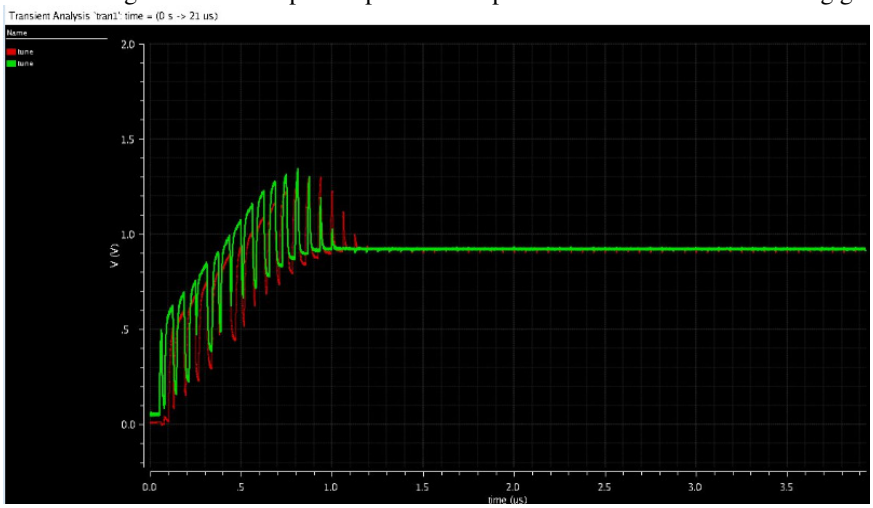
Furthermore, the data signals can be classified to four types only:

- Digital data signal
- Large-analog data signal
- Small-analog data signal
- Periodic-analog data signal

These types are captured by a pin table, which consists a specification provided by designers. These pin types can be represented in System-Verilog. All the circuit blocks that operate on these real value types can be automatically abstracted from schematics using topological tracing and model order reduction. A tool has been developed to generate System-Verilog, Wreal, or Verilog-A models from the schematic and pin table.

Results and Impacts: The methodology has been deployed on several recent mixed-signal SoC tapes out. The results of ADCXXX SoC product is summarized below. ADXXXX Soc is an analog front end for customers making 3PH power meters with current transformers or Rogowski coil sensors for high-performance ADC and metrology computation; It contains 7 channels Delta-Sigma modulator, 12-bt SAR-ADC, SPLL, bandgap, LDO, peak detector, and reference buffers. The methodology was used for all circuit blocks to generate System-Verilog and Verilog-A models.

Using the auto-generated models, several bugs were discovered. First-pass silicon functions correctly. One notable example was that a SPLL bug discovered in prior tape out from post-silicon was confirmed using generated models.



Vctrl voltage comparison between Schematic (red) vs RNM model (green)

Conclusions: A new signal-driven automatic abstraction methodology was presented to generate automatically schematic-consistent high-fidelity real value models for any analog and custom digital schematics. It has been deployed successfully on several ADI's SoC tapeouts, and uncovered several subtle design bugs, which were otherwise hard to catch. The automated abstracted models are 1000x or more times faster than circuit simulation while achiving the same analog function accuracy.