

- 1) Avoidable errors such as inverted or disconnected signals
- 2) Timing violations , synchronization

Analog and digital IP blocks at different levels of abstraction as schematics, SPICE netlists, analog behavioral models, or purely digital models. Functional analog RNM

- In digital design, the metric-driven verification approach, standardized for reusability as Universal Verification Methodology (UVM)1, helps engineers build confidence in the verification by increasing coverage to the desired level.
- On the analog side, verification is driven by directed tests run over sweeps, corners, and Monte Carlo analysis.
- Design failures including pin connection errors, inverted polarity, incorrect bus order, or pins connected to the wrong power domains. In the absence of simple checks, such errors are often found only in lengthy analog simulation runs, if they are found at all.

Analog behavioral modeling

To achieve reasonable simulation speeds, many mixed-signal teams employ analog behavioral modeling. This approach can be 5 to 100 times faster than SPICE. The actual speedup varies widely depending on the application and the level of detail in the model. Analog behavioral models are

Analog behavioral modeling

To achieve reasonable simulation speeds, many mixed-signal teams employ analog behavioral modeling. This approach can be 5 to 100 times faster than SPICE. The actual speedup varies widely depending on the application and the level of detail in the model. Analog behavioral models are typically written in one of the following languages:

- Verilog-AMS—A mixed-signal modeling language based on the IEEE 1364 Verilog standard that defines analog and digital behavior, and provides both continuous-time and event-driven modeling semantics
- Verilog-A—A continuous-time subset of Verilog-AMS, aimed at analog design
- VHDL-AMS—Similar in concept to Verilog-AMS, this language provides analog and mixed-signal extensions to IEEE 1076 VHDL
- SystemVerilog—This language is extended with real number types, such as IEEE 1800-2012 SV-DC extensions, also known as SV-RNM

Modeling Tradeoffs

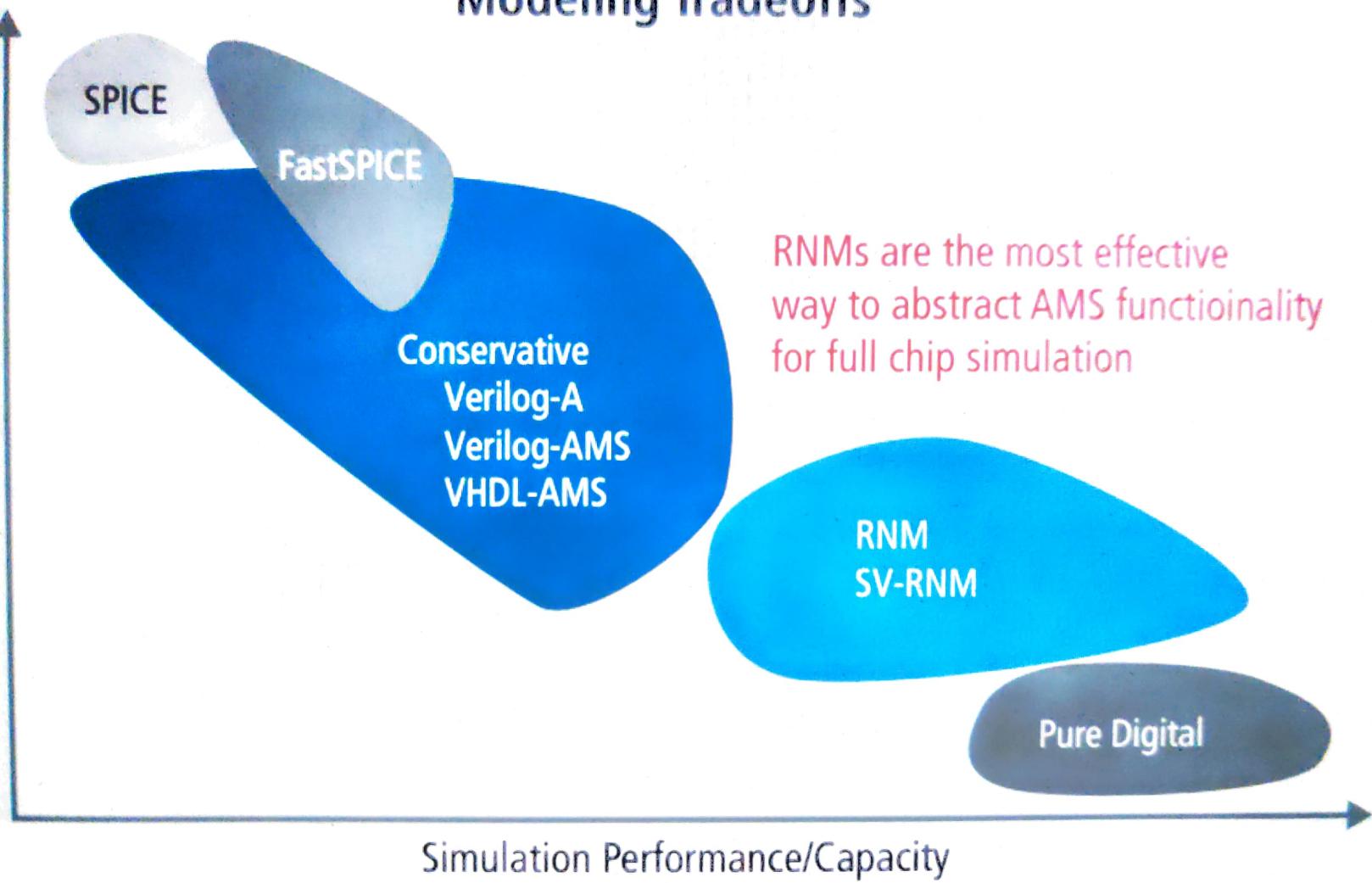


Figure 2: Model accuracy versus performance gain for mixed-signal simulation.

Real Number Modeling (**RNM**) is the process of modeling an analog circuit's behavior as signal flow model.

RNM is restricted to a signal flow approach, while analog convergence is less of an issue. Typically, it takes less modeling effort to develop RNM than traditional analog behavioral model.

A performance model must precisely capture critical circuit behavior. Functional models capture circuit behavior only to the level of detail that is needed to verify the correct design functionality.

Real number models are portable between digital (Incisive) and analog (Virtuoso) design environments. For example, a model can be developed and validated for an AMS block in Virtuoso, then used during SoC verification in Incisive.

UVM is a standardized structured methodology for testbench creation. UVM has coding guidelines and is used for building reusable verification components (UVCs) that consist of monitors and drivers.

Many languages support real and wire-real, or wreal, RNM, including Verilog, SystemVerilog, VHDL, and Verilog-AMS. The first four support a real data type, while **Verilog-AMS**

A performance model must precisely capture critical circuit behavior. Functional models capture circuit behavior only to the level of detail that is needed to verify the correct design functionality.

Real number models are portable between digital (Incisive) and analog (Virtuoso) design environments. For example, a model can be developed and validated for an AMS block in Virtuoso, then used during SoC verification in Incisive.

UVM is a standardized structured methodology for testbench creation. UVM has coding guidelines and is used for building reusable verification components (UVCs) that consist of monitors and drivers.

Many languages support real and wire-real, or wreal, RNM, including Verilog, SystemVerilog, VHDL, e, and Verilog-AMS. The first four support a real data type, while **Verilog-AMS** supports wreal.

Verilog real supports the following features:

- Module internal use of **real** variables
- No real value ports (requires **real2bits/bits2real**)
- No support for X/Z state
- No multiple ~~wreal~~ driver

VHDL **real** supports the following features:

- Real valued ports [www.cadence.com 8 Solutions for Mixed-Signal SoC Verification](http://www.cadence.com/8_Solutions_for_Mixed-Signal_SoC_Verification)
Real Number Models
- Resolution function
- Multiple drivers
- User-defined types
- Limited connection to ~~analog~~

~~Specman®~~ /e **real** supports the following features:

- Mainly for testbenching

Specman® /e real supports the following features:

- Mainly for testbenching
- Random generation, coverage, checking
- Direct access to analog values (receive/drive)

SV-DC (SV-RNM) supports the following features:

- User-defined types
- User-defined resolution functions
- Definition of a nettype based on its connectivity

I

Verilog-AMS wreal supports the following features:

- Easy interaction with analog
- Direct connection to electrical nets using E2R and R2E connect modules
- Disciplines association
- Multiple wreal driver support

- Multiple wreal driver support
- Ability for scope-based wreal resolution function specification
- Identification of high-impedance/unknown state (X/Z support)

SystemVerilog IEEE 1800-2009 standard supports the following features:

- Real number variables

I

- I/O real ports
- Assign statements to real variables
- SVA assertions including real variables

A major bottleneck in the adoption of RNM among SoC-level verification used to be the lack of support in SystemVerilog as of IEEE 1800-2009 standard. Some of the limitations of this standard included:

I

- No real number coverage or randomization
- No bidirectional real number ports (no in/out)
- Only one driver allowed on each port connection (no support for real number resolution functions for multiple drivers)
- No direct support for real valued signals
- No inter-language connections (Verilog-AMS, SPICE)

Due to these limitations, the SystemVerilog implementation of RNM required more code than wreal for the same implementation.

User-defined types (UDTs) – Allows for real valued nets – Allows for multi-value nets (multi-field record style)

- User defined resolution (UDRs) – Functions to resolve user-defined types – Association of function with user-defined nets
- Explicit Interconnects – Type-less nets – For connecting ports only – No continuous or procedural assignments allowed

LRM additions include:

- UDTs that can hold one or more real values – New keyword: nettype – UDTs can be used for port and net connections (unidirectional, bidirectional)
- UDR function for combining UDTs – Function associated to nettype using keyword: with
- Type-less interconnects

- User defined resolution (UDRs) – Functions to resolve user-defined types – Association of function with user-defined nets

- Explicit Interconnects – Type-less nets – For connecting ports only – No continuous or procedural assignments allowed

LRM additions include:

- UDTs that can hold one or more real values – New keyword: `nettype` – UDTs can be used for port and net connections (unidirectional, bidirectional)
- UDR function for combining UDTs – Function associated to `nettype` using keyword: `with`
- Type-less interconnects between nets and ports – New keyword: `interconnect` For users using Verilog-AMS `wreal`, Cadence has built-in `wreal nettypes`, such as `wrealsum`, `wrealavg`, `wrealmin`, and `wrealmmax`, that enable Verilog-AMS code reuse and ease the migration to SystemVerilog.

Data Type and Resolution Function

Model

Nettype

```
// user-defined data type T
typedef struct (
    real field;
    bit field2;
) T;                                UDT

//user-defined resolution
// function Tsum
function automatic T Tsum
    (input T driver []);

    Tsum.field1 = 0.0;
    foreach (driver [i])
        Tsum.field1 += driver[i].field1;

endfunction                         UDR
```

```
// a nettype wTsum whose data type
// is T and resolution function
// is Tsum
nettype T wTsum with Tsum

// SV module using UDT port
// Notice: Although "foo_p"
// has 2 values in its datatype,
// there is only 1
// port in this module
module foo (foo_p);
    wTsum foo_p;
    input foo_p;
    wTsum q; // an internal net

    assign q = foo_p;
endmodule

module bar (bar_p)
    wTsum bar_p;
    output bar_p;
    assign bar_p = T'(14.5., 1'bl);
endmodule
```

Figure 9: SV-RNM nettype illustrations.

What is Verilog AMS?

A single unified language with both analog and digital semantics.

Initial, always, and analog procedural blocks can appear in the same module — both analog and digital signal values can be accessed (read operations) from any context (analog or digital) in the same module.

A new construct, connect statement, is added to facilitate auto-insertion of user-defined connection modules between the analog and digital domains.: Analog port, Digital port, mixed-port.

Analog/mixed signal associated with a node(only one value allowed), digital signal not associated with a node.

Conservative systems ::There are two values associated with every node, **potential** and **flow**.

A branch is a path of flow between two nodes through a component. Every branch has an associated potential (the potential difference between the two nodes) and flow.

A discipline with only a potential nature or only a flow nature is known as a **signal flow discipline**.

Disciplines with two natures are known as **conservative disciplines**.