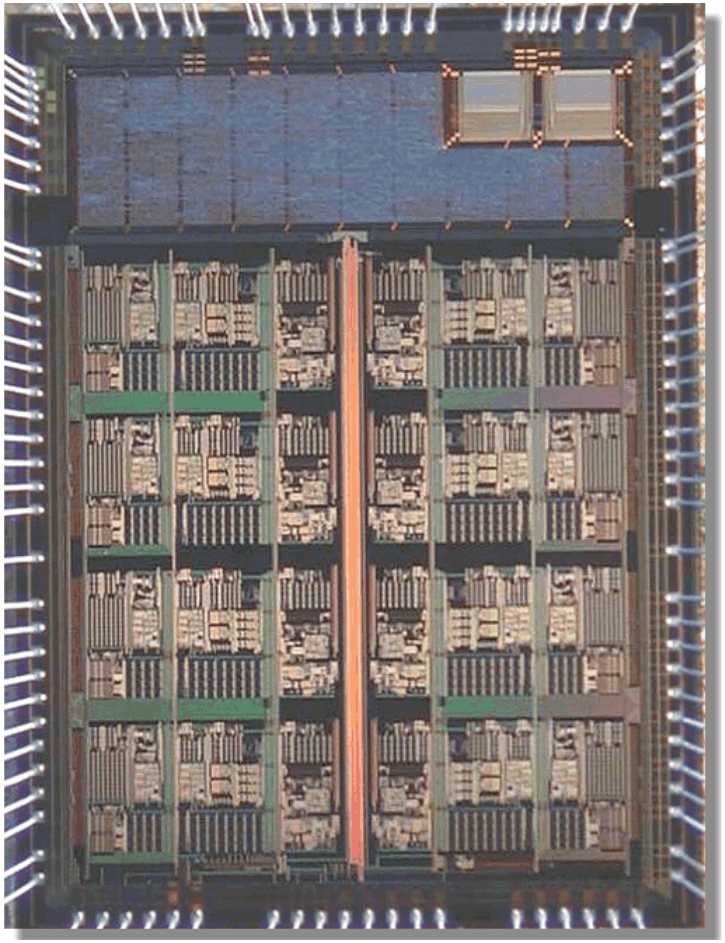
Guidelines for Verilog-A Compact Model Coding

Gilles DEPEYROT,
Frédéric POULLET, Benoît DUMAS
DOLPHIN Integration

Outline

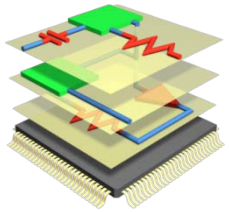
- Dolphin
  - EDA Solutions by Dolphin
  - Overview of SMASH
- Context & Goals
- Verilog-A for Compact Modeling
  - Benchmark of Verilog-A vs. SPICE
  - Verilog-A Limitations (for Compact Modeling)
- Recommendations
  - Subset of Verilog-A for Compact Models
  - Focus on Spice integration
- Conclusion and Perspectives

Dolphin Integration

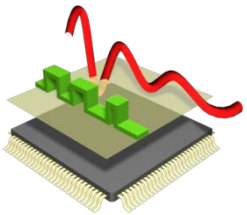


- **A strong synergetic potential between**
  - **Silicon Intellectual Property Products**
    - Embedded memories - SRAM, ROM
    - Standard cell libraries, low power, high density
    - Mixed signal - ADC, DAC
    - Power management
    - 8b, 16b microcontrollers
  - **EDA Solutions**
  - **Delegation Services**
    - Four professions in growing demand for design in microelectronics
  - **Turnkey SoC Design**
    - low power, mixed signal ASICs
- **Since 1985, now 180 including 145 engineers**
- **On Alternext stock-market**
- **11 M€ sales turnover**

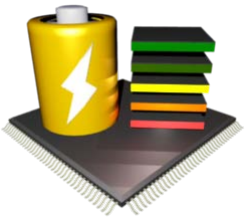
EDA Solutions by Dolphin



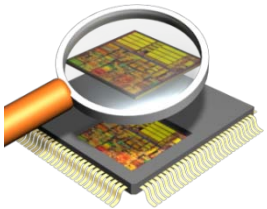
SLED – Schematic Link Editor



SMASH™ – Mixed-Signal Multi-Language Simulator



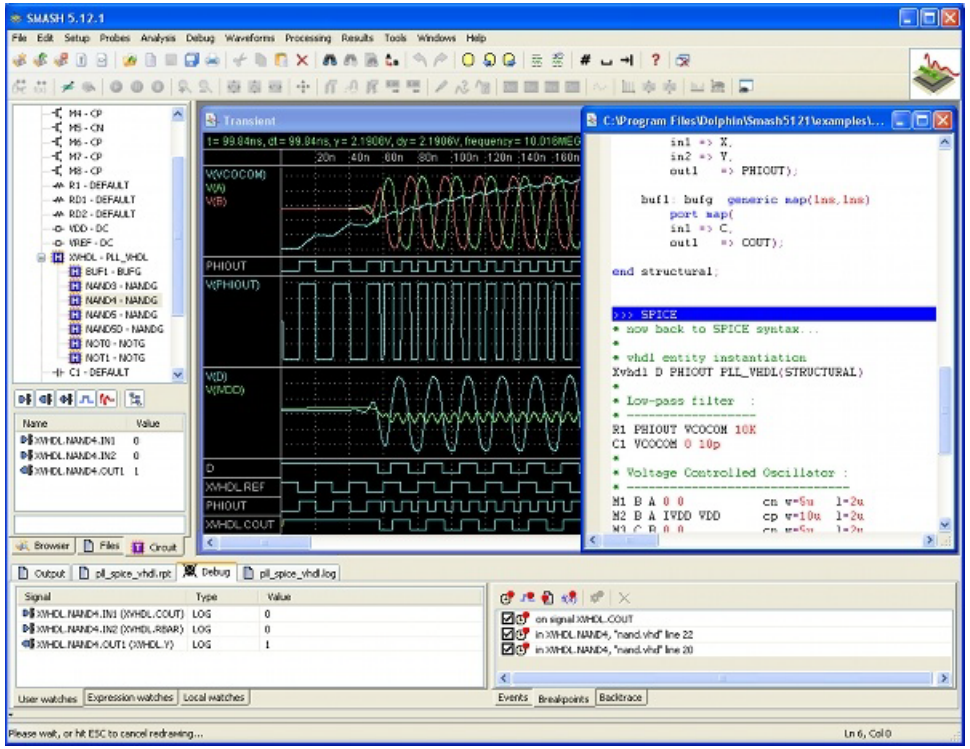
SCROOGE – Mixed-Signal Power Consumption Estimator
– Powered by SMASH™



SoC GDS – Layout Analyzer & Processor

Overview of SMASH

- Mixed-signal
  - Analog
  - Logic
- Multi-language
  - SPICE (including flavors)
  - Verilog
  - VHDL
  - Verilog-A
  - VHDL-AMS
  - C
- Multi-level
  - Structural / Gate
  - RTL
  - Behavioral
- Multi-platform
  - Windows & Linux

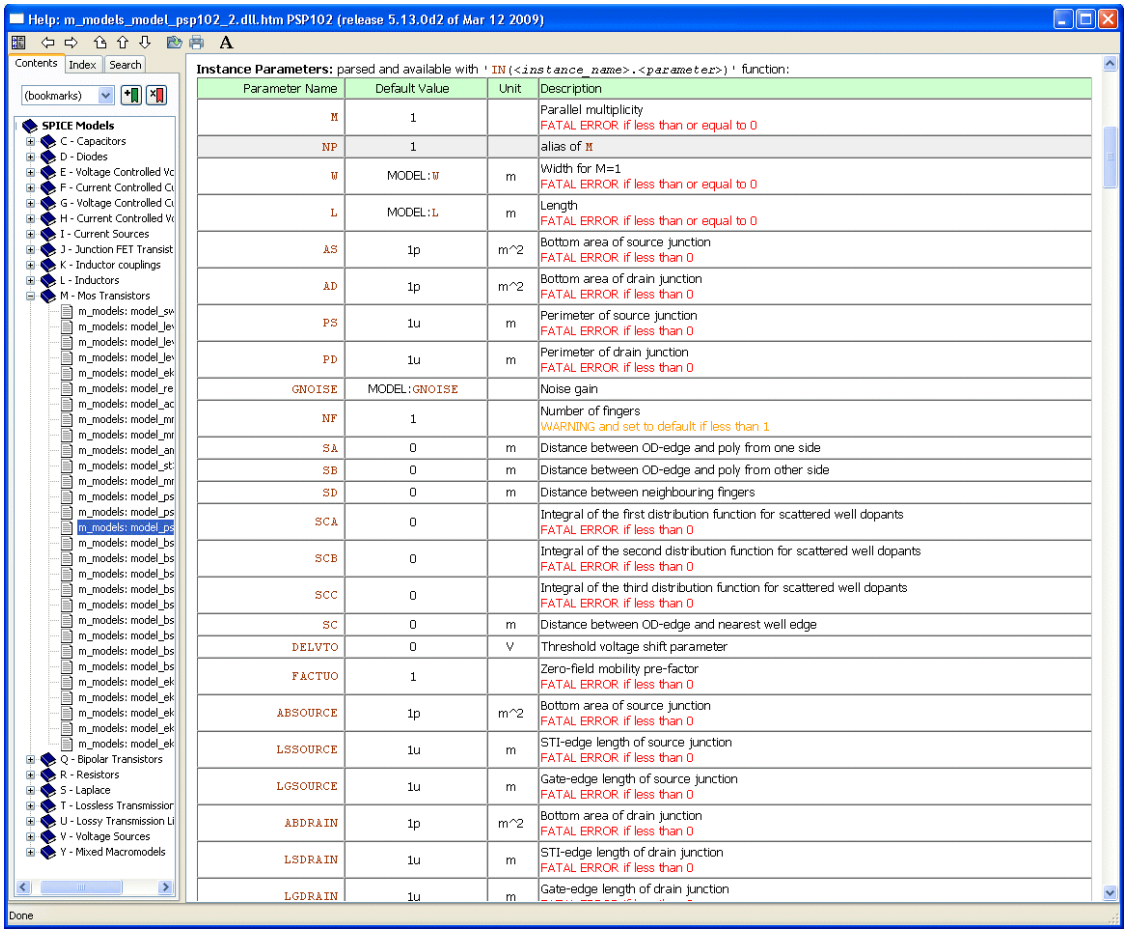


- Wide range of integrated Compact Models
  - BSIM3, BSIM4, EKV2.6, EKV3, ACM, PSP, VBIC, HICUM, MEXTRAM, MM9…

- Since 1989...

Dolphin's Experience Integration of Compact Models

- …since 1989
- More than 85 SPICE and mixed models
- 29 MOS models, 5 bipolar models, 5 diode models, 2 JFET models
- SPICE flavor handling
  - compatibility with competitor specificities (HSPICE, ELDO, PSPICE…)
- Hierarchical approach
  - Diode or resistor models called by MOS models
  - Common parameters or functionalities (geometry, diode, noise, matching, documentation…)
- C interface
  - BSIM4, BSIM3, MM11, MM9, EKV2…
- Verilog-A with ADMS XML
  - PSP, EKV3, Juncap2, HiCUM



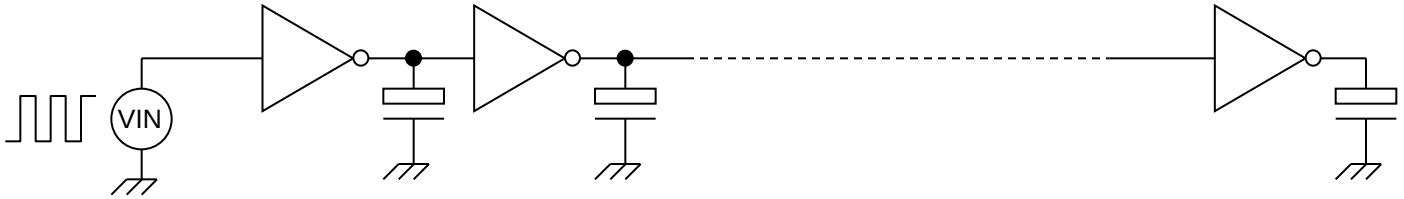
Instance Parameters: parsed and available with 'IN(<instance_name>.<parameter>)' function:

Parameter Name	Default Value	Unit	Description
M	1		Parallel multiplicity FATAL ERROR if less than or equal to 0
NP	1		alias of M
W	MODEL:W	m	Width for M=1 FATAL ERROR if less than or equal to 0
L	MODEL:L	m	Length FATAL ERROR if less than or equal to 0
AS	1p	m^2	Bottom area of source junction FATAL ERROR if less than 0
AD	1p	m^2	Bottom area of drain junction FATAL ERROR if less than 0
PS	1u	m	Perimeter of source junction FATAL ERROR if less than 0
PD	1u	m	Perimeter of drain junction FATAL ERROR if less than 0
GNOISE	MODEL:GNOISE		Noise gain
NF	1		Number of fingers WARNING and set to default if less than 1
SA	0	m	Distance between OD-edge and poly from one side
SB	0	m	Distance between OD-edge and poly from other side
SD	0	m	Distance between neighbouring fingers
SCA	0		Integral of the first distribution function for scattered well dopants FATAL ERROR if less than 0
SCB	0		Integral of the second distribution function for scattered well dopants FATAL ERROR if less than 0
SCC	0		Integral of the third distribution function for scattered well dopants FATAL ERROR if less than 0
SC	0	m	Distance between OD-edge and nearest well edge
DELVTO	0	V	Threshold voltage shift parameter
FACTUO	1		Zero-field mobility pre-factor FATAL ERROR if less than 0
ABSOURCE	1p	m^2	Bottom area of source junction FATAL ERROR if less than 0
LSSOURCE	1u	m	STI-edge length of source junction FATAL ERROR if less than 0
LGSOURCE	1u	m	Gate-edge length of source junction FATAL ERROR if less than 0
ABDRAIN	1p	m^2	Bottom area of drain junction FATAL ERROR if less than 0
LSDRAIN	1u	m	STI-edge length of drain junction FATAL ERROR if less than 0
LGDRAIN	1u	m	Gate-edge length of drain junction

Context & Goals

- What needs to be done so that Verilog-A can become the standard for CM coding?
  - Benchmarking performed to understand current status and provide guidelines
  - Guidelines put together for CM coding
- What is at stake?
  - Fully taking into account SPICE-like integration of Verilog Compact Models in the ecosystem
  - Providing a viable and open alternative to “controlled” initiatives (such as TMI or CMI)

Benchmark of Verilog-A vs. SPICE Conditions



Test bench:

- Configurable CMOS delay (400, 4k or 40k MOS)
- Use default values for the parameters of the MOS models
- Use two models, one PMOS and one NMOS
- Computed iterations 2550±5
- Use TRAP method for integration

Benchmark of Verilog-A vs. SPICE Memory

× 3

Memory usage (Mb)		SMASH 5.15*		Simulator B		Ratio Verilog-A / SPICE
		SPICE	Verilog-A	SPICE	Verilog-A	
PSP Model	Circuit #1	40Mb	51Mb	**15Mb**	**18Mb**	1.20
	Circuit #2	57Mb	115Mb	**47Mb**	**97Mb**	2.06
	Circuit #3	**216Mb**	**633Mb**	330Mb	854Mb	2.93
EKV3 Model	Circuit #1	**39Mb**	51Mb	NA	**18Mb**	0.46
	Circuit #2	**51Mb**	116Mb	NA	**66Mb**	1.29
	Circuit #3	**170Mb**	807Mb	NA	**540Mb**	3.18

* The SMASH graphic user interface consumes 34Mb out of the total memory consumption.

Benchmark of Verilog-A vs. SPICE Loading Time

× 5

Loading time (seconds)		SMASH 5.15		Simulator B		Ratio Verilog-A / SPICE
		SPICE	Verilog-A	SPICE	Verilog-A	
PSP Model	Circuit #1	0.25s	1.00s	**0.06s**	**0.75s**	12.5
	Circuit #2	0.40s	2.28s	**0.25s**	**1.75s**	7
	Circuit #3	**2.60s**	25.29s	3.24s	**10.75s**	4.1
EKV3 Model	Circuit #1	**0.23s**	0.51s	NA	**0.20s**	0.9
	Circuit #2	**0.44s**	1.93s	NA	**0.74s**	1.7
	Circuit #3	**2.34s**	21.40s	NA	**11.87s**	5.1

Benchmark of Verilog-A vs. SPICE Operating-Point Time

× 10

Operating point time (seconds)		SMASH 5.15		Simulator B		Ratio Verilog-A / SPICE
		SPICE	Verilog-A	SPICE	Verilog-A	
PSP Model	Circuit #1	0.23s	0.40s	**0.02s**	**0.20s**	10.00
	Circuit #2	2.17s	15.8s	**0.17s**	**2.30s**	13.53
	Circuit #3	28.9s	6382s	**2.34s**	**21.17s**	9.05
EKV3 Model	Circuit #1	**0.17s**	**0.47s**	NA	0.48s	2.76
	Circuit #2	**1.22s**	90.51s	NA	**5.53s**	4.53
	Circuit #3	**26.3s**	Too Big	NA	**73.06s**	2.78

Benchmark of Verilog-A vs. SPICE Transient Speed

× 15

Simulation time (seconds)		SMASH 5.15		Simulator B		Ratio Verilog-A / SPICE
		SPICE	Verilog-A	SPICE	Verilog-A	
PSP Model	Circuit #1	**1.1s**	**17.7s**	2.61s	30.1s	16.1
	Circuit #2	**17.2s**	**249.7s**	29.95s	416.8s	14.5
	Circuit #3	**206.9s**	**3384s**	284.6s	8 822s	16.4
EKV3 Model	Circuit #1	**2.43s**	39.7s	NA	**31.8s**	13.1
	Circuit #2	**31.3s**	594.9s	NA	**351.8s**	11.2
	Circuit #3	**372.7s**	Too Big	NA	**10 197s**	27.4

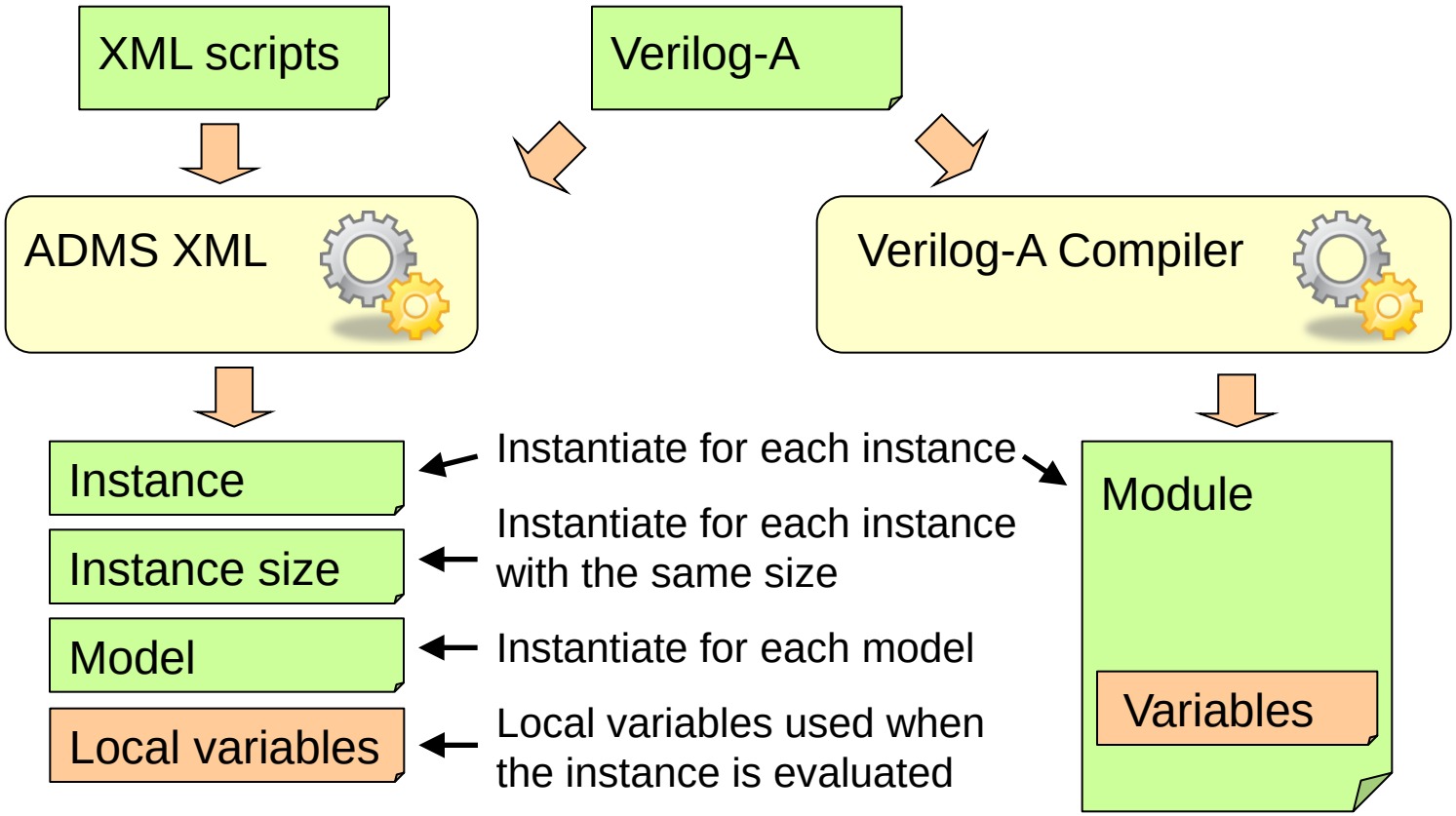
Benchmark of Verilog-A vs. SPICE Summary

Verilog-A vs. SPICE	Ratio
Memory Consumption	3
Loading Time	5
Operating-Point Time	10
Transient Speed	15

The order of magnitude of the ratio is what we are looking at.

Verilog-A Limitations
Memory Consumption

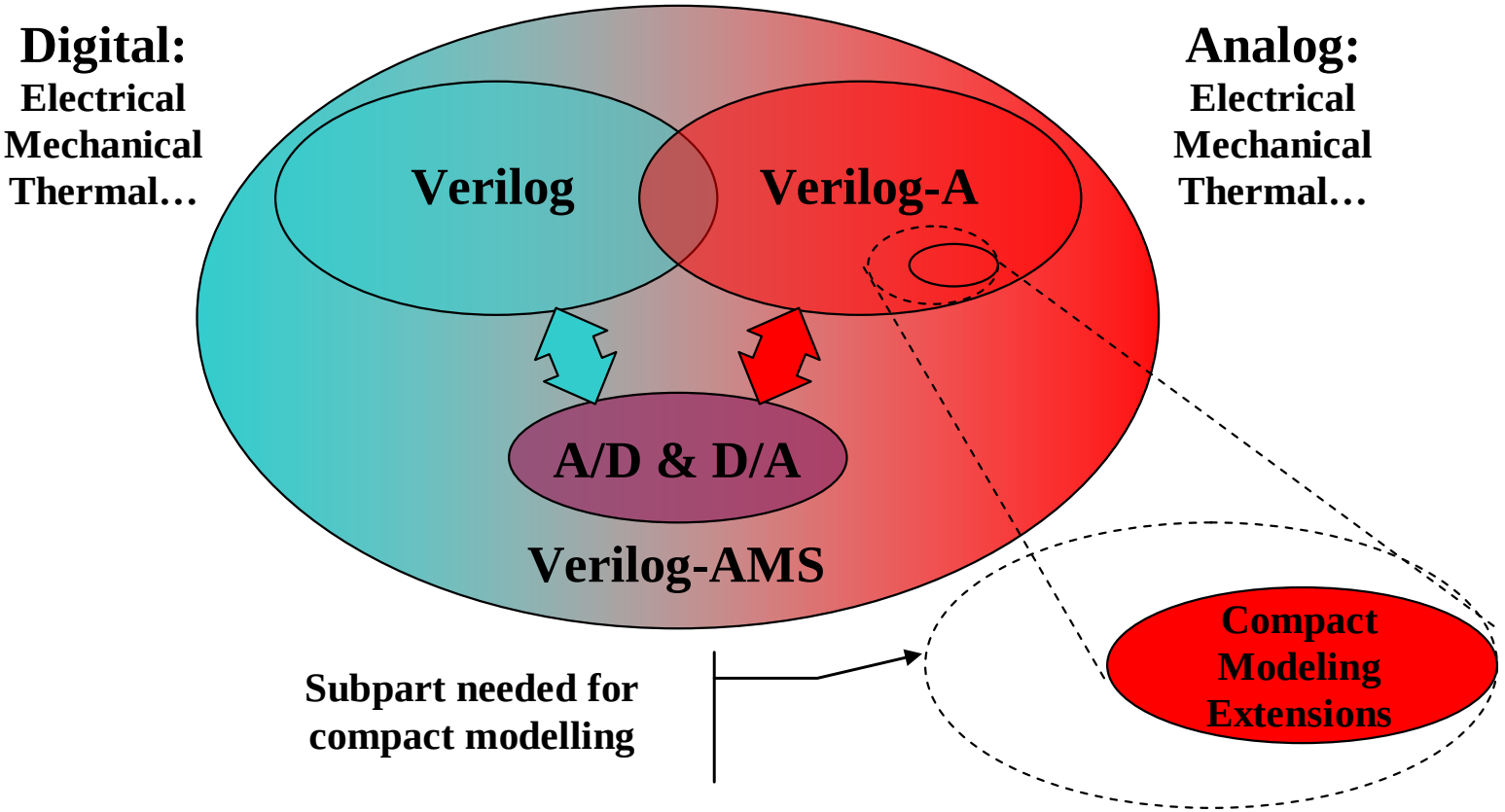
Both approaches generate C code from the Verilog-A model



Verilog-A Limitations
Simulation Speed

- Collapsible nodes: used, for instance, to collapse nodes when access resistances are not created.
- Bypass/linearization: for small variations, the SPICE simulator replaces the compact model by a linear model which is far faster.
- Derivation/integration: which requires one additional node.
- Specific code: at each iteration of the transient analysis, the Verilog-A simulator executes the code corresponding to the model/instance initialization, noise computation or temperature adaptation, while the SPICE simulator does not.
- Hidden states: for variables that depend on the previous point and output variables, and at each iteration of the transient analysis, the Verilog-A simulator initializes the variables with the previous value, while the SPICE simulator does not.

Recommendations
Subset of Verilog-A



Recommendations
SPICE specificities

1. To reduce human interventions and the associated risks of having different behaviors in different simulators
   - Facilitate an efficient conversion for integration into different SPICE simulators
2. To reduce the memory footprint
   - Load several tens of thousands of transistors in a conventional SPICE simulator
3. To reduce the simulation time
   - It is of critical importance for the analog designer that the compact models run as quickly as possible in the SPICE simulator

Proposal
Collapsible Nodes

- Collapsible nodes have the benefit that they reduce the size of the system matrix
  - Collapsible nodes should be defined during the instantiation phases.
  - Two ports, or one port and the ground, should not be collapsed as, in general, this will be implemented as an extra node.

- The Verilog-AMS 2.3.1 LRM does not yet specify the syntax and conditions for collapsing nodes. A commonly used “idiom” is:

```
if (r/$mfactor < 1.0e-3)
    V(a,b) <+ 0.0;
else
    I(a,b) <+ V(a,b) / r;
```

- **Proposal**: use a syntax based on attributes, automatically handled during conversion of Verilog-A models into SPICE models, for instance when using ADMS XML:

```
(* collapse = "r/$mfactor < 1.0e-3" *) electrical a, b;
```

Conclusion

- For the moment, SPICE simulators remain faster than their Verilog-A counterparts.
  - Compact Models in Verilog-A should target SPICE simulators and respect the inherent constraints to facilitate their integration into different SPICE simulators
- EDA vendors will fill the performance gap between SPICE and Verilog-A simulators. Therefore, the time is coming to:
  - Make Verilog-A more attractive than SPICE for semiconductor foundries as well as for final users
  - Compete with the “Standard Model API” to address the problems of deep submicron processes such as dynamic degradation, power consumption, system-level complexity…

This work is partially supported by the European Commission FP7 under contract number 218255 (COMON).

SEVENTH FRAMEWORK PROGRAMME

Thanks!

MOS-AK / GSA
GSA Modeling Working Group
http://www.mos-ak.org

COMON
The Compact Modelling Network
http://compactmodelling.eu