MOS-AK / GSA Workshop
IHP, Frankfurt (Oder)

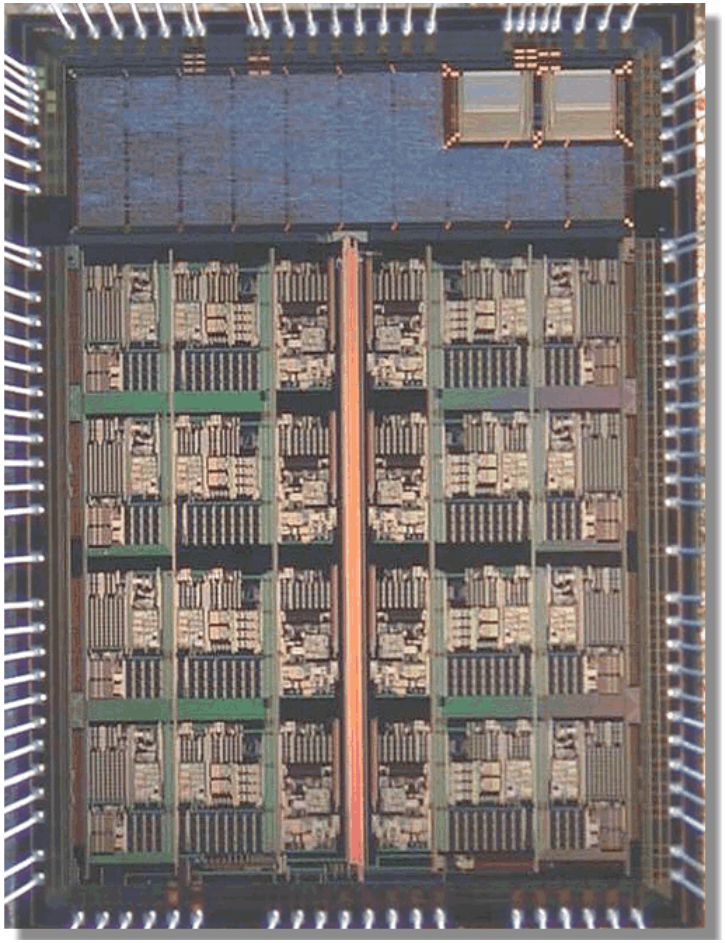
Need for
A standard subset of Verilog-A for Compact Modeling

Gilles DEPEYROT & Frédéric POULLET
DOLPHIN Integration

Outline

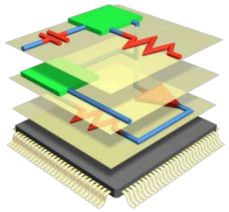
- Dolphin
  - EDA Solutions by Dolphin
  - Overview of SMASH
- Verilog-A for Compact Modeling
  - Next step: direct use by designers
  - Benchmark of Verilog-A vs. SPICE
  - Verilog-A Limitations (for Compact Modeling)
- Verilog-A Compact Models in SMASH
  - Examples of additions during integration
- Need for a standard subset of Verilog-A
- Conclusion

Dolphin

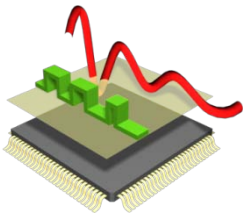


- **A strong synergetic potential between**
  - **Silicon Intellectual Property Products**
    - Embedded memories SRAM, ROM
    - Standard cell libraries, low power, high density
    - Mixed signal: ADC, DAC
    - Power management
    - 8b, 16b microcontrollers
  - **EDA Solutions**
  - **Dolphin Delegation Services**
    - Five professions in growing demand for design in Microelectronics
  - **Turnkey SoC Design**
    - low power, mixed signal ASICs
- **Since 1985, now 175 including 145 engineers**
- **On Alternext stock-market**
- **11.5 M€ sales turnover**

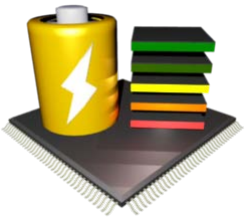
EDA Solutions by Dolphin



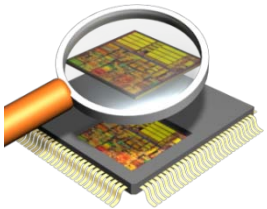
SLED – Schematic Link Editor



SMASH™ – Mixed-Signal Multi-Language Simulator



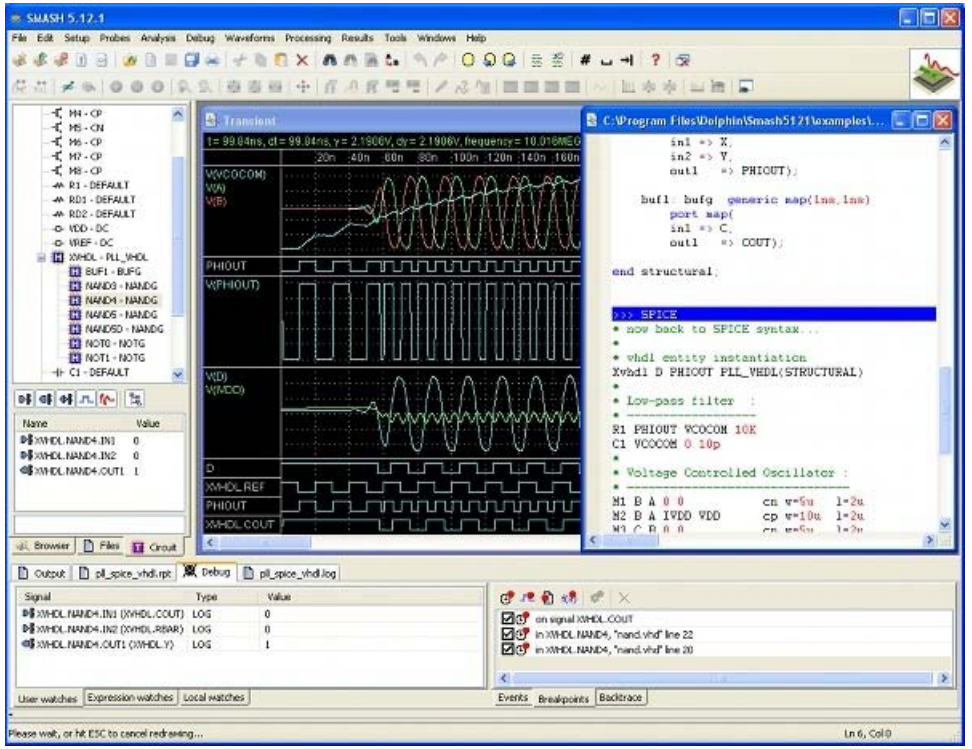
SCROOGE – Mixed-Signal Power Consumption Estimator
– Powered by SMASH™



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 & Solaris

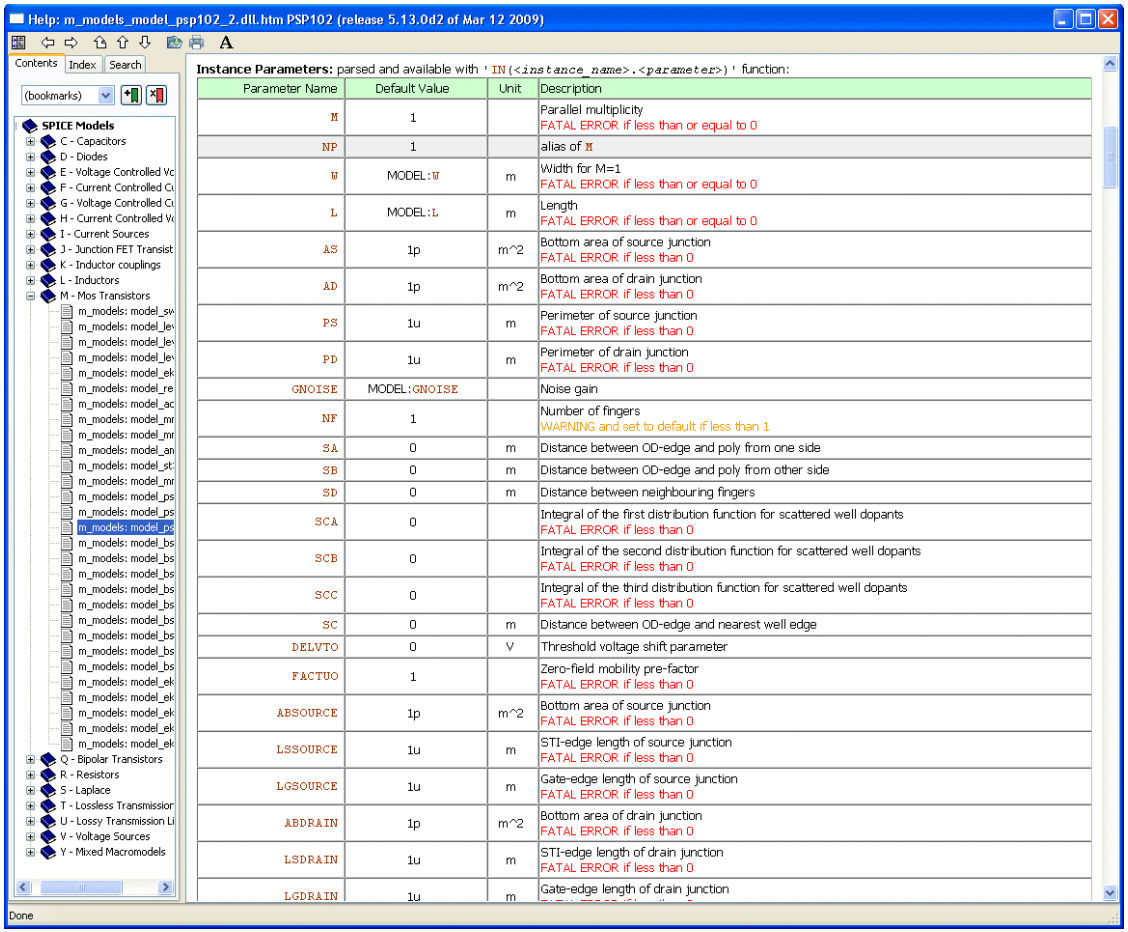


- Wide range of integrated Compact Models
  - BSIM3, BSIM4, EKV2.6, EKV3, ACM, PSP, VBIC, MEXTRAM, MM9…

- Since 1989...

Dolphin's Experience Integration of Compact Models

- More than 85 device models
  - Hierarchical approach
- SPICE flavor handling
  - Compatibility with competitor specificities (HSPICE, ELDO, PSPICE…)
- Partnerships
  - EKV2, ACM, EKV3…
- C interface
  - BSIM3, BSIM4, MM9, MM11, 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

Verilog-A for Compact Modeling

- Advantages of Verilog-A
  - Standardized
  - More viable than C (with simulator specific interfaces)

Modeler

- Standard language
- No simulator specific interface
- Derivations handled automatically
- Fewer bugs in models (derivations, parameter default values…)

Integrator

- Standard language
- Fewer bugs introduced during integration
- Simplified integration
- Leeway for introducing proprietary additions
- Faster availability of new Compact Models

Designer/Foundry

- Implementation quasi-identical in every simulator
- Faster availability of new Compact Models in a wider range of simulators

Next step: direct use of Verilog-A models instead of SPICE models

- Directly distributing Verilog-A models would have several advantages

Modeler/Integrator

- No integration needed, direct distribution to foundry and final users

⚠

- Must be compatible with existing SPICE level design kits, schematics, and netlists

Foundry

Use a standard language for:

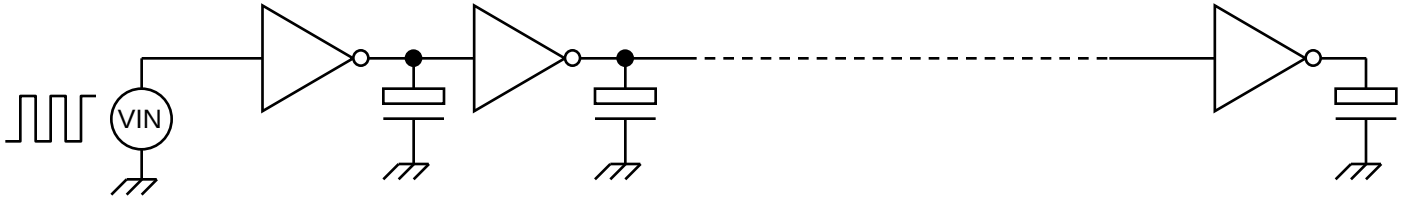
- Customizing model features (replacing TMI and CMC API)
- Adding parasitic behavior
- Describing statistical models
- Describing macro-models

Designer

- Identical in every simulator
- Faster availability of new models and design kits in a wider range of simulators and foundries

➡ Based on Dolphin's experience, we will present the issues that must be addressed to reach this goal

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

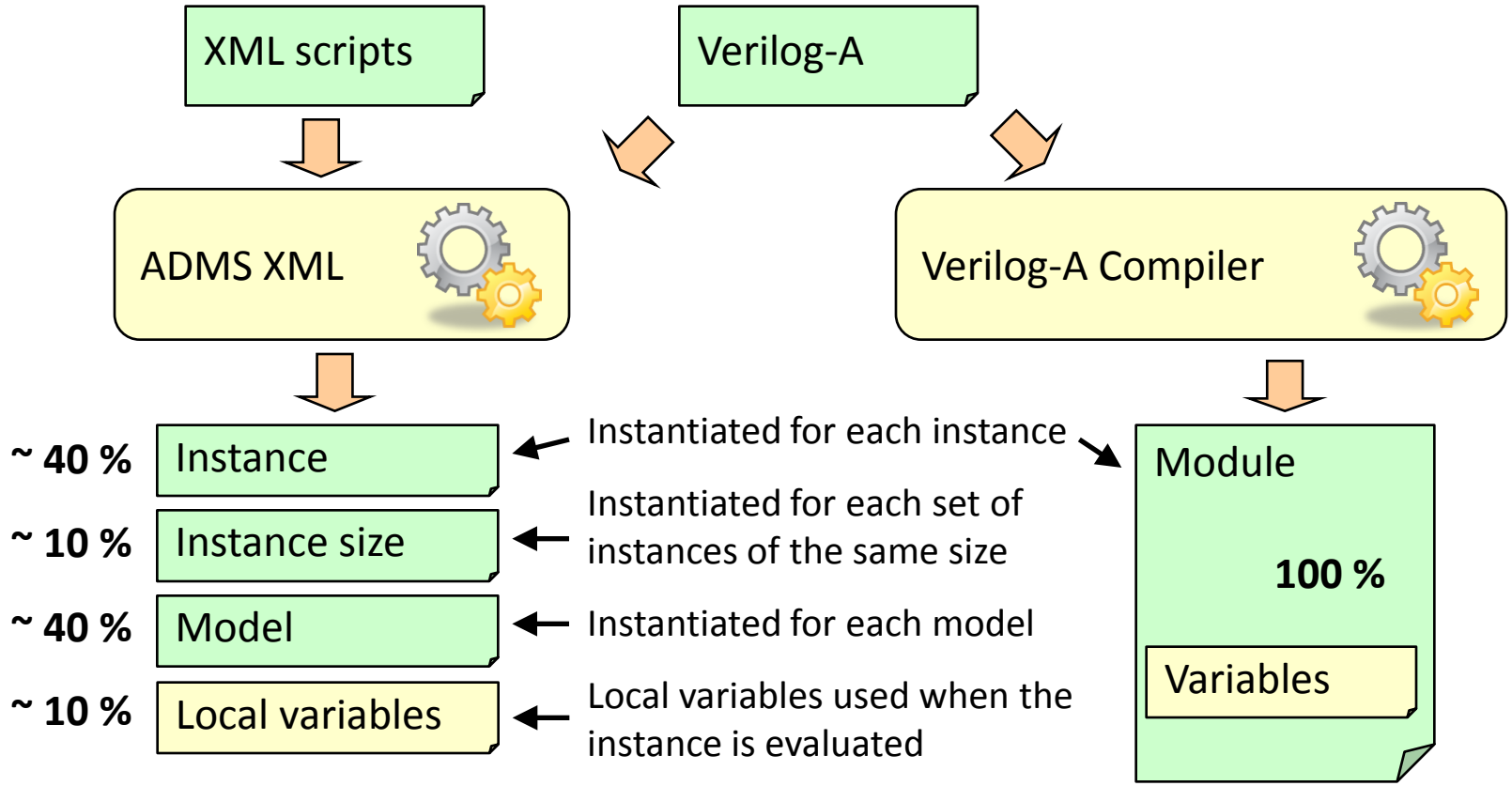
Memory usage (MB)		SMASH		Simulator B		Ratio Verilog-A / SPICE
		SPICE	Verilog-A	SPICE	Verilog-A	
PSP Model	Circuit #1 400 MOS	38 MB	75 MB	15 MB	19 MB	1.2
	Circuit #2 4k MOS	55 MB	407 MB	65 MB	96 MB	1.7
	Circuit #3 40k MOS	204 Mb	×	566 MB	854 MB	4.1
EKV3 Model	Circuit #1 400 MOS	39 MB	68 MB	NA	15 MB	0.4
	Circuit #2 4k MOS	51 MB	305 MB	NA	62 MB	1.2
	Circuit #3 40k MOS	166 MB	×	NA	436 MB	2.7

Benchmark of Verilog-A vs. SPICE Speed

Simulation time (seconds)		SMASH		Simulator B		Ratio Verilog-A / SPICE
		SPICE	Verilog-A	SPICE	Verilog-A	
PSP Model	Circuit #1 400 MOS	1.2 s	32.4 s	3.1 s	28.5 s	23.1
	Circuit #2 4k MOS	17.7 s	675.4 s	40.3 s	405.1 s	22.9
	Circuit #3 40k MOS	216.5 s	×	421.1 s	8565.8 s	39.6
EKV3 Model	Circuit #1 400 MOS	2.1 s	47.7 s	NA	30.1 s	14.2
	Circuit #2 4k MOS	30.5 s	872.9 s	NA	339.2 s	11.1
	Circuit #3 40k MOS	375.2 s	×	NA	6676.7 s	17.8

Verilog-A Limitations
Memory Consumption

- Model / Instance size / Instance

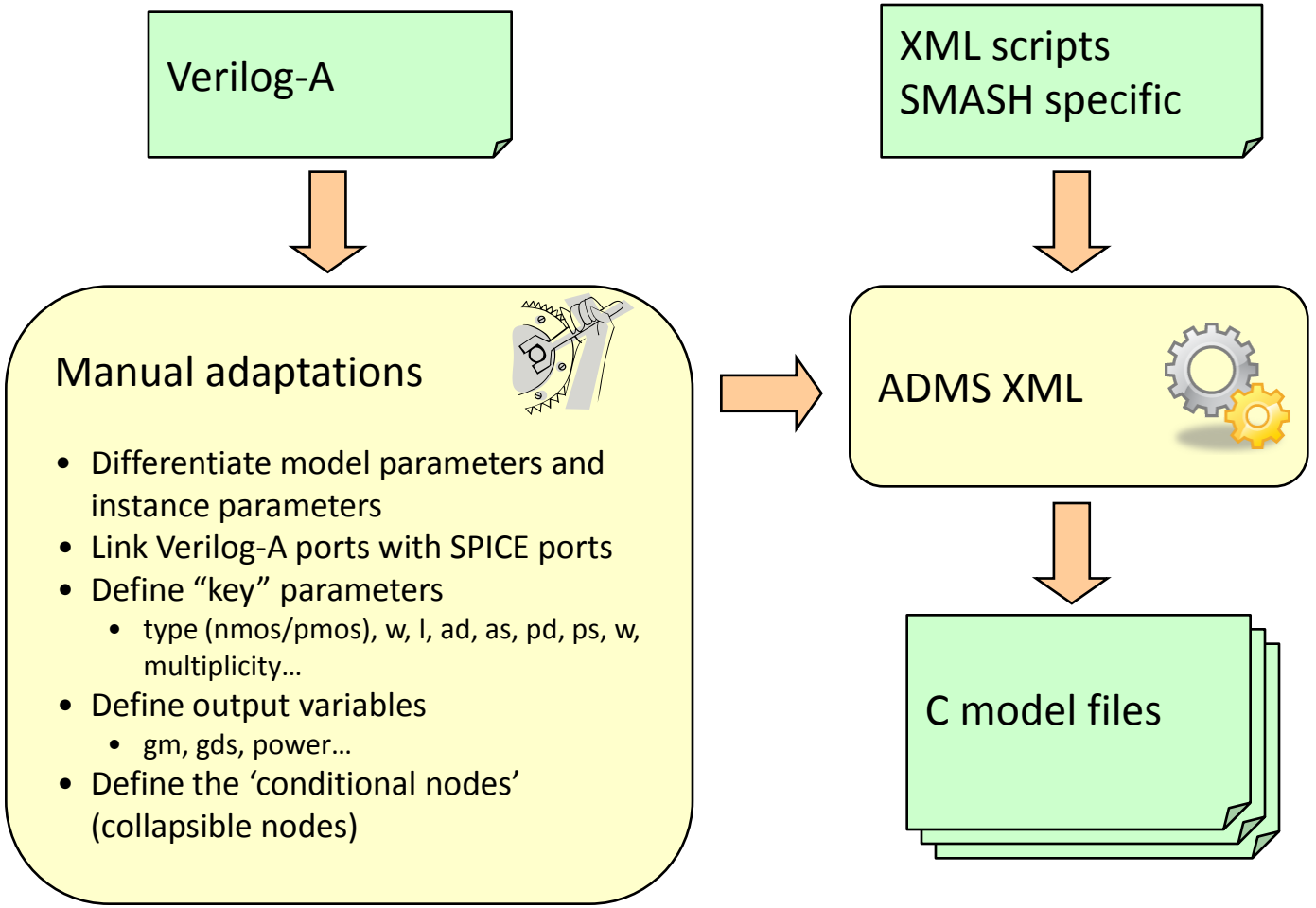


Verilog-A Limitations
Simulation Speed

- Implementation dependent
  - Flow/Potential branches
  - Derivation/Integration
  - Bypass/Linearization
  - Iteration specific code vs. specific code (initialization, noise)
  - Hidden states
  - Extra nodes
    - added for correlated noise due to ADMS XML limitation
- Language (or coding) standard dependent
  - Iteration specific code vs. specific code (output variables)
  - Conditional nodes (collapsible nodes)

1 to 85 x

Verilog-A Compact Models in SMASH Integration

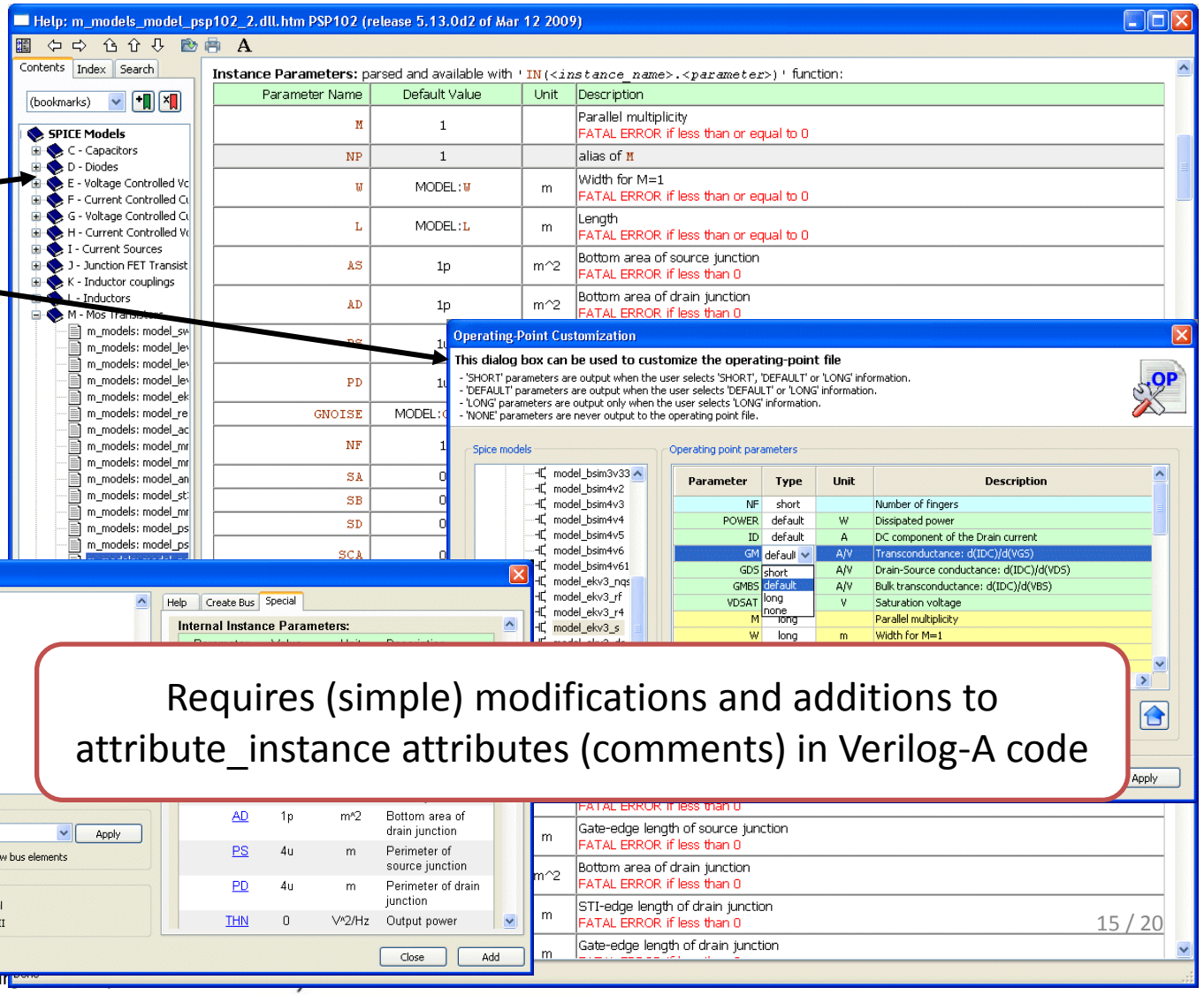


- Differentiate model parameters and instance parameters
- Link Verilog-A ports with SPICE ports
- Define “key” parameters
  - type (nmos/pmos), w, l, ad, as, pd, ps, w, multiplicity…
- Define output variables
  - gm, gds, power…
- Define the ‘conditional nodes’ (collapsible nodes)

Verilog-A Compact Models in SMASH Auto-documentation

Automatically generate:

- Model documentation
- Dialog to customize operating point file
- Dialog to trace an internal parameter

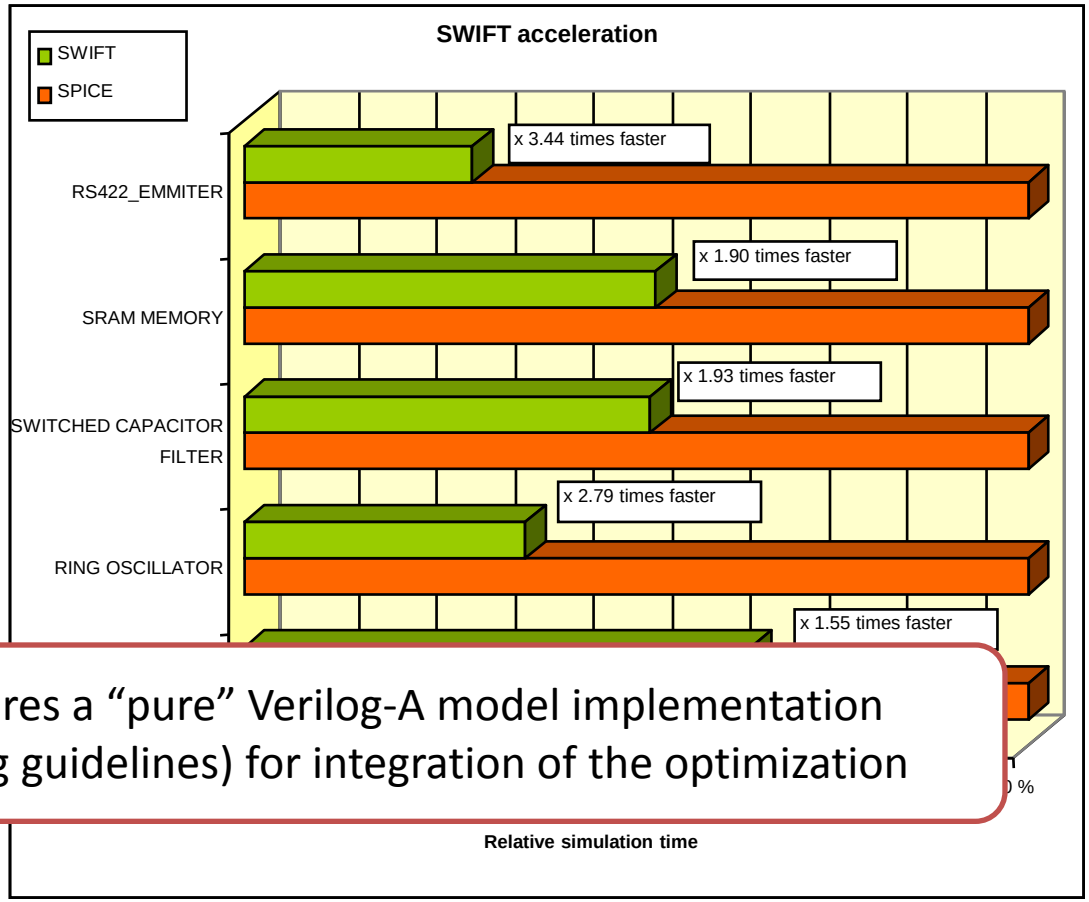


Requires (simple) modifications and additions to attribute_instance attributes (comments) in Verilog-A code

Verilog-A Compact Models in SMASH
SWIFT Linear Optimization

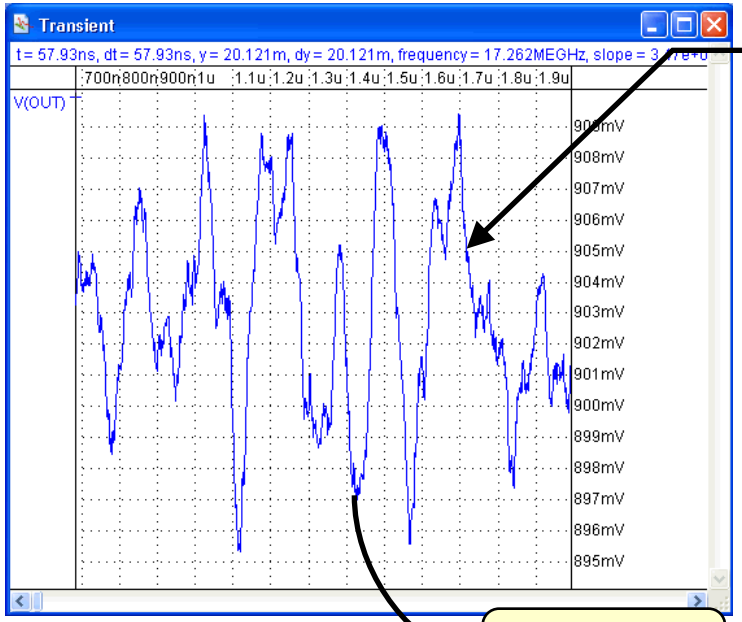
PATENTED

- Patented linearization algorithm to remove the discontinuities
- Acceleration 2 to 3 times for all transistor based circuits in transient analysis
- Reduced loss of accuracy (less than 0.1 %)
- No counter-indication concerning the applicable class of circuits
- Simplified setup with a single threshold value
- Continuous dynamic speed-accuracy trade-off

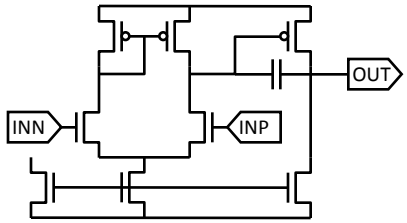


Requires a “pure” Verilog-A model implementation (coding guidelines) for integration of the optimization

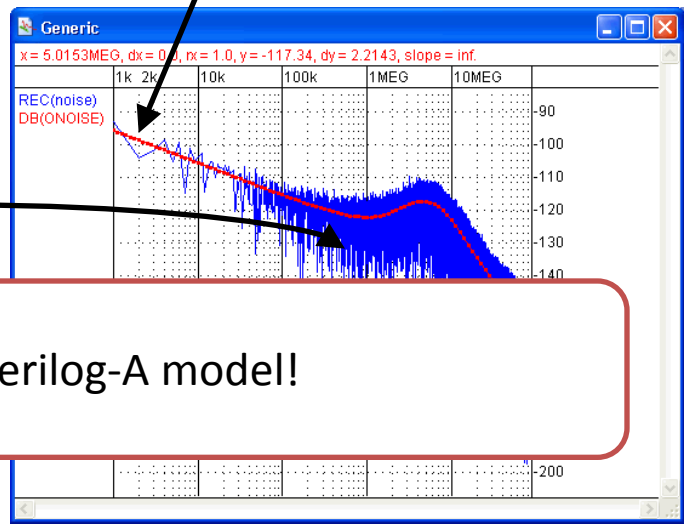
Verilog-A Compact Models in SMASH Transient Noise



Transient noise simulation



Noise simulation (AC)



FFT

SMASH can simulate in small signal as in transient any noise of form:

$$\sum_n f_n(time) * g_n(freq$$

Where:

f_n is a function of time (g

g_n is a function of frequency

Valid for any Verilog-A model!

Need for A standard subset of Verilog-A

- What is missing for direct use of Verilog-A Compact Models to be as efficient as SPICE for the final user?
  - Definition of a Verilog-A subset for the Compact Modeling
    - do not use state variables, $simprobe…
  - Differentiate model parameters and instance parameters
    - paramsets are a better solution only for Verilog-A netlists
  - Normalize the definition of 'conditional nodes' (collapsible nodes)
  - Be able to link Verilog-A ports with SPICE ports
  - Be able to define "key" parameters
    - type (nmos/pmos), w, l, ad, as, pd, ps, w, multiplicity…
  - Be able to define "key" variables
    - gm, gds, power…

➡ Target speed and memory occupation similar to SPICE

Conclusion

- Adoption of Verilog-A for Compact Modeling presents numerous benefits for all actors
  - Modelers, Integrator, Foundries, Final Users
- Adoption of Verilog-A models in place of SPICE models by the final users requires
  - Like for synthesis, definition and standardization of a subset of Verilog-A and of coding guidelines for Compact Modeling
  - Improvement of existing Verilog-A compilers to catch up with SPICE simulators
  - Extraction strategy for model parameter sets in Verilog-A?

Thanks!