Design-Adaptive Device Modeling in Model Compiler for Efficient and Accurate Circuit Simulation

Bo Wan (Cadence)

Emrah Acar, Sani Nassif (IBM Research - Austin)

C. -J. Richard Shi (Univ. of Washington)

Outline

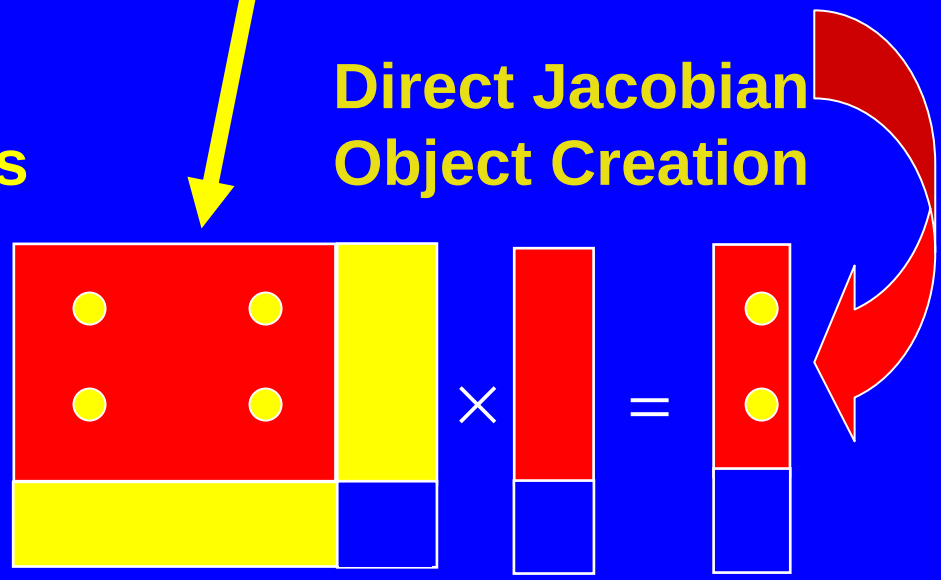
- Background
- Motivation
- Design-adaptive Device Modeling
- Experimental Results
- Conclusions

Manually Coded SPICE Built-in Models

$$\ln\left(\frac{1+e^{(B-C+n_1\cdot V)/Vt}}{1+e^{(B-C-n_1\cdot V)/Vt}}\right)\cdot\left(\frac{\pi}{2}+\arctan\left(\frac{C-n_1\cdot V}{D}\right)\right)$$

- 10-30K lines C codes
- Model & Instance Parameters
- Manually Matrix setup & loading
  - Compute matrix stamps
  - Compute derivatives
  - Implement multiple, self-consistent entry pointes for analysis types
- Boundary / Limiting
- Error prone, debug cost

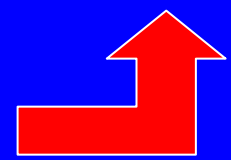
Direct Jacobian Object Creation



Mosfet Model Implementation in Spice3f5

High Cost!

Number	Level 1	Level 3	BSIM3	BSIM4	BSIMSOI
If statements	501	609	822	1495	1757
Parameters in codes	32	38	399	610	615
Intermediate Variables	56	56	430	811	972
Total codelines	7,673	8,634	12,348	20,113	20,664



So much work & still NOT stable!

BSIMSOIv3.2 Bug Fixes Report (2/24/2004)

	Bug description	Involved c-file	Bug reporter
1	GDoverlapCap/GSoverlapCap is not cancelled out	B3soild.c b3soiacld.c	M. Ahmed (Mentor Graphics)
2	Vgd/vgs replaced by vgmd/vgms	B3soild.c	(Renesas)
3	Here->B3SOIqinv for CapMod=2	B3soild.c	S. Wu, P. Yao (Cadence)
4	Calculation of gcegb is removed	B3soild.c	H. Wan (UC Berkeley)
5	Gcegmb term is removed in ceqqb	B3soild.c	B. Ritchie(Cadence)
6	Vgme replaced by vgmb in ceqqd	B3soild.c	B. Ritchie(Cadence)
7	Gcegmb term is added in ceqqe	B3soild.c	H. Wan (UC Berkeley)
8	Vgme replaced by vgmb in ceqqgmid	B3soild.c	H. Wan (UC Berkeley)
9	Rbogy clamping value changed	B3soild.c	B. Ritchie (Cadence)
10	pParam->B3SOIacde wrong position	B3soitemp.c	B. Ritchie (Cadence)
11	dAbult_dVg is added when Abulk<0.01	B3soild.c	C. Bittner, J. Watts (IBM)

(*(here->B3SOIDPbPtr)-=(-gddpb-Gmbs+gcdgb+gcddb +gcdeb+gcdsb)+gcdgmb+gIdtotb);

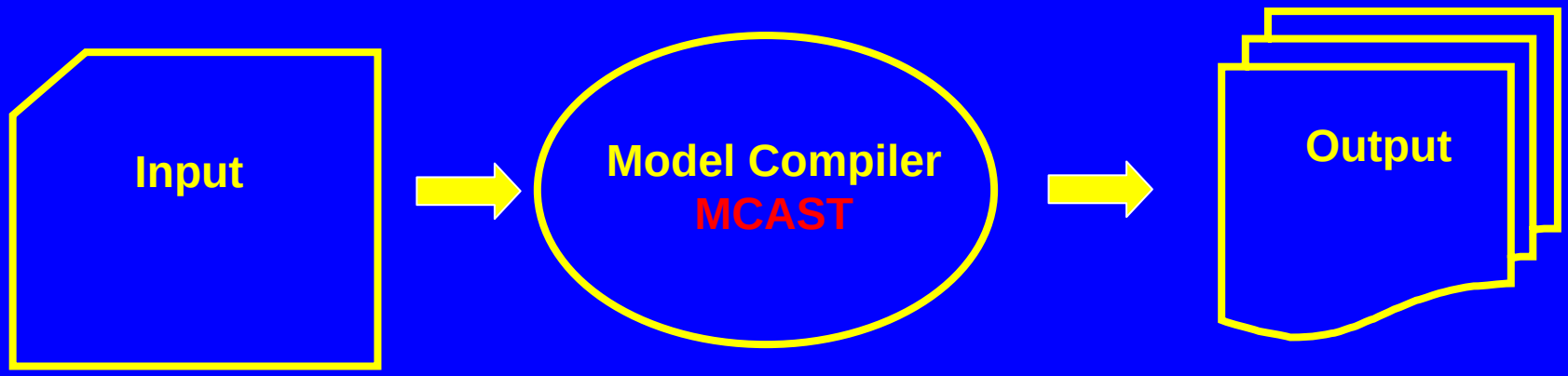
No Implementation bugs using model compiler!

Manual Implementation Issues

- Model implementation is more difficult than model creation. (time consuming, error-prone)
- Maintenance is a nightmare (“I hate this”)
- A lot of new models created, few implemented
- Modification of model is difficult
- Requires independent implementations for different simulator vendor
- Same model, different results in different simulators. Which one is trustworthy? No industrial standard!

Model Compiler in a Nutshell

A Model Compiler is a design automation tool that supports fast prototyping of device models.



1. Input: compact device models described in high level behavioral language VHDL-AMS/Verilog-AMS
2. Output: device code in C/C++ that can be directly compiled in target circuit simulators.

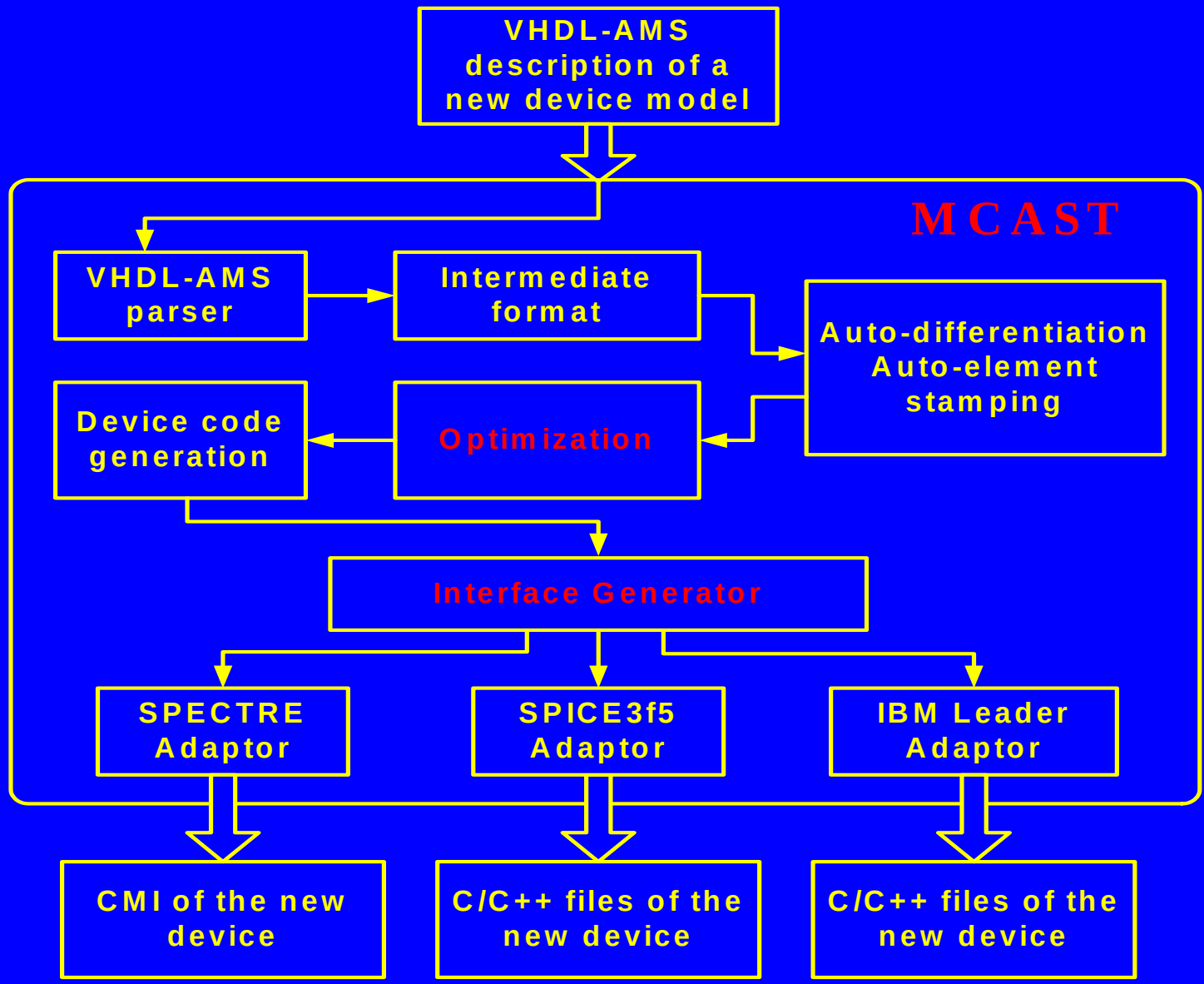
Model Compiler Solution

- Use VHDL-AMS/Verilog-A for device model definition.
- Use device model compiler and support tools to create automatic device models that can be used in a wide range of simulation platforms.
- Use automation techniques to provide robustness better than manually coded built-in models.
- Use optimization techniques in model compiler to generate efficient models for circuit simulation.

Related Work

- S. Liu, K. C. Hsu, P. Subramaniam, “ADMIT-ADVICE Modeling Interface Tool”, 1988
- A. T. Yang, and S. M. Kang, “iSMILE: A Novel Circuit Simulation Program with Emphasis on New Device Model Development”, 1989
- R. V. H. Booth, “An Extensible Compact Model Description Language and Compiler”, 2001
- L. Lemaitre, C. McAndrew, and S. Hamm, “ADMS-Automatic Device Model Synthesizer”, 2002
- Model compilers already available: ADS (Tiburon Design Automation), ADMS(Motorola), …

MCAST Architecture

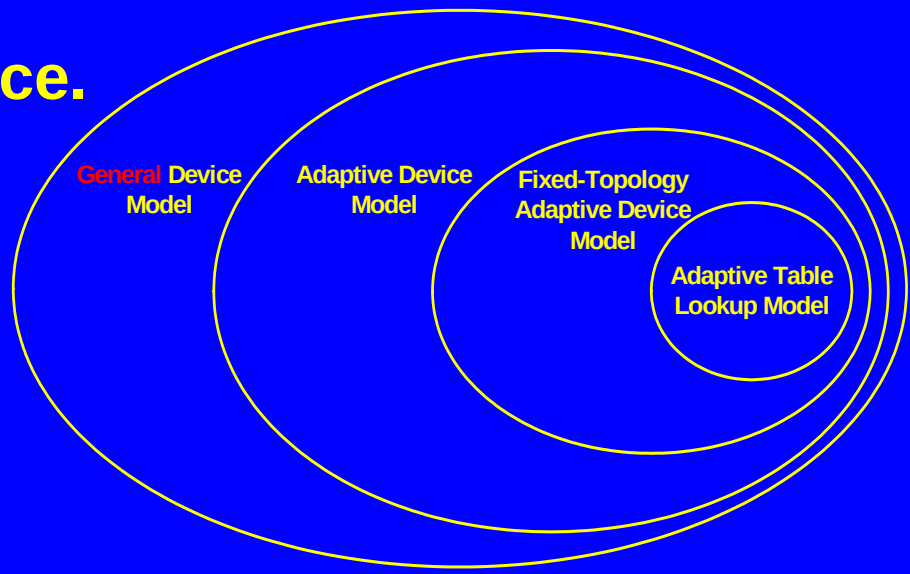


Motivation of Design-Adaptive Modeling

- Automatically generated model can be slower than built-in model. Memory usage is larger (~5x) than built-in model.
- No previous work has taken user design information into account when generating device models.
- User design information (model/instance parameters, topology, etc) can be used to simplify the general device model and generate very efficient device models for fast circuit simulation.
- It is only feasible through a design automation tool such as model compiler.

Design-adaptive Device Model

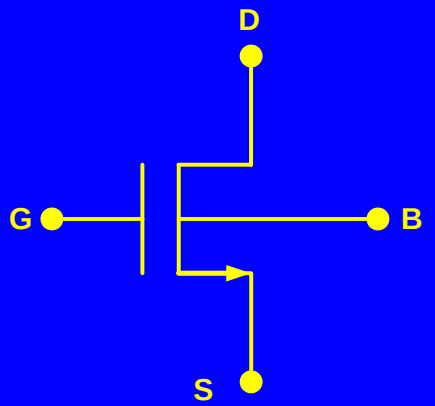
Design-adaptive device model is a more *specific* model of a device.



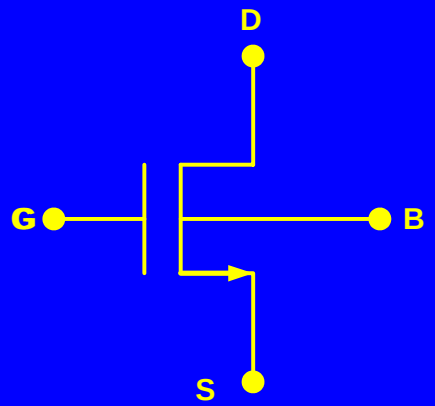
Characteristics:

- **Fixed model parameters**
- **Fixed some or all instance parameters**
- **Specific topology**

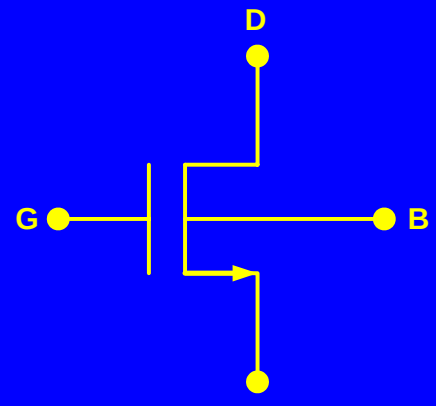
MOSFET Design-adaptive Device Models



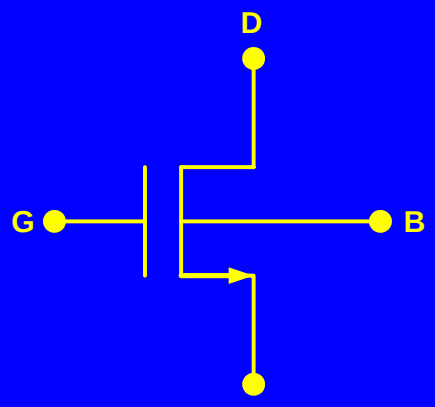
All model and instance parameters NOT fixed
General model
(a)



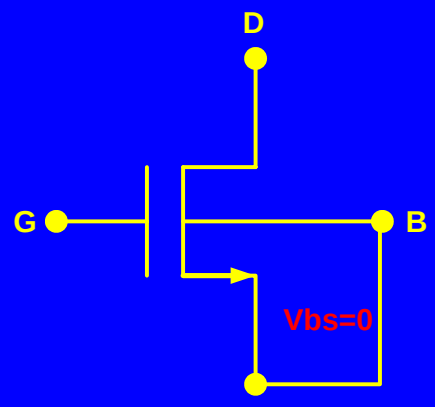
Model parameters fixed
Instance parameters NOT fixed
Dynamic model
(b)



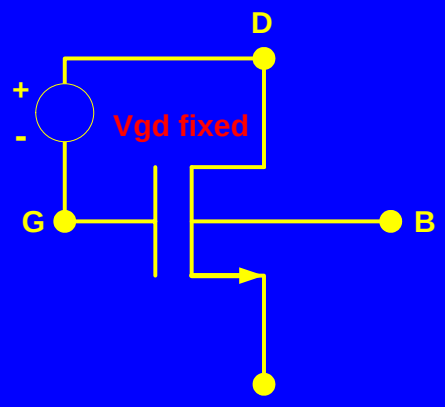
W NOT fixed, L fixed
Dynamic model
(c)



W & L fixed
Dynamic model
(d)

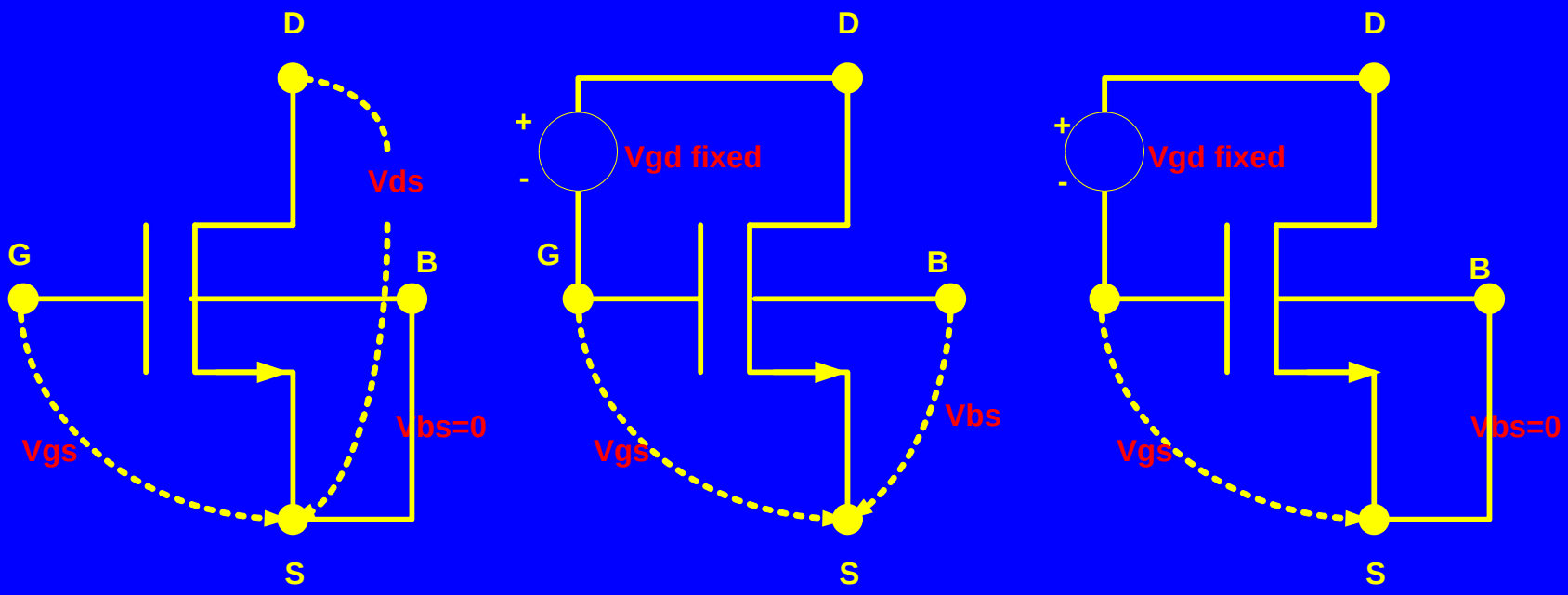


Vbs=0
Dynamic model
(e)



Vgd fixed
Dynamic model
(f)

Adaptive Low-dim Table Lookup Models



Vbs=0 Adaptive model

(d+e)

2-D table lookups: IDS=IDS(Vgs,Vds) etc.

Vgd fixed Adaptive model

(d+f)

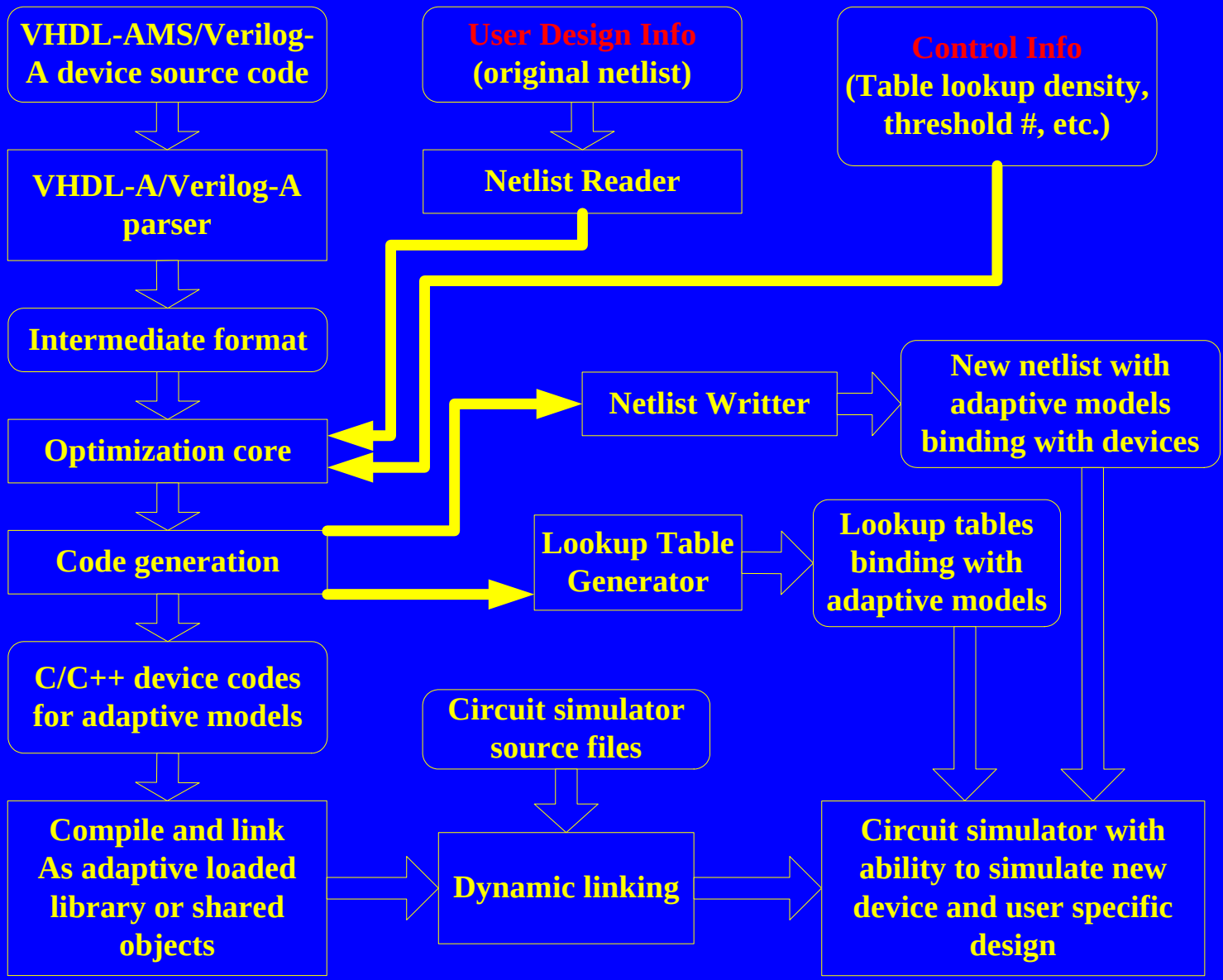
2-D table lookups: IDS=IDS(Vgs,Vbs) etc.

Vgd fixed Adaptive model

(d+e+f)

1-D table lookups: IDS=IDS(Vgs) etc.

New Architecture for Design-adaptive



New Netlist

Old netlist read in by MCAST

```
.MODEL mypmos PMOS (LEVEL=8 TOX=4.2E-9 …)
.MODEL mynmos NMOS (LEVEL=8 TOX=4.2E-9 …)

M1 out in vdd vdd mypmos w=30u l=6u
M2 out in gnd gnd mynmos w=10u l=6u
```

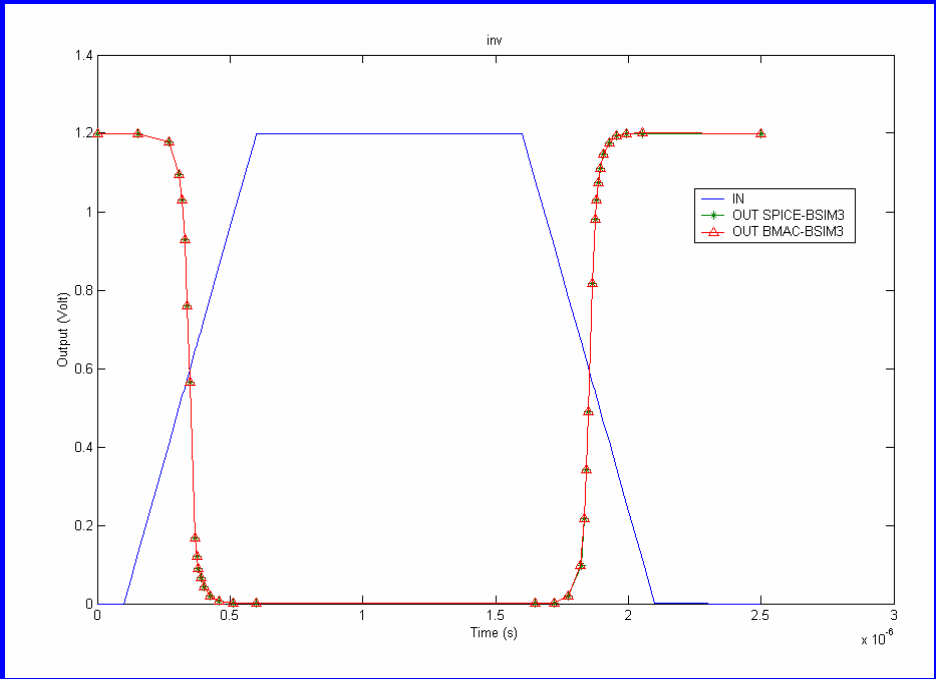
New netlist generated by MCAST

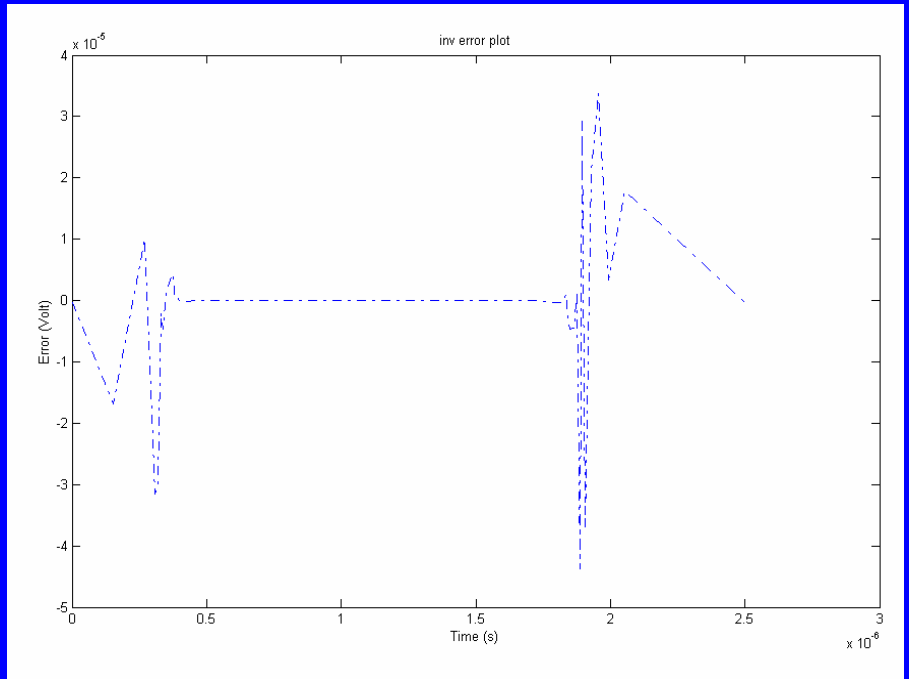
```
* GENERAL MODEL GPMOS GENERATED BY MCAST
.MODEL myp GPMOS TNOM=22 TOX=4.2E-9 ...
* GENERAL MODEL GNMOS GENERATED BY MCAST
.MODEL myn GNMOS TNOM=22 TOX=4.2E-9 ...

* DYNAMIC MODEL DM1: type (d+e), PMOS
* model parameters: TNOM=22 TOX=4.2E-9 ...
* instance parameters: w=30u l=6u
* topology: Vbs=0 (e)
.MODEL mypmos DM1

* DYNAMIC MODEL DM2: type (d+e), NMOS
* model parameters: TNOM=22 TOX=4.2E-9 ...
* instance parameters: w=10u l=6u
* topology: Vbs=0 (e)
.MODEL mynmos DM2

* USE DYNAMIC MODEL
* NO INSTANCE PARAMETER NECESSARY
N1 out in vdd vdd mypmos
N2 out in gnd gnd mynmos

* STILL USE GENERAL MODEL
N3 out in vdd vdd myp w=25u l=6u
N4 out in gnd gnd myn w=8u l=6u
```

Experimental Results: Benchmark Circuits

Index	Circuit	# MOS	# ADM	# ADTLM
1	One-Shot	22	4	2
2	VCO	10	2	2
3	Power AMP	4	0	2
4	Ring Oscillator	12	0	2
5	Boeing Comparator	38	2	2
6	Complex Cell	30	1	4
7	INV	2	0	2
8	INV Chain	8	0	2
9	NAND2	4	0	2
10	NOR2	4	0	2
11	AOI22	8	0	2
12	OAI22	16	0	2
13	ACCAS	1038	0	2
14	DFF	24	0	2
15	SRAM	6910	9	4
16	26-bit Adder	4274	18	6
17	13-bit Multiplier	9545	24	16
18	12-bit Divider	3081	22	8

Experimental Results: Accuracy

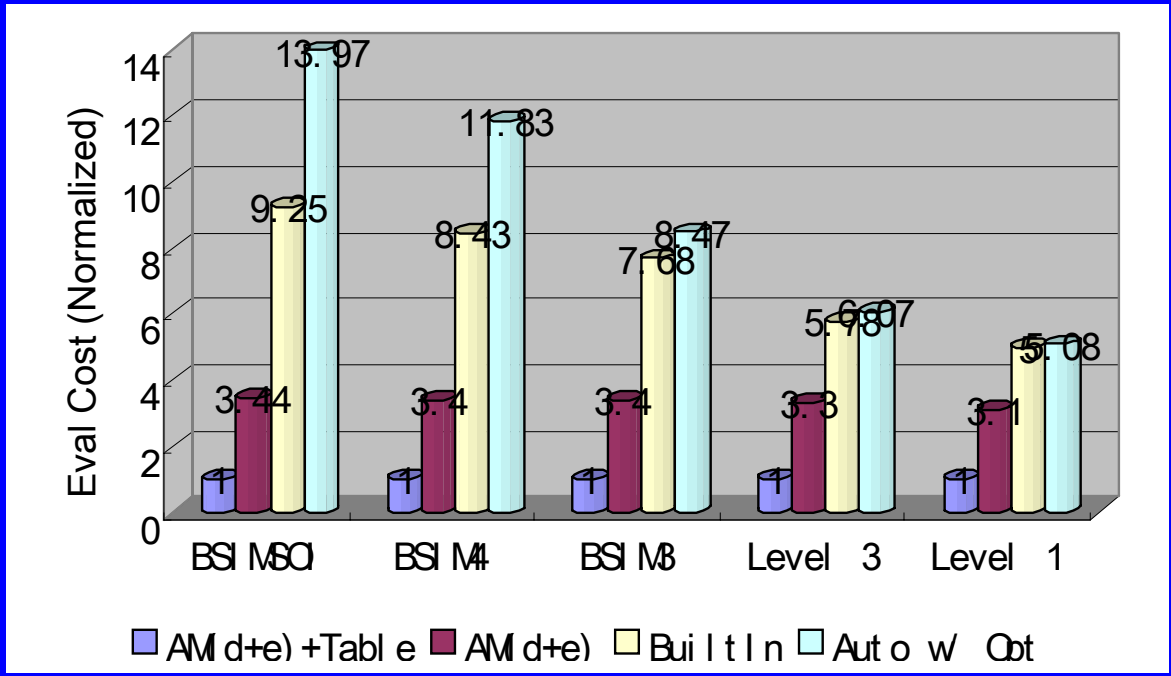




Inherently Very Accurate:

No loss of Accuracy

Performance

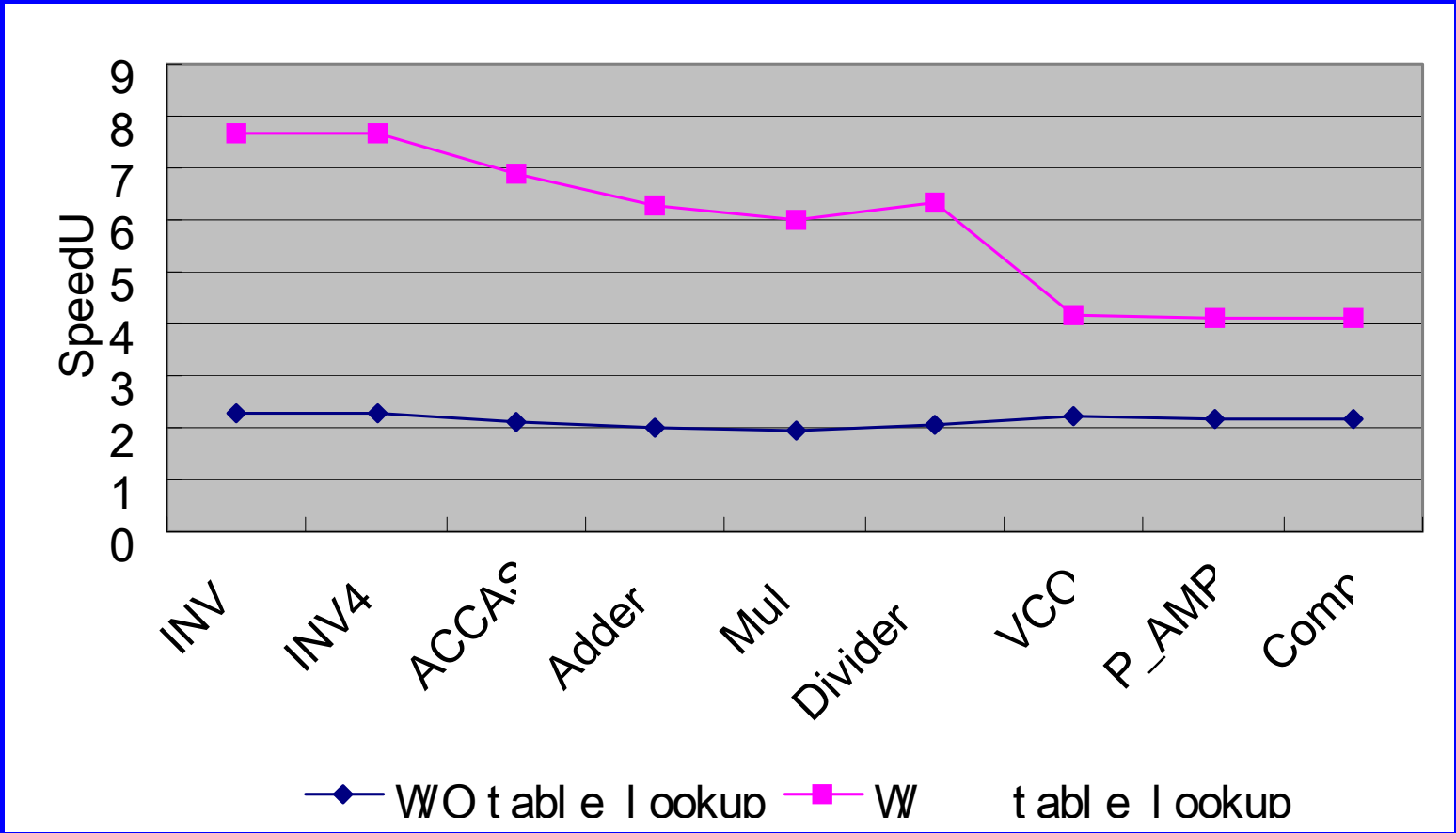


Normalized Model Evaluation Cost Comparison to Manually Coded Built-in Model:

Design-adaptive models are > 3x faster!

Adaptive table lookup models up to 9.25x faster!

Transient Comparison



Speedup: 4x-8x

Conclusions

- **High-level behavioral languages (VHDL-AMS/Verilog-AMS): good for compact device modeling. They are becoming industry standard.**

- **Model compiler has realized the concept: “Device modeling in a day”**

- **Design-adaptive generated models are 1) Robust, 2) Efficient and 3) Accurate.**

- **Model compiler engine future applications:**
  1. **hierarchical behavioral design**
  2. **circuit optimization**
  3. **model debugging**
  4. **statistical analysis**