High-Performance Model Compilation for Complex Behavioral Models

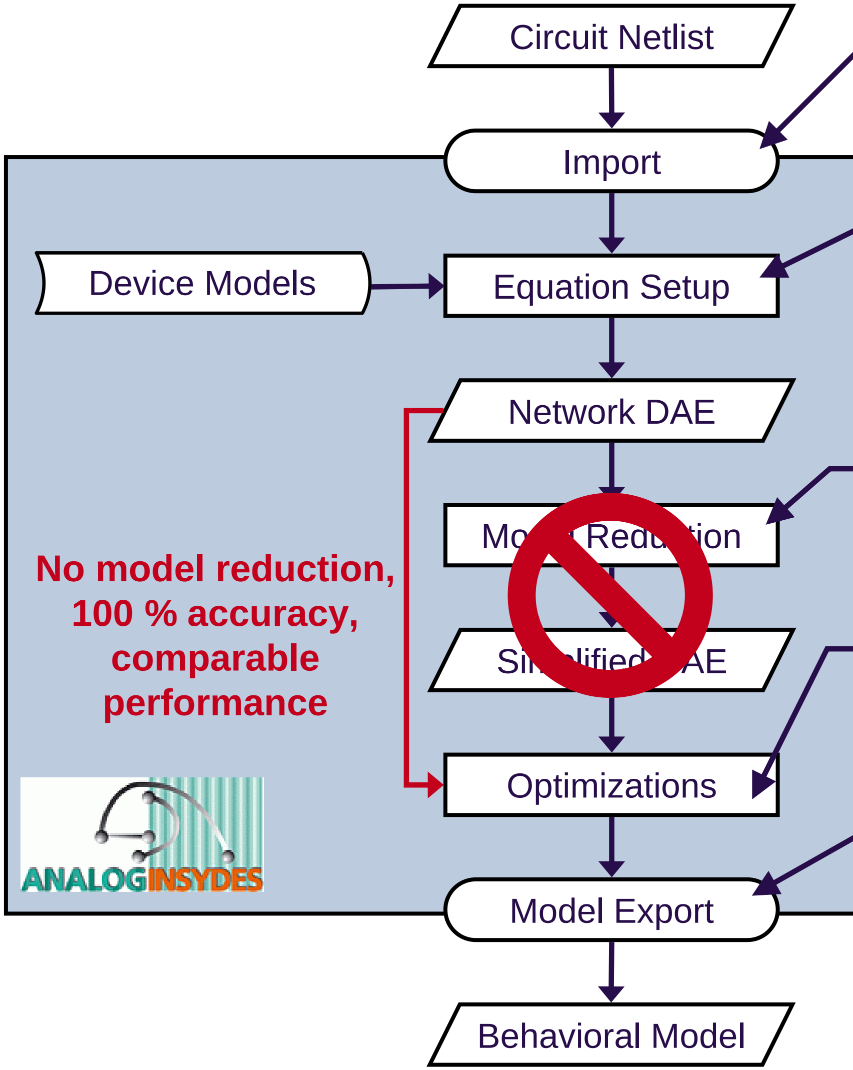
September 20th, 2007

Daniel Platte

Outline

- **Bottom-Up Modeling Through Symbolic Analysis**
- Model Compilation
- Simulation Results
- Summary

Bottom-Up Modeling Flow



Import of several netlist formats (Spice, Spectre, etc), Cadence Design Framework integration

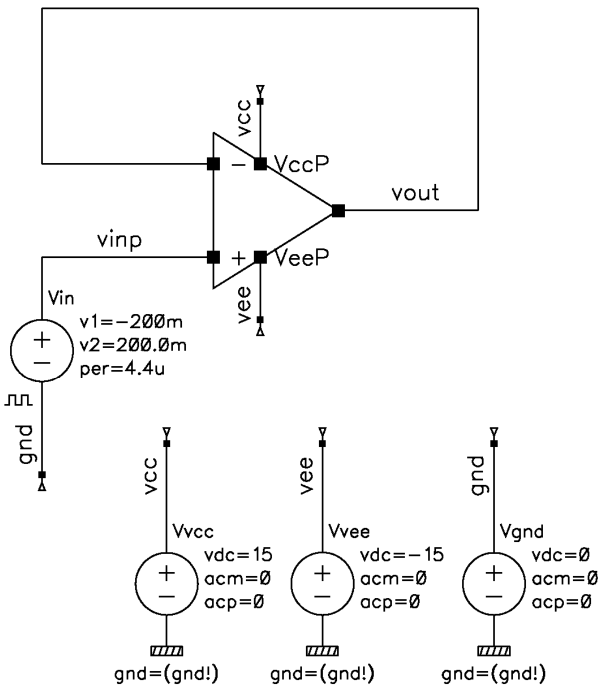
Automated setup of symbolic network equations through an extended MNA and symbolic device models

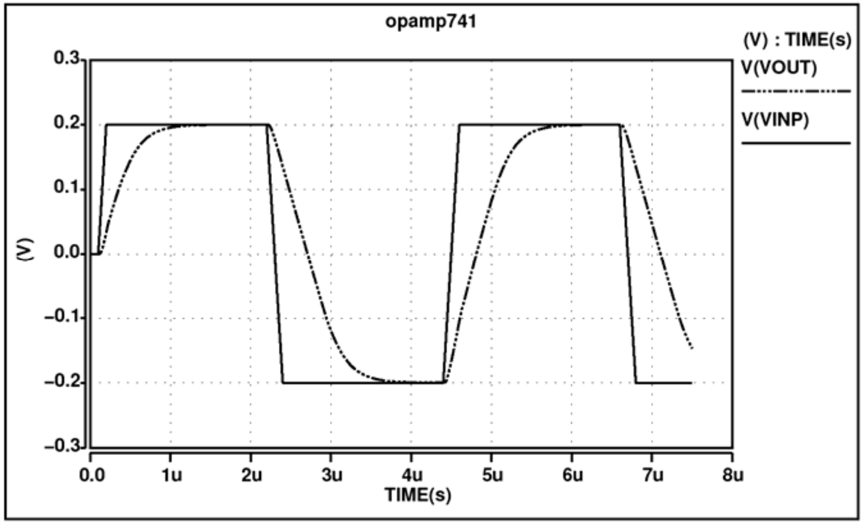
Efficient symbolic model reduction techniques for nonlinear DAEs with continuous numeric error control

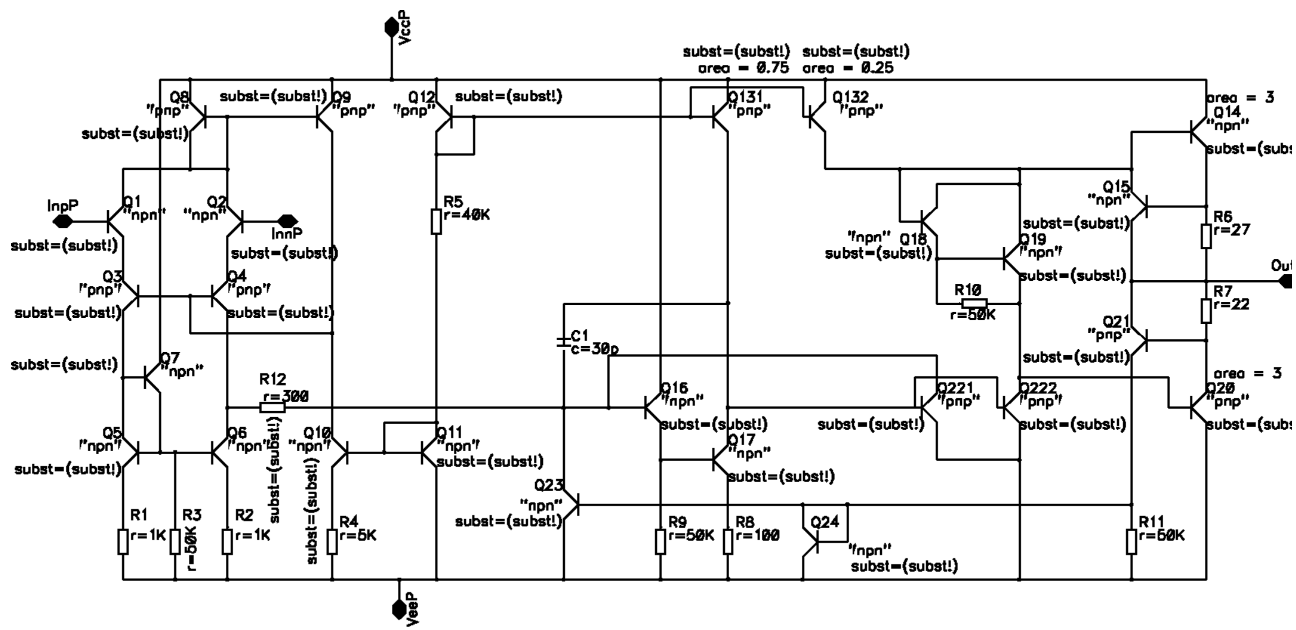
Algebraic reformulation and restructuring of the DAEs to optimize simulation performance (no impact on the accuracy)

Export the DAEs to several common AHDLs incl. VHDL-AMS, Verilog-A, MAST, (Modelica)

Example µA741





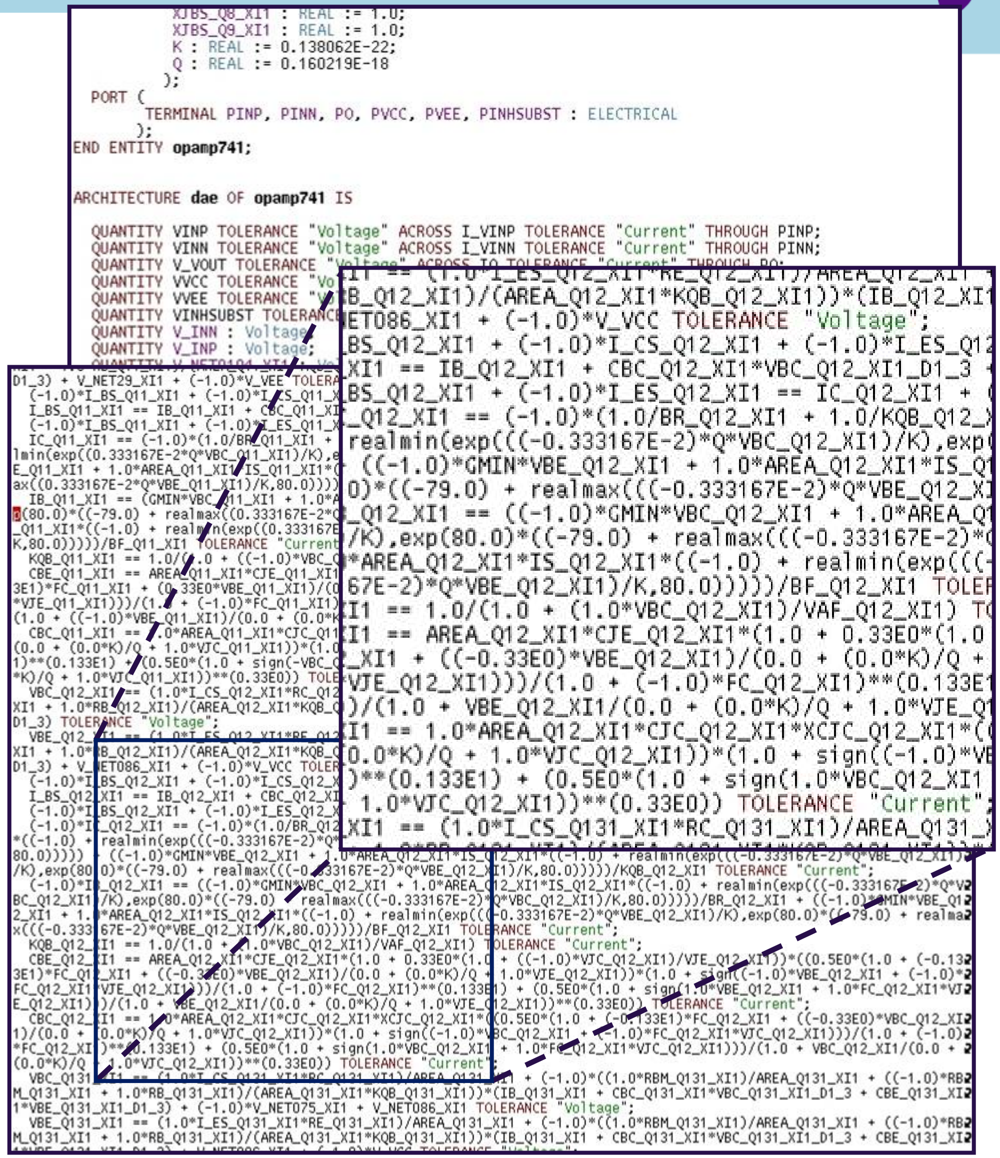


Analytic Model of µA741

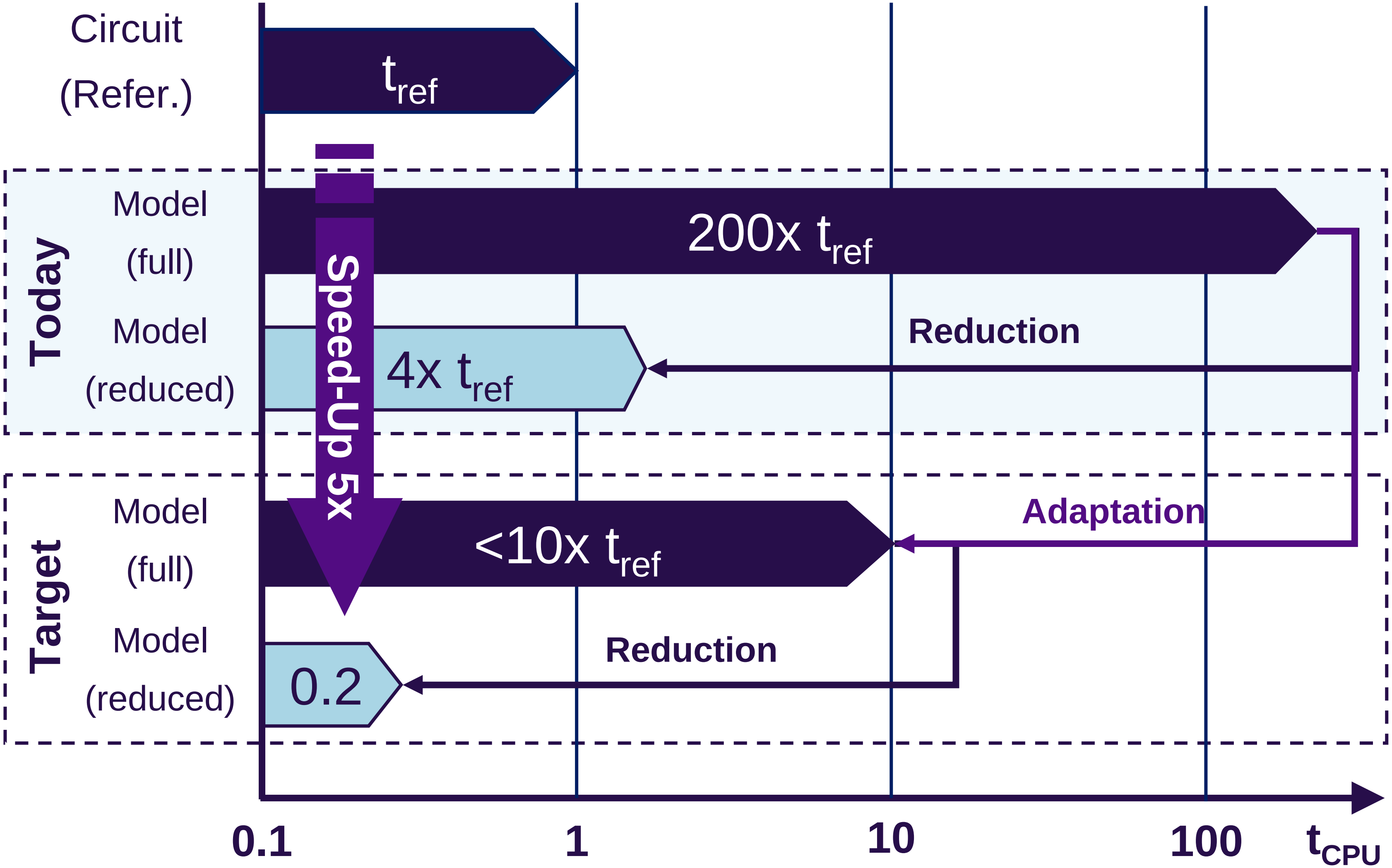
µA741	Dimension	Sparsity
Circuit	62	90.3%
Model (MNA)	368	98.8%

26 Transistors
(Gummel-Poon)

No model reduction applied
564 Parameters
115 kB Code Size
1300 Lines of Code VHDL
10 additional internal variables per BJT



Motivation



Circuit
(Refer.)
t_{ref}
Today
Model
(full)
$200x\ t_{ref}$
Model
(reduced)
$4x\ t_{ref}$
Reduction
Speed-Up 5x
Target
Model
(full)
$<10x\ t_{ref}$
Adaptation
Model
(reduced)
0.2
Reduction
0.1
1
10
100
t_{CPU}

Reasons for Performance Problems

- „Intelligence“ of built-in device models missing
- Equation formulation not optimized for numerical methods
- Model compilers do not efficiently handle such complex behavioral models
- Restrictions by using common AHDLs
  - procedural statements in VHDL-AMS
  - simultaneous equation sets in Verilog-A
  - no optimization of Jacobian matrix possible

Outline

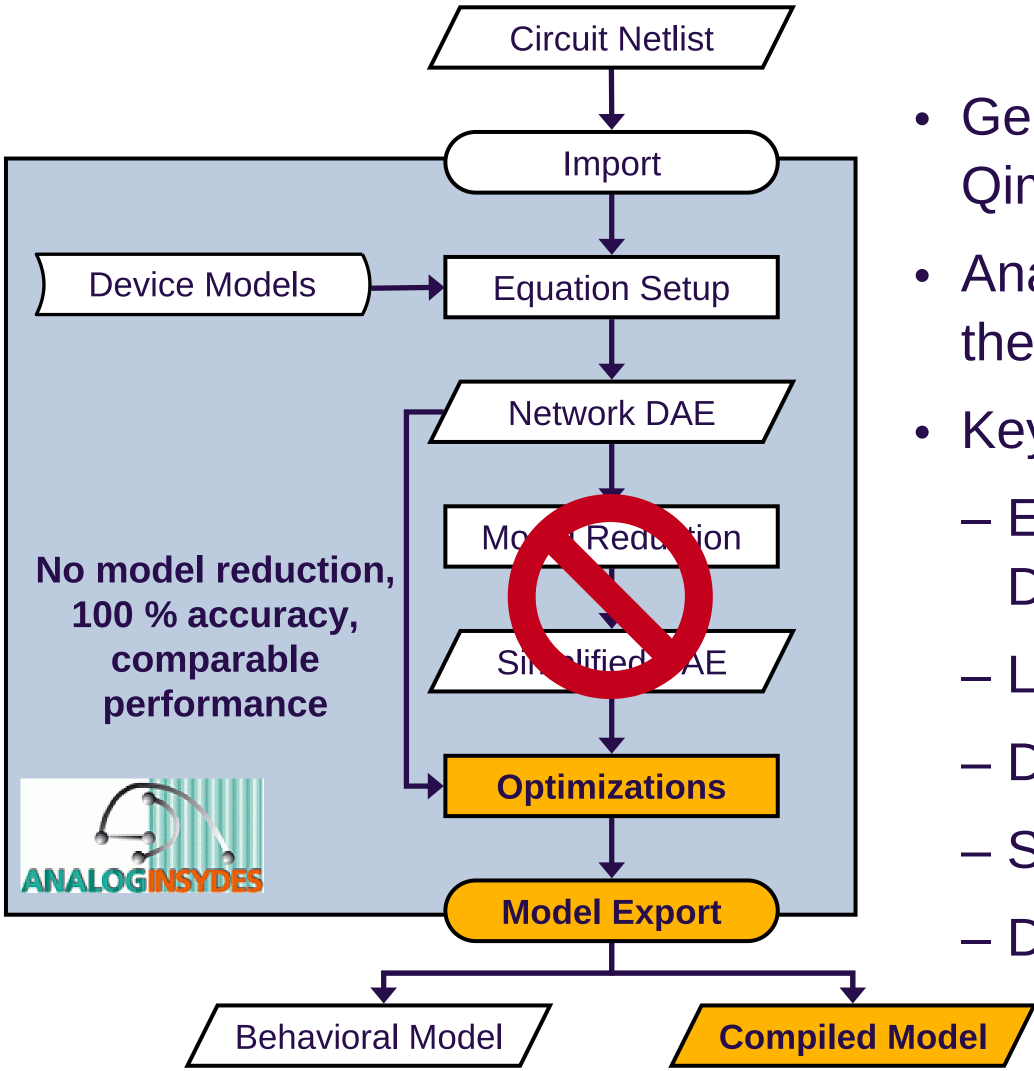
Bottom-Up Modeling Through Symbolic Analysis

Model Compilation

Simulation Results

Summary

Model Compilation with Analog Insydes



- Generation of compiled models for Qimonda‘s inhouse-simulator Titan
- Analog Insydes is used as platform for the model compilation
- Key features:
  - Efficient processing of simultaneous DAE
  - Local solving of sequential equations
  - Data locality (cache)
  - Sparse handling and data structures
  - Direct interface to the simulator kernel

Example – Sequential Network Equations

GND
A
K
B
VIN
VOUT
VGND

Sequential

$$\mathrm{vd}[t] == -\frac{\mathrm{RS}\ \mathrm{I\$AC}[t]}{\mathrm{AREA}} + \mathrm{V\$IN}[t] - \mathrm{V\$OUT}[t]$$

$$\mathrm{id}[t] == \mathrm{AREA}\left(-e^{\frac{0.00333167\, q\, (\mathrm{BV}+\mathrm{vd}[t])}{k}}\ \mathrm{IBV} + \left(-1 + e^{\frac{0.00181069\, q\, \mathrm{vd}[t]}{k}}\right)\mathrm{IS}\right) + \mathrm{GMIN}\ \mathrm{vd}[t]$$

$$\mathrm{cj}[t] == \mathrm{AREA}\ \mathrm{CJO}\ \mathrm{If}\left[\mathrm{vd}[t] < 0.25,\ \frac{1}{(1.-2.\,\mathrm{vd}[t])^{0.333}},\ 2.51926\,(0.3335 + 0.666\,\mathrm{vd}[t])\right]$$

Simultaneous

$$\mathrm{I\$VGND}[t] + \frac{\mathrm{V\$GND}[t]-\mathrm{V\$OUT}[t]}{\mathrm{R0}} + \mathrm{C0}\,(\mathrm{V\$GND}'[t] - \mathrm{V\$OUT}'[t]) == 0$$

$$\mathrm{I\$AC}[t] + \mathrm{I\$VIN}[t] == 0$$

$$-\mathrm{I\$AC}[t] + \frac{-\mathrm{V\$GND}[t]+\mathrm{V\$OUT}[t]}{\mathrm{R0}} + \frac{\mathrm{V\$OUT}[t]-\mathrm{V\$OUT2}[t]}{\mathrm{RL}} + \mathrm{C0}\,(-\mathrm{V\$GND}'[t] + \mathrm{V\$OUT}'[t]) == 0$$

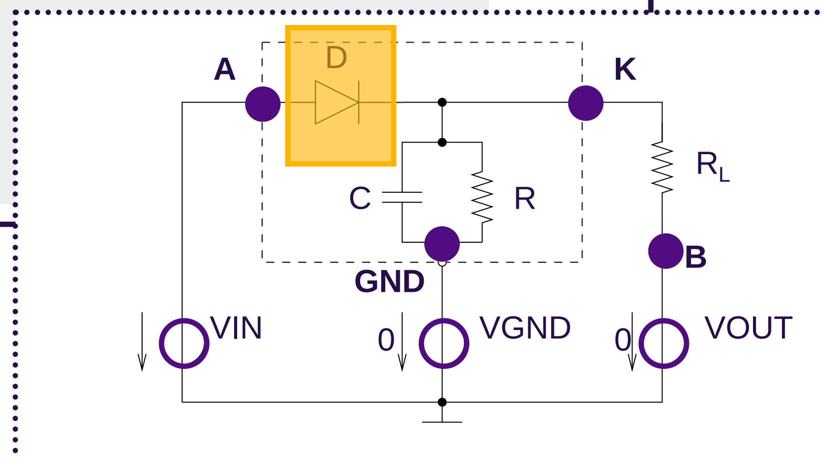
$$\mathrm{I\$VOUT}[t] + \frac{-\mathrm{V\$OUT}[t]+\mathrm{V\$OUT2}[t]}{\mathrm{RL}} == 0$$

$$\mathrm{I\$AC}[t] == \mathrm{id}[t] + 1.15\times 10^{-8}\,\mathrm{id}'[t] + \mathrm{cj}[t]\,\mathrm{vd}'[t]$$

$$\mathrm{V\$IN}[t] == \mathrm{VIN}$$

$$\mathrm{V\$OUT2}[t] == 0$$

$$\mathrm{V\$GND}[t] == 0$$



Newton‘s Method for Sequential DAEs

$$\begin{bmatrix} \partial f_{seq} / \partial \mathbf{y} & \partial f_{seq} / \partial \mathbf{x} \\ \partial f_{sim} / \partial \mathbf{y} & \partial f_{sim} / \partial \mathbf{x} \end{bmatrix} \cdot \begin{bmatrix} \Delta \mathbf{y} \\ \Delta \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix} \cdot \begin{bmatrix} \Delta \mathbf{y} \\ \Delta \mathbf{x} \end{bmatrix} = - \begin{bmatrix} f_{seq}(\mathbf{y}, \mathbf{x}) \\ f_{sim}(\mathbf{y}, \mathbf{x}) \end{bmatrix}$$

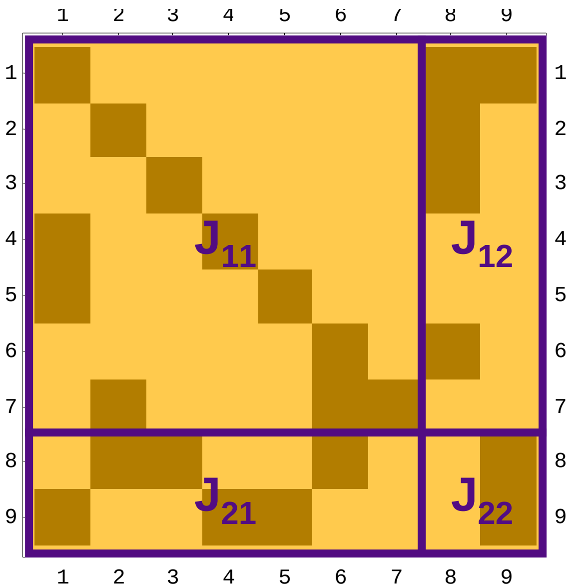
$$\Delta \mathbf{y} = \mathbf{J}_{11}^{-1} \cdot \left(f_{seq}(\mathbf{y}, \mathbf{x}) + \mathbf{J}_{12} \Delta \mathbf{x} \right)$$

$$\left(\mathbf{J}_{22} - \mathbf{J}_{21} \cdot \mathbf{J}_{11}^{-1} \cdot \mathbf{J}_{12} \right) \cdot \Delta \mathbf{x} = -f_{sim}(\mathbf{y}, \mathbf{x}) + \underbrace{\mathbf{J}_{21} \cdot \mathbf{J}_{11}^{-1} \cdot f_{seq}(\mathbf{y}, \mathbf{x})}_{0}$$

$$\mathbf{J}^{*}_{22} \cdot \Delta \mathbf{x} = -f_{sim}(\mathbf{y}, \mathbf{x})$$

1. Calculate y_{n+1} from x_n (sequential equations)
2. Update J with y_{n+1} and x_n
3. Calculate Schur complement $$\mathbf{J}^{*}_{22} = \mathbf{J}_{22} - \mathbf{J}_{21} \cdot \mathbf{J}_{11}^{-1} \cdot \mathbf{J}_{12}$$
4. Solve $$\mathbf{J}^{*}_{22} \cdot \Delta \mathbf{x} = -f_{sim}(\mathbf{y}, \mathbf{x})$$

Methods to Calculate Schur Complement



Schur Complement

$$\tilde{\mathbf{J}}_{22} = \mathbf{J}_{22} - \mathbf{J}_{21} \cdot \mathbf{J}_{11}^{-1} \cdot \mathbf{J}_{12}$$

1) **Symbolic Schur complement**
- extremely complex expressions, impossible

2) **Numerical calculation**
- matrix inversion and multiplications necessary

3) **Semi-symbolic Schur complement with roll-out**
- symbolically calculate the Schur complement with references to the evaluated Jacobian matrix
- generate an efficient transformation process by elimination of the submatrix J_{21}

Example Schur Complement

Structural Matrix

-J[2,1]

1	0	0	J[1,4]	J[1,5]
0	1	0	J[2,4]	J[2,5]
0	0	1	J[3,4]	J[3,5]
0	0	0	J[4,4]	J[4,5]
0	0	0	J[5,4]	J[5,5]

Transformation Process

J[2,4] = J[2,1]*J[1,4]
J[2,5] = J[2,1]*J[1,5]
J[2,1] = 0

J[3,4] = J[3,1]*J[1,4]
J[3,5] = J[3,5]-J[3,1]*J[1,5]
J[3,1] = 0

J[4,4] = J[4,4]-J[1,4]*J[4,1]
J[4,5] = J[4,5]-J[1,5]*J[4,1]

J[4,4] = J[4,4]-J[2,4]*J[4,2]
J[4,5] = J[4,5]-J[2,5]*J[4,2]
…

Optimization Strategies

- **Identification of Sequential Equations (BLT Transf.)**
  Automated recognition of sequential equations from general DAE systems to reduce the number of the simult. equations
- **Common Subexpression Elimination (CSE)**
  CSE on DAE level to reduce redundancy within the function evaluation and improve data locality
- **Loop Invariant Expressions / Preloading**
  Preevaluation of constant and parametric expressions within the initialization phase, recognition of parametric subexpressions
- **Copy / Constant Propagation**
  Reduce memory accesses by propagation of copied or constant values into subsequent expressions

Optimized DAE System

Example - Optimized DAE System

Sequential

$$\mathrm{vd}[t] == \mathrm{VIN} - \frac{\mathrm{RS}\,\mathrm{I\$AC}[t]}{\mathrm{AREA}} - \mathrm{V\$OUT}[t]$$

$$\mathrm{SeqExpr2}[t] == -\frac{\mathrm{V\$OUT}[t]}{\mathrm{R0}}$$

$$\mathrm{SeqExpr3}[t] == \frac{\mathrm{V\$OUT}[t]}{\mathrm{RL}}$$

$$\mathrm{id}[t] == -\mathrm{AREA}\left(e^{-\frac{0.00333167\,q\,(\mathrm{BV}+\mathrm{vd}[t])}{k}}\,\mathrm{IBV} - \left(-1 + e^{\frac{0.00181069\,q\,\mathrm{vd}[t]}{k}}\right)\mathrm{IS}\right) + \mathrm{GMIN}\,\mathrm{vd}[t]$$

$$\mathrm{cj}[t] == \mathrm{AREA}\,\mathrm{CJ0}\,\mathrm{If}\left[\mathrm{vd}[t] < 0.25,\ \frac{1}{(1.-2.\,\mathrm{vd}[t])^{0.333}},\ 2.51926\,(0.3335 + 0.666\,\mathrm{vd}[t])\right]$$

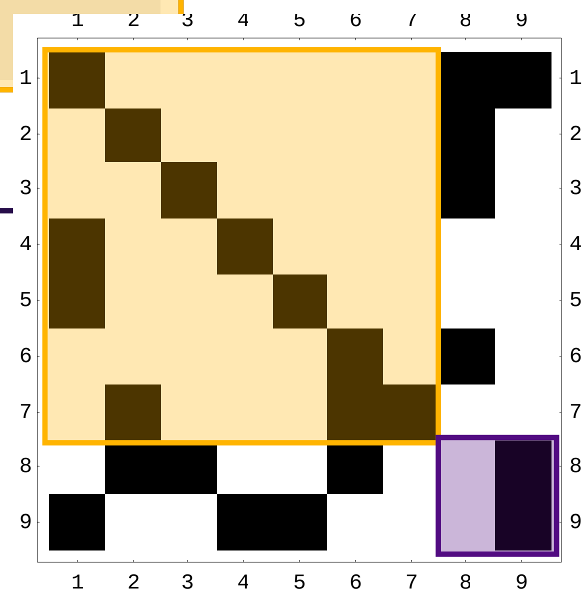
$$\mathrm{SeqExpr1}[t] == -\mathrm{C0}\,\mathrm{V\$OUT}'[t]$$

$$\mathrm{I\$VGND}[t] == -\mathrm{SeqExpr1}[t] - \mathrm{SeqExpr2}[t]$$

$$-\mathrm{I\$AC}[t] - \mathrm{SeqExpr1}[t] - \mathrm{SeqExpr2}[t] + \mathrm{SeqExpr3}[t] == 0$$

$$\mathrm{I\$AC}[t] == \mathrm{id}[t] + 1.15 \times 10^{-8}\,\mathrm{id}'[t] + \mathrm{cj}[t]\,\mathrm{vd}'[t]$$

Simultaneous

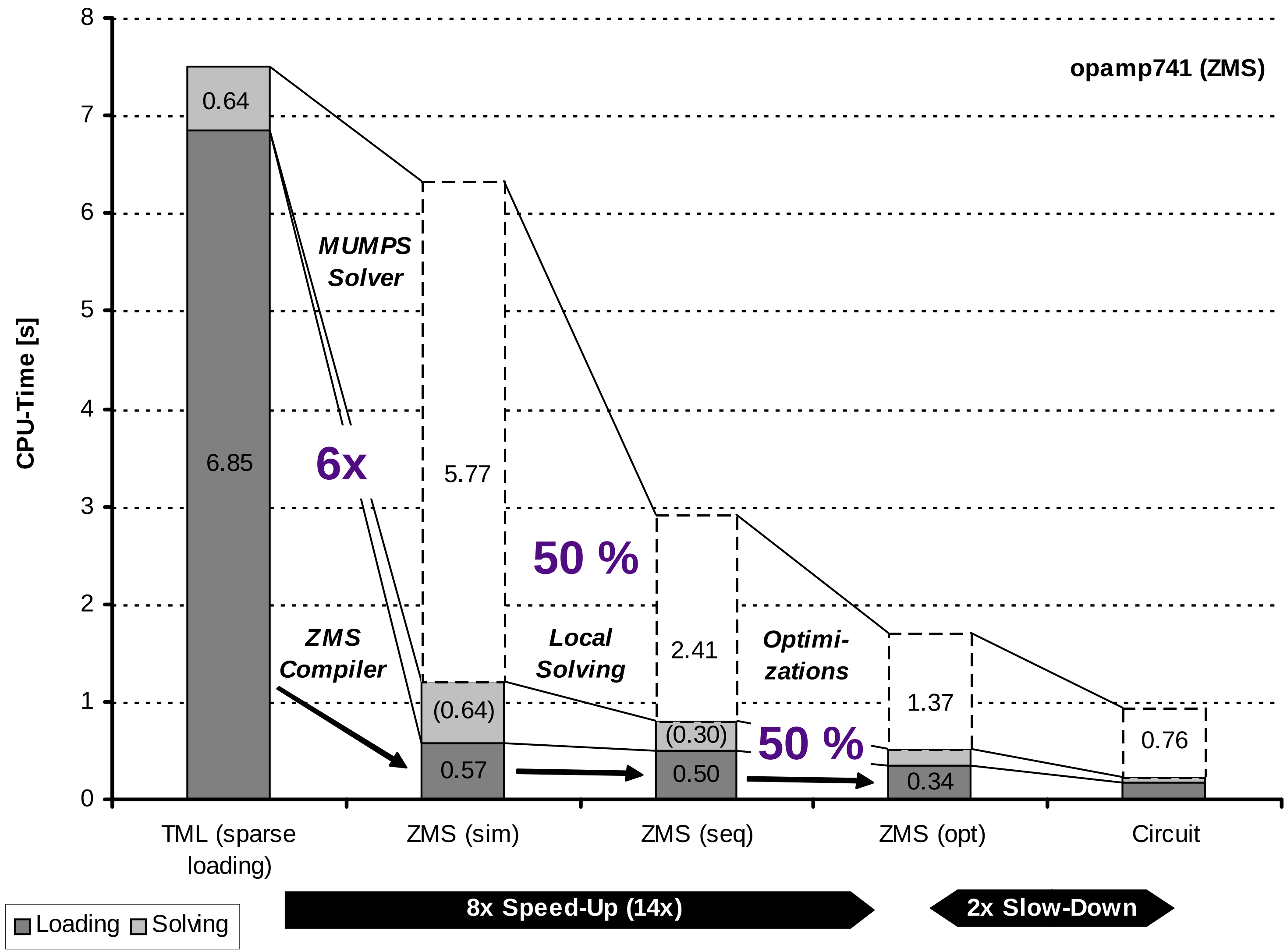


- Dimension reduced from 8x8 to 2x2
- No redundancy within DAEs (3 common subexpressions eliminated)

Outline

- Bottom-Up Modeling Through Symbolic Analysis
- Model Compilation
- **Simulation Results**
- Summary

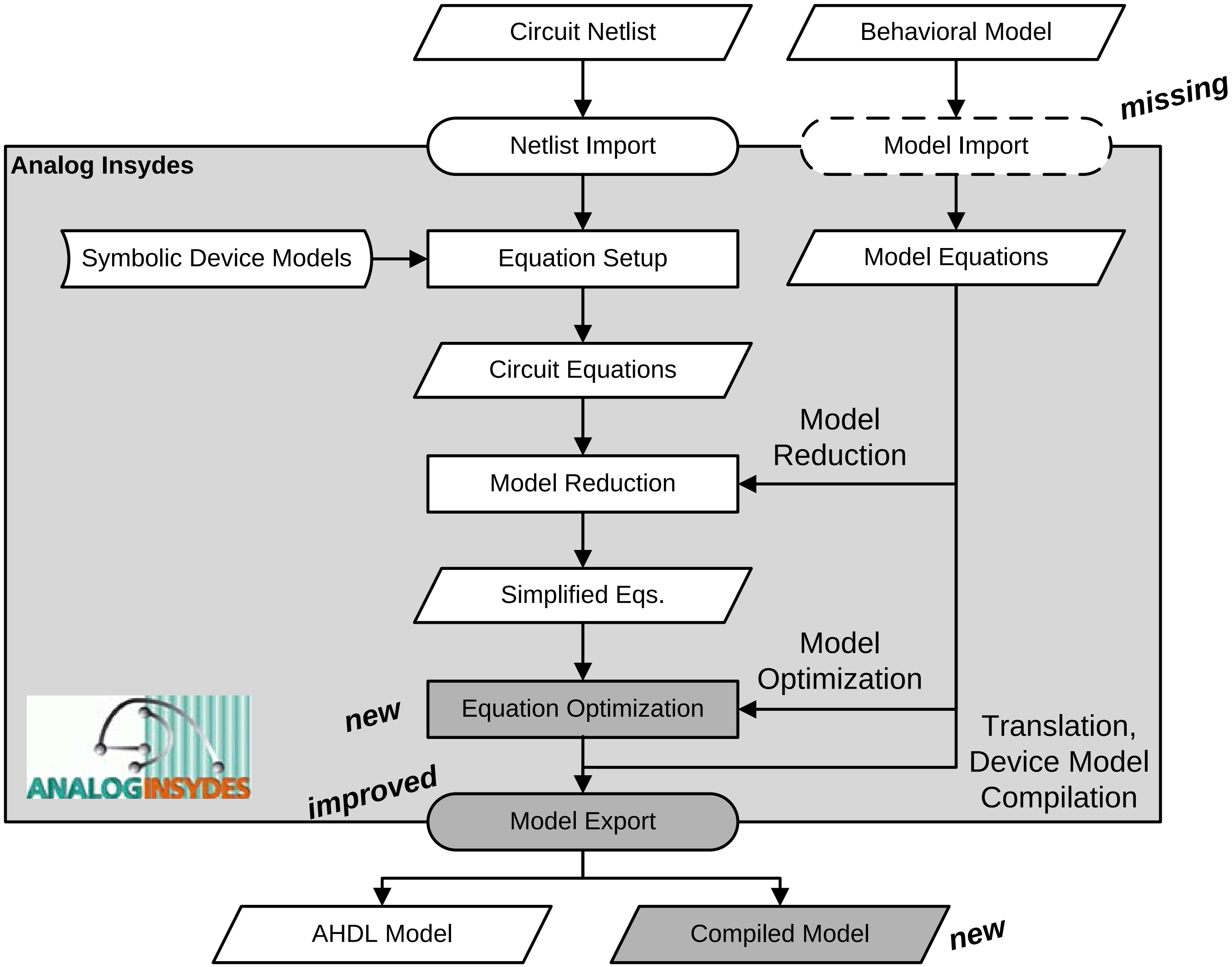
Performance Improvement



Outline

- Bottom-Up Modeling Through Symbolic Analysis
- Model Compilation
- Simulation Results
- **Summary**

Outlook



Circuit Netlist
Behavioral Model
missing
Netlist Import
Model Import
Analog Insydes
Symbolic Device Models
Equation Setup
Model Equations
Circuit Equations
Model Reduction
Model Reduction
Simplified Eqs.
Model Optimization
new
Equation Optimization
Translation, Device Model Compilation
ANALOGINSYDES
improved
Model Export
AHDL Model
Compiled Model
new

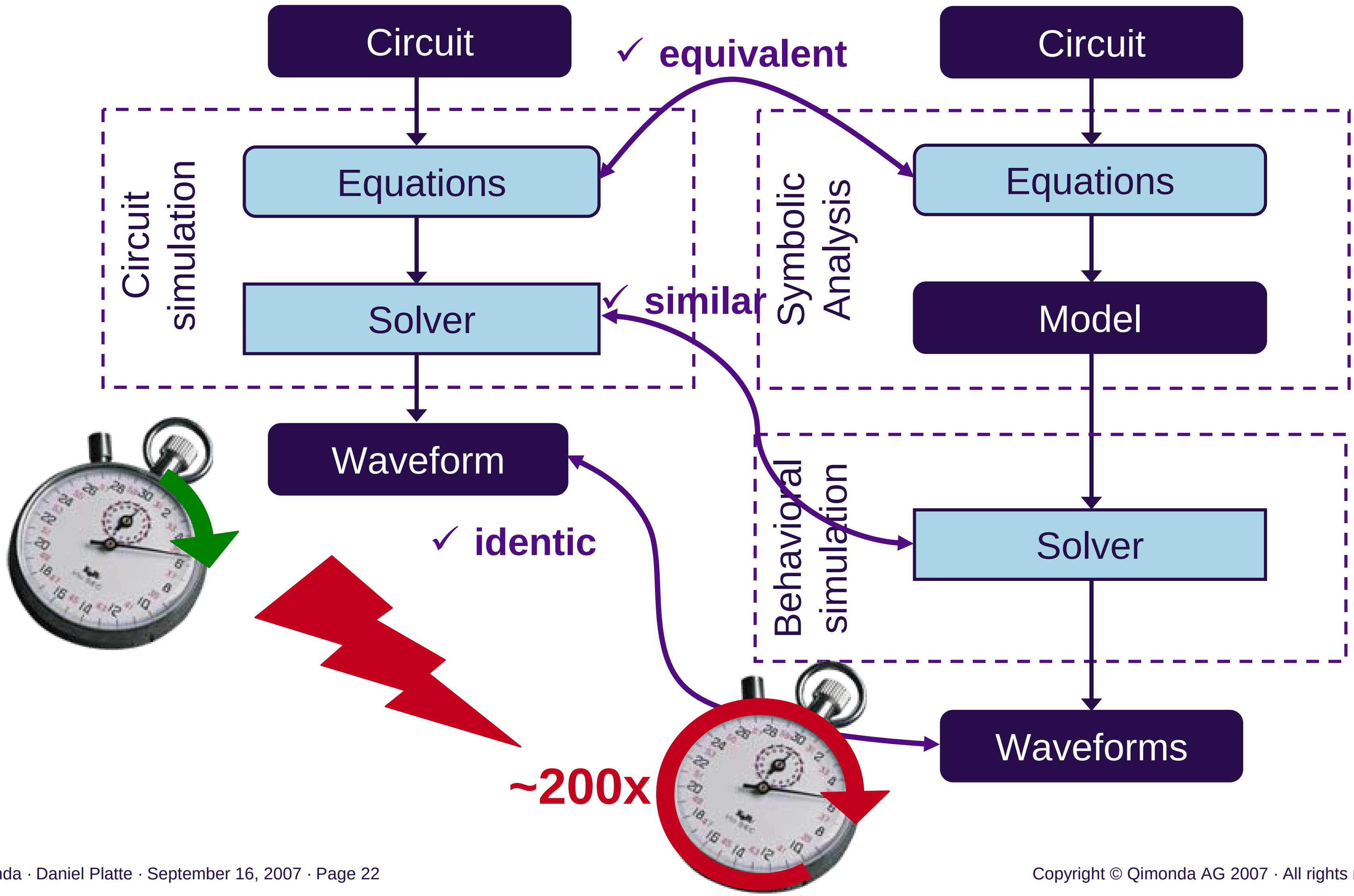
Summary

- Model generation based on a **symbolic analysis flow** using the toolbox Analog Insydes presented
- Such complex and high-dimensional AHDL-based models have **insufficient simulation performance**
- **New model compiler** for the Titan simulator developed
- **Efficient processing** of the generated models due to
  - Local solving with semi-symbolic Schur complement
  - Optimization techniques on DAE level
  - Close interaction between model and simulator
- **Performance improvement** by a factor of at least 8

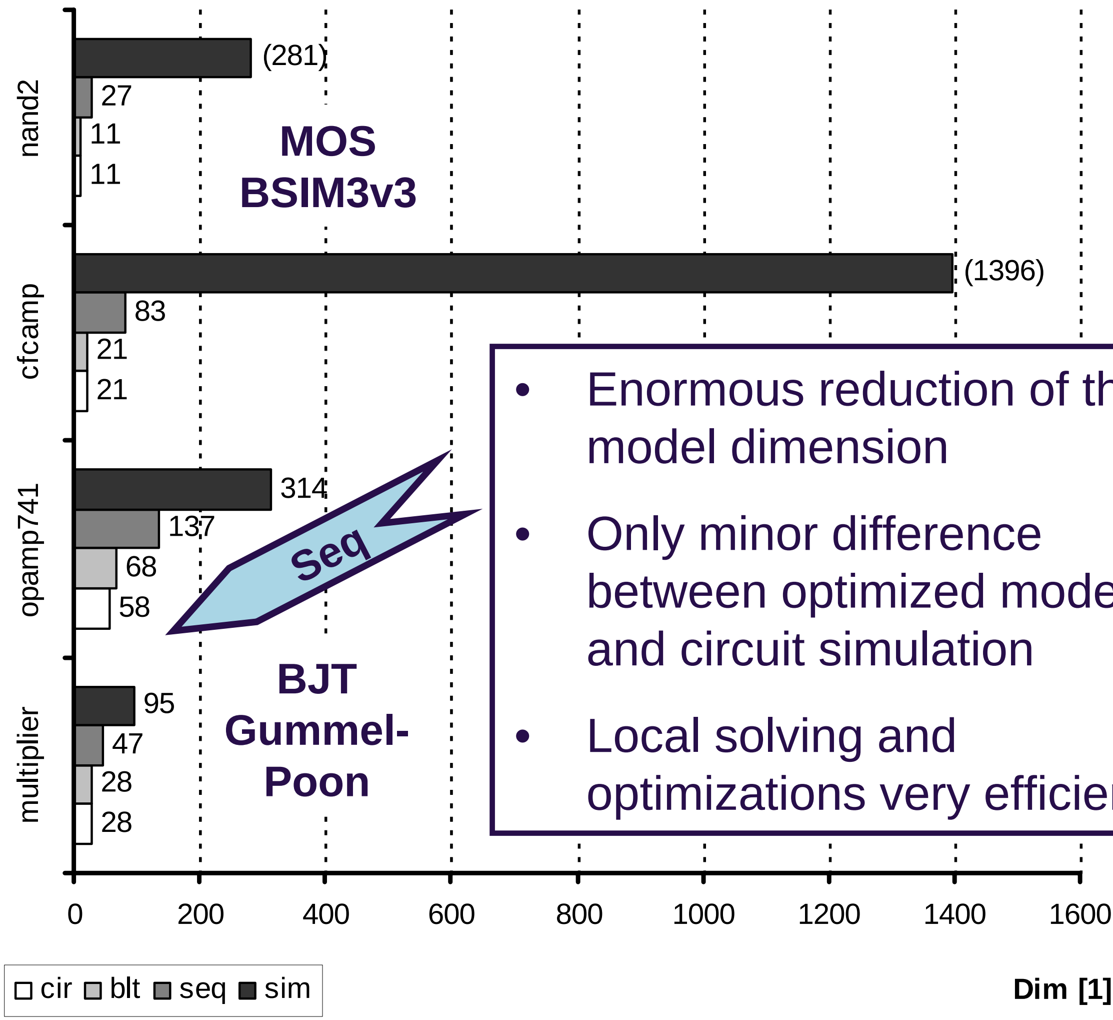
Thank you

The World’s Leading
Creative Memory Company

Performance Problem



Dimension of the Linearized Systems



- Enormous reduction of the model dimension
- Only minor difference between optimized model and circuit simulation
- Local solving and optimizations very efficient

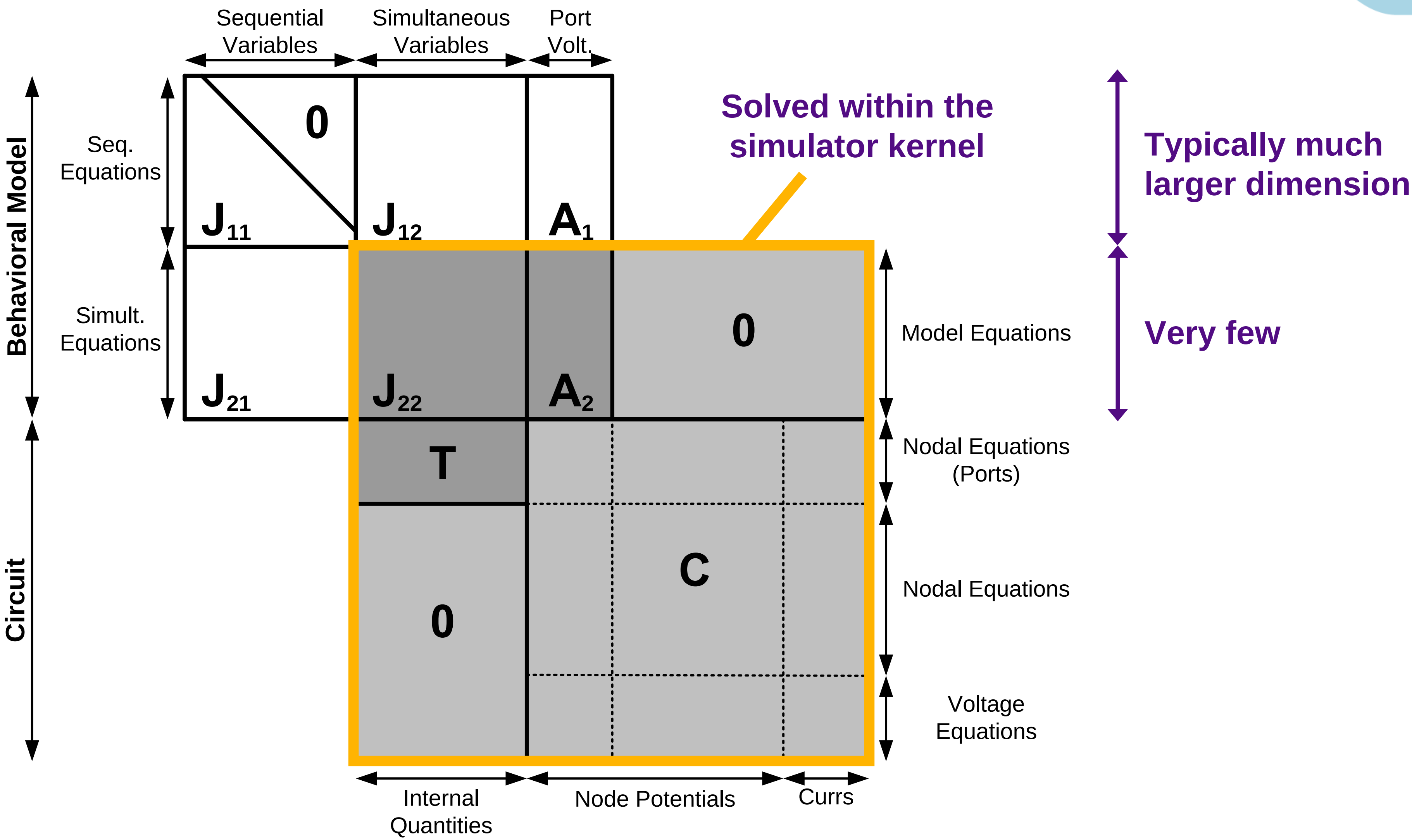
Sim: All model equations simultaneous

Seq: Sequential equations defined within device models

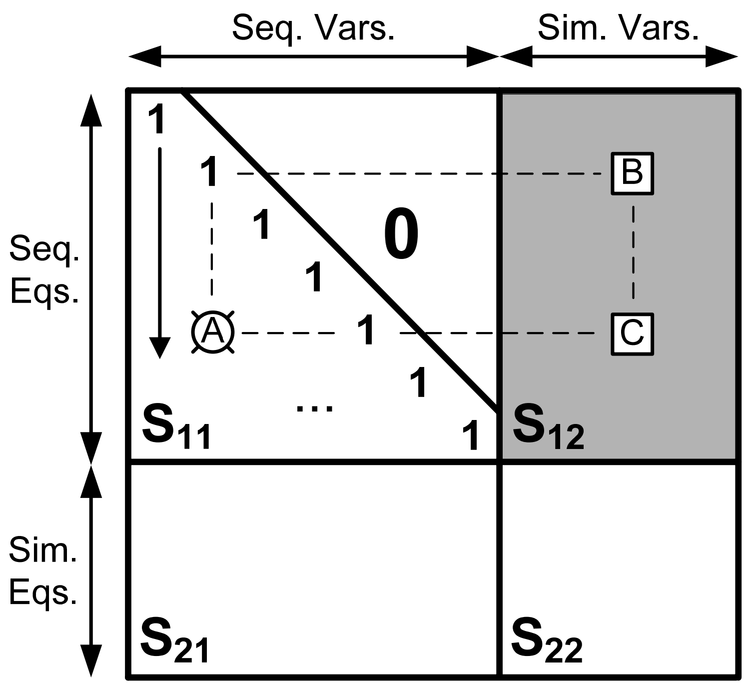
Blt: With identification of sequential equations

Cir: Dimension of the MNA equations in circuit simulation

Jacobian Matrix of Sequential DAEs

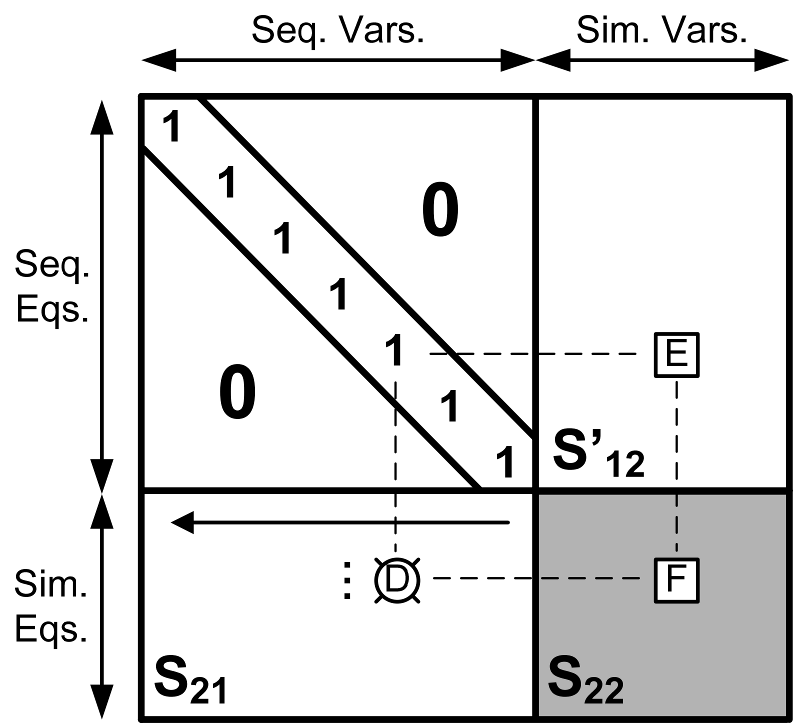


Semi-Symbolic Schur Complement



1) **Transform S_{11} to unity matrix**

➔ changes within S_{12}, fill-ins



2) **Eliminate S_{21}**

➔ update of S_{22}, fill-ins

DAE System with Sequential Structure

A system of differential algebraic equations (DAE)

$$f\left(\mathbf{y}, \mathbf{y}', \mathbf{x}, \mathbf{x}', t\right) = 0$$

with

$$\mathbf{y} \in R^m$$ sequential variables

$$\mathbf{x} \in R^n$$ simultaneous variables

$$f = \left\{f_{seq}, f_{sim}\right\}$$

$$f_{seq} : R^n \times R^m \times R \longrightarrow R^m$$ sequential equations

$$f_{sim} : R^n \times R^m \times R \longrightarrow R^n$$ simultaneous equations

is of sequential structure, if

$$\mathbf{y}_i = f_{seq,i}\left(y_1 \ldots y_{i-1}, y_1' \ldots y_{i-1}', \mathbf{x}, \mathbf{x}', t\right) \quad \text{for} \quad i = 1 \ldots n$$

Modeling DAE Systems with Seq. Structure

	Verilog-A

Sequential Equations	`// Analog procedural`
	`// assignment`
	`real vd;`
	`vd = -RS*X(I_A))/AREA-`
	`      X(V_K)+V(A);`

Simultaneous Equations	`// (Indirect) branch`
	`// contribution`
	`DAEVar V_G;`
	`X(V_G) <+ X(V_G)+id-0.115E-`
	`7*X(id_d1)-`
	`X(I_A)+cj*X(vd_d1);`

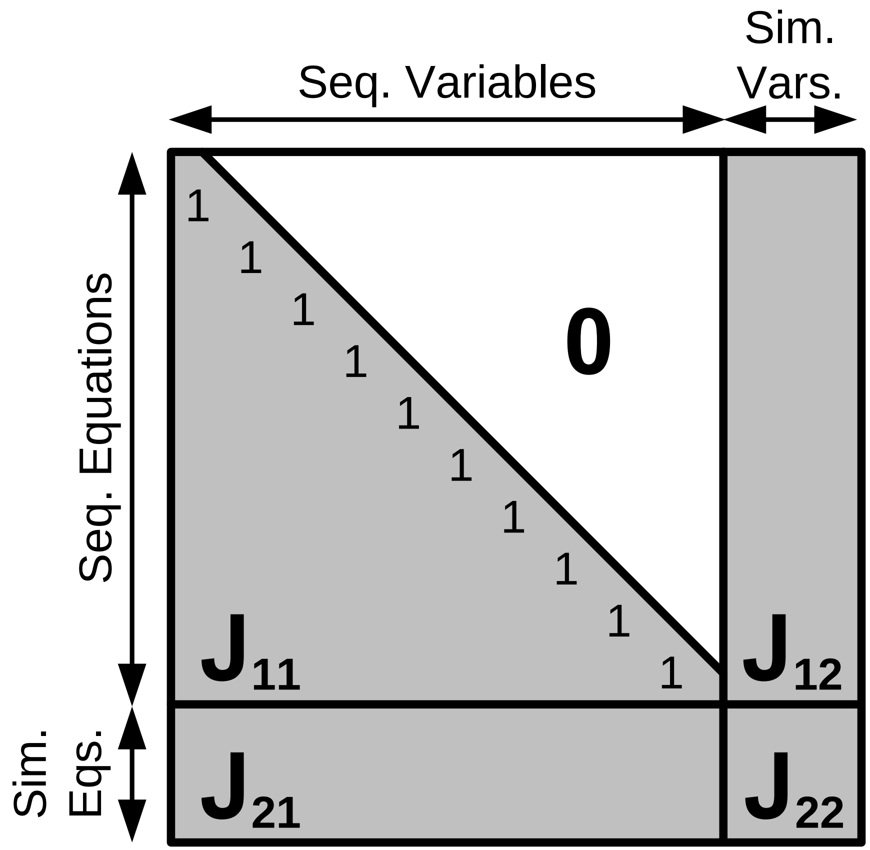
Modeling DAE Systems with Seq. Structure

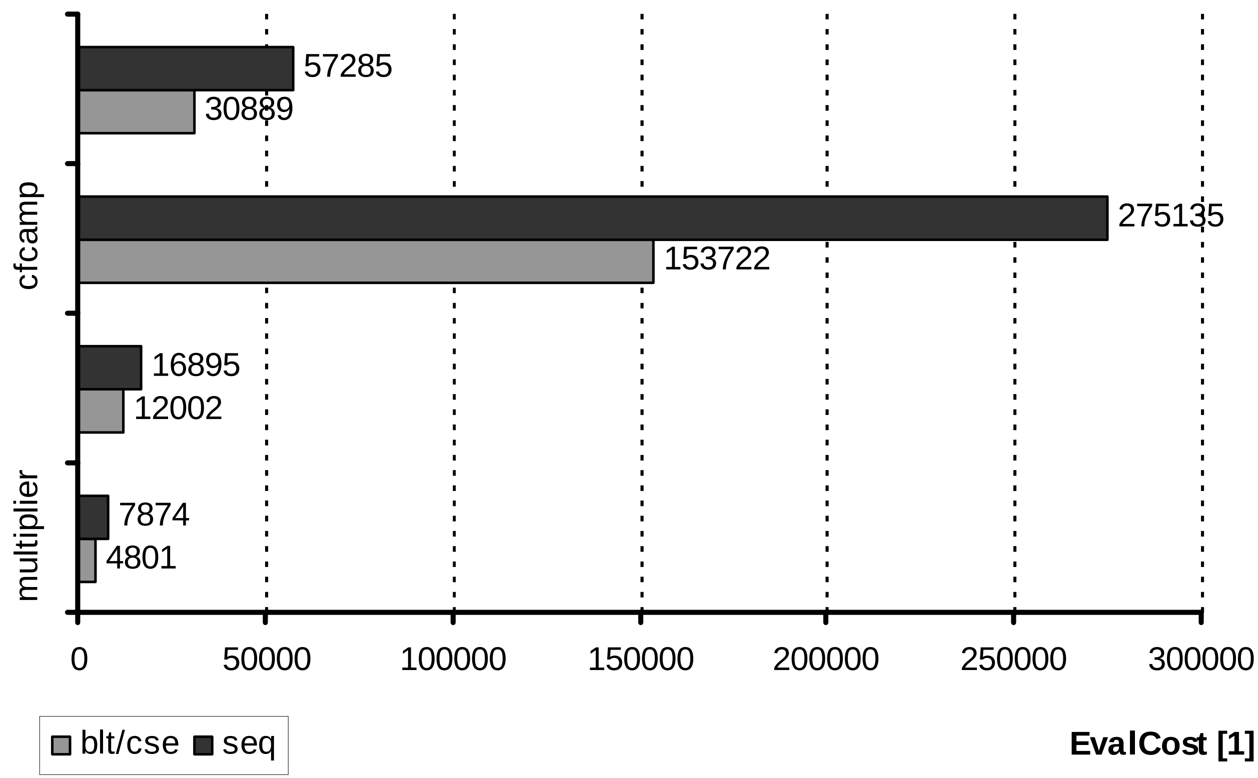
	Verilog-A	VHDL-AMS

Sequential Equations	`// Analog procedural`
	`// assignment`
	`real vd;`
	`vd = -RS*X(I_A))/AREA-`
	`     X(V_K)+V(A);`
		`-- Simultaneous`
		`-- procedural statement`
		`vd:=-RS*I_A/AREA-V_K+A;`
		`-- or: Function`
		`function vd (I_A,…) is`
		`…`

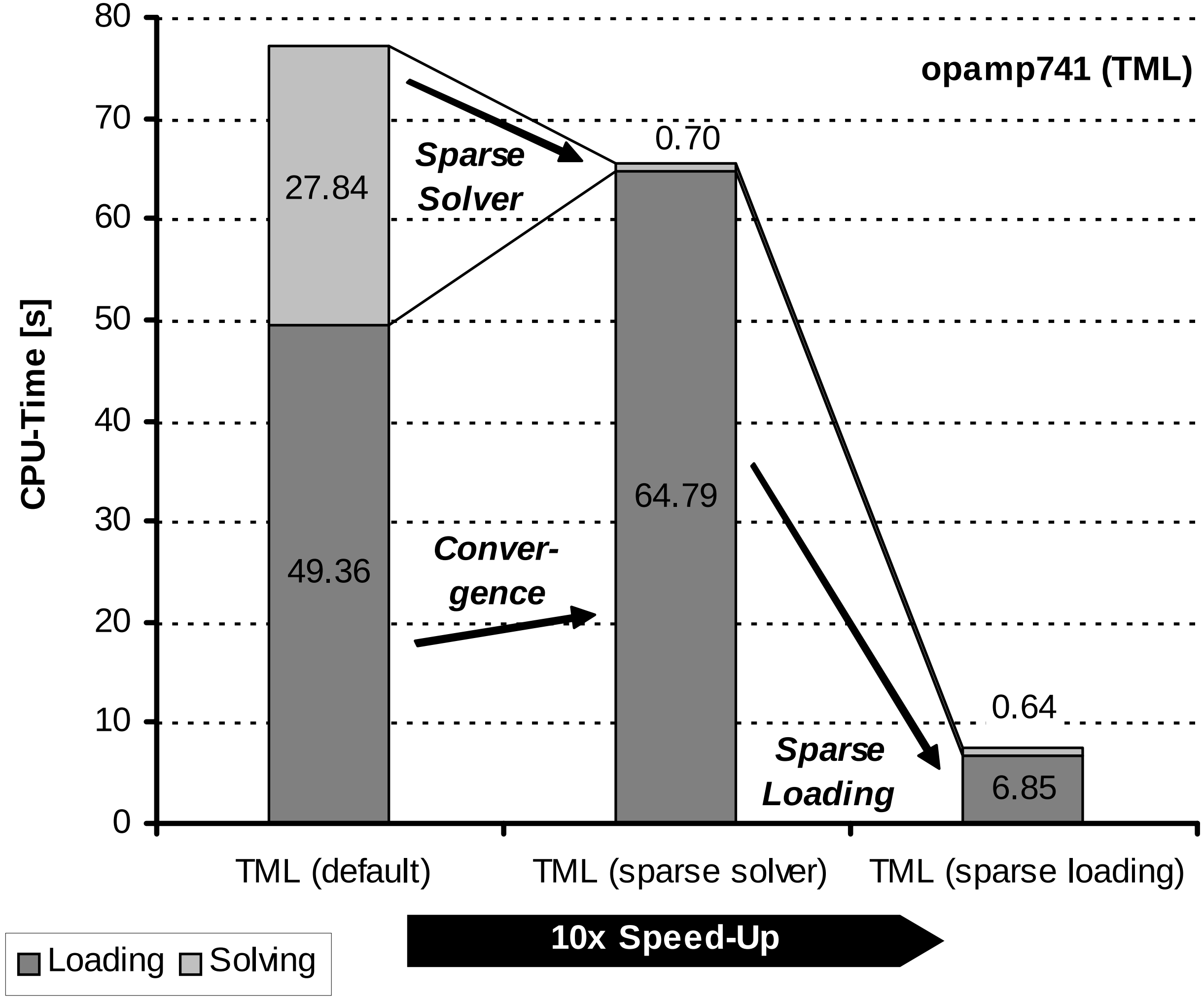
Simultaneous Equations	`// (Indirect) branch`
	`// contribution`
	`DAEVar V_G;`
	`X(V_G) <+ X(V_G)+id-`
	`0.115E-7*X(id_d1)-`
	`X(I_A)+cj*X(vd_d1);`
		`-- Simple simultaneous`
		`-- statement`
		`QUANTITY V_G : Voltage;`
		`I_A-ID-0.115E-7*ID_D1-`
		`CJ*VD_D1 == 0.0 Tolerance`
		`“Current”;`

Block Lower-Triangular Matrix




57285
30889
275135
153722
16895
12002
7874
4801
cfcamp
multiplier
0
50000
100000
150000
200000
250000
300000
blt/cse
seq
EvalCost [1]

Speed-Up for TML Models



Sequential Equations (Definition)

$$x_1 = G_1(y)$$
$$x_2 = G_2(y, x_1)$$
$$x_3 = G_3(y, x_1, x_2)$$
$$\dots$$
$$x_n = G_1(y, x_1, \dots, x_{n-1})$$

n sequential equations
n sequential variables $\underline{x}$

sequential variables as function of
simultaneous variables $\underline{y}$

$$\underline{x} = \underline{x}(\underline{y})$$

$$\underline{F}(\underline{x}, \underline{y}) = 0$$

m simultaneous equations
m simultaneous variables $\underline{y}$

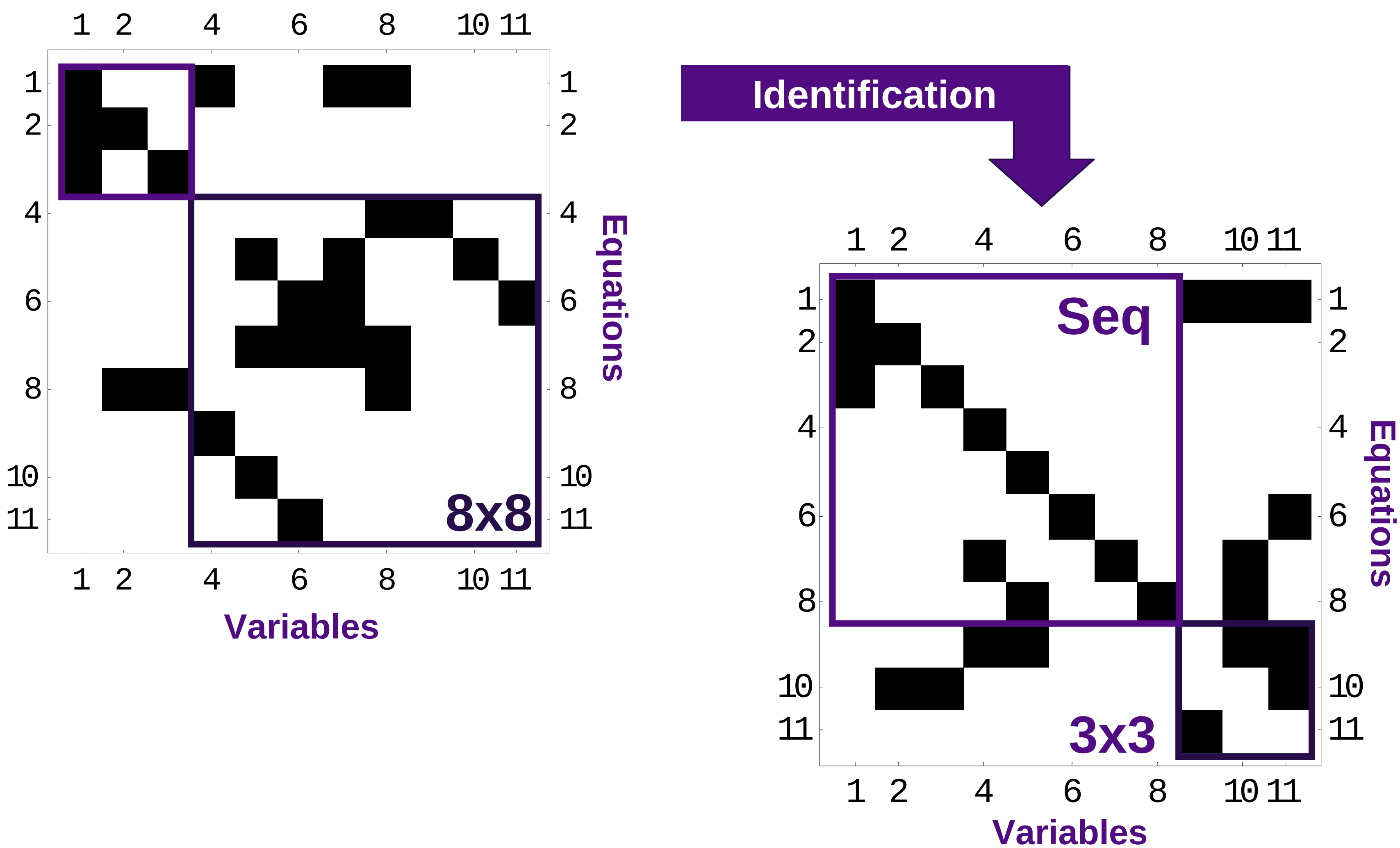
$$\Rightarrow \quad \tilde{F}(\underline{y}) = \underline{F}(\underline{x}(\underline{y}), \underline{y}) = 0$$

Advantages:

- Reduced dimension of Jacobian matrix
- Iterative solution for less variables

$$(n+m)\times(n+m) \rightarrow m\times m$$

Recognition of Sequential Equations



Simulator Comparison for opamp741