

Fast Dynamic Simulation of VLSI circuits using Reduced Order Compact Macromodel of Standard Cells

Shivam Priyadarshi, Nikhil Kriplani, T. Robert Harris, and
Michael B. Steer
North Carolina State University

2010 IEEE International Behavioral Modeling and Simulation Conference
24 September 2010

Overview

- ▷ **Motivation**
- ▷ **Reduced Order Macromodeling**
- ▷ **Macromodel Implementation examples**
 - ◆ CMOS Inverter Macromodel
 - ◆ CMOS NAND Macromodel
- ▷ **Results and Discussion**
- ▷ **Conclusion**

Motivation

- ▷ **Some applications require Long Dynamic Simulation**
 - ◆ Transient Electro-thermal simulation to see the impact of self heating of devices on circuit performance

- ▷ **Transistor-level simulation is challenging for such applications**
 - ◆ Extremely Time consuming
 - ◆ High Memory requirement

- ▷ **A Dynamic Simulation methodology is required which can**
 - ◆ Reduce computational and storage cost
 - ◆ Produce sufficiently accurate results

Motivation

- ▷ **Macromodel based simulation methodology**
 - ◆ An alternative to transistor-level simulation
 - ◆ In past, used for timing analysis of standard cell based VLSI circuits
 - ◇ Table look up models [1-3]
 - ◇ Current Source models [4-6]

- ▷ **Proposed Dynamic simulation methodology**
 - ◆ Uses physics based reduced order compact macromodels of standard cells in constructing large scale circuits
 - ◆ Suitable for applications where long duration dynamic simulation is required
 - ◇ Electro-thermal simulation to study the impact of transient thermal effects
 - ◆ Can be used for fast and accurate timing and power characterization of standard cells

Overview

- ▷ Motivation
- ▷ **Reduced Order Macromodeling**
- ▷ Macromodel Implementation examples
 - ◆ CMOS Inverter Macromodel
 - ◆ CMOS NAND Macromodel
- ▷ Results and Discussion
- ▷ Conclusion

Reduced Order Macromodeling

- ▷ **Reduced Order Macromodel of a circuit**
 - ◆ Preserve the input-output behavior
 - ◆ Reduces the complexity of the circuit
 - ◆ Can significantly reduce simulation run time and memory requirements

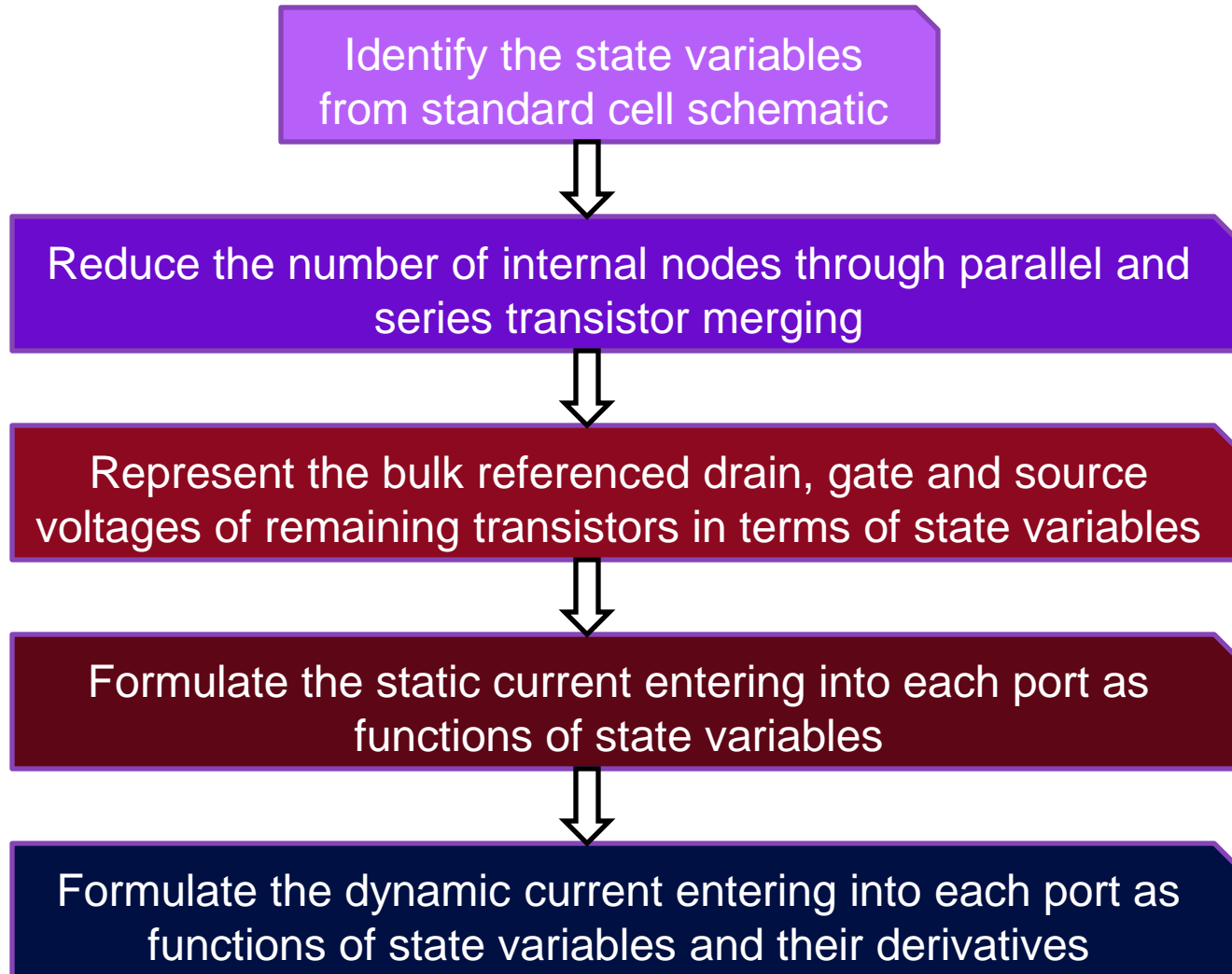
- ▷ **Developed reduced order macromodels of standard cells**
 - ◆ Describe the behavior using fewer number of state variables compared to equivalent transistor-level implementation
 - ◇ Reduction in state variables reduces complexity
 - ◆ Based on EKV MOSFET model equations
 - ◇ Physical basis make them accurate
 - ◆ Implemented in multi-physics simulator fREEDA

fREEDA: A Universal Circuit Simulator

- Multi-physics simulator: Concurrent EM, Electrical, Mechanical and Thermal Simulations are possible.
- Follow State Space simulation approach
 - Port voltages and currents are expressed as functions of state variables and their derivatives.
- Supports high dynamic range Transient, Harmonic balance, DC and AC analysis
- Enables Rapid model development
 - Uses Object Oriented Paradigm (C++) : Drastically reduces the amount of code required to implement a model
 - Uses Automatic Differentiation Packages: Eliminates the need of coding the derivatives

$$v_{\text{NL}}(t) = u \left[x(t), \frac{dx}{dt}, \dots, \frac{d^m x}{dt^m}, x_{\text{D}}(t) \right]$$
$$i_{\text{NL}}(t) = w \left[x(t), \frac{dx}{dt}, \dots, \frac{d^m x}{dt^m}, x_{\text{D}}(t) \right]$$

Macromodel development flow



Overview

- ▷ Motivation
- ▷ Reduced Order Macromodeling
- ▷ **Macromodel Implementation examples**
 - ◆ CMOS Inverter Macromodel
 - ◆ CMOS NAND Macromodel
- ▷ Results and Discussion
- ▷ Conclusion

CMOS Inverter Macromodel

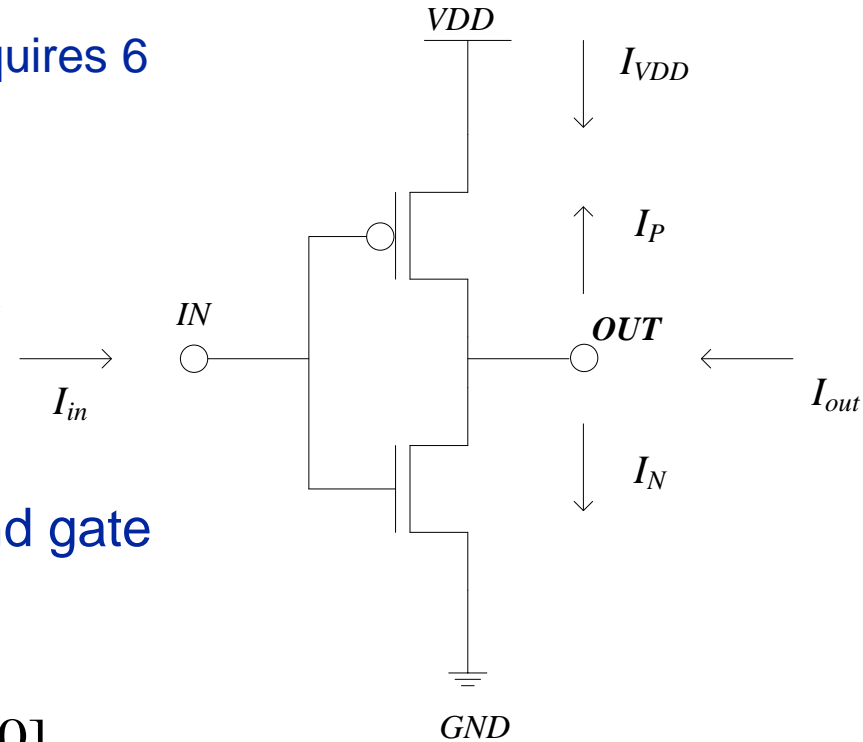
- Uses 3 state variables
 - Transistor level implementation requires 6 state variables

$$x[0] = VDD, \quad x[1] = V_{in}, \quad \text{and} \quad x[2] = V_{out}$$

- Represent bulk referenced drain and gate voltages in terms of state variables

$$V_{db_N} = x[2], \quad V_{db_P} = x[2] - x[0]$$

$$V_{gb_N} = x[1], \quad V_{gb_P} = x[1] - x[0]$$



CMOS Inverter Macromodel

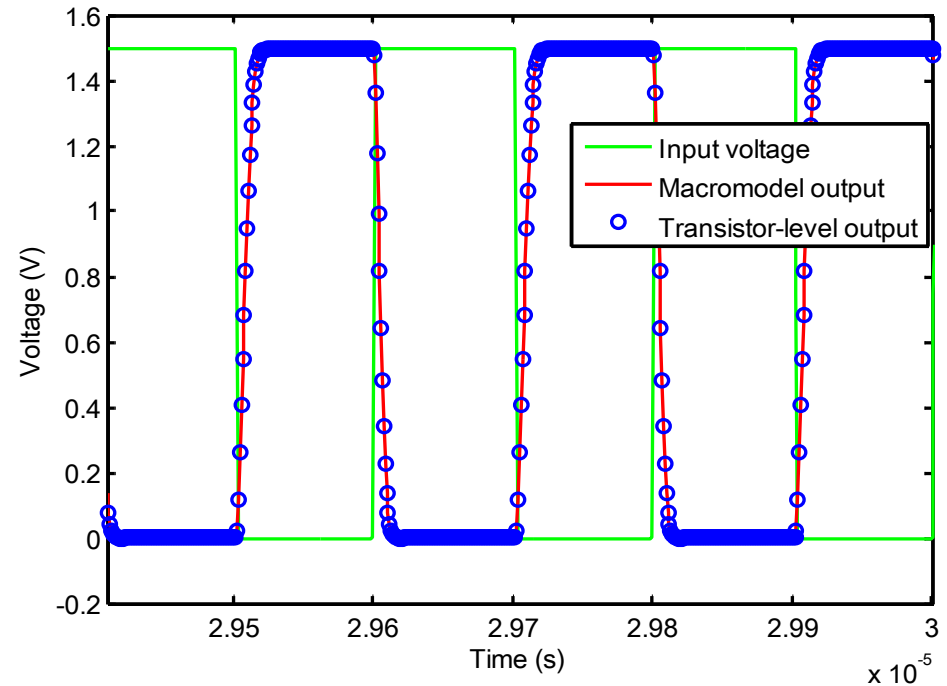
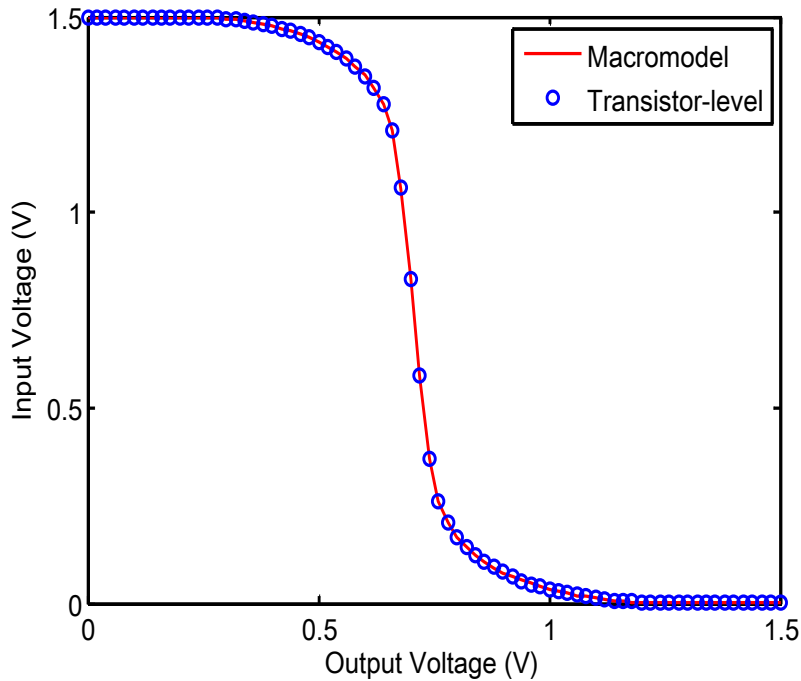
- Formulate the total current entering into each port
 - Dynamic current is calculated by taking the time derivative of charge

$$I_{\text{in}} = \frac{dQ_{\text{gN}}}{dt} + \frac{dQ_{\text{gP}}}{dt} \quad I_{\text{out}} = I_{\text{N}} + I_{\text{P}} + \frac{dQ_{\text{dN}}}{dt} + \frac{dQ_{\text{dP}}}{dt} \quad I_{\text{VDD}} = -(I_{\text{P}} + \frac{dQ_{\text{VDD}}}{dt})$$

- Based on EKV MOSFET model equations, static currents and charges are formulated as functions of bulk referenced drain and gate voltages of NMOS and PMOS

$$\{I_{\text{N}}, I_{\text{P}}, Q_{\text{gN}}, Q_{\text{gP}}, Q_{\text{dN}}, Q_{\text{dP}}, Q_{\text{VDD}}\} = f(V_{\text{db}_\text{N}}, V_{\text{db}_\text{P}}, V_{\text{gb}_\text{N}}, V_{\text{gb}_\text{P}})$$

CMOS Inverter Macromodel



Comparison of DC transfer characteristic and transient characteristic of the Inverter macromodel with transistor-level implementation on 150 nm CMOS process

CMOS NAND Macromodel

- Uses 4 state variables
 - Transistor level implementation requires 12 state variables

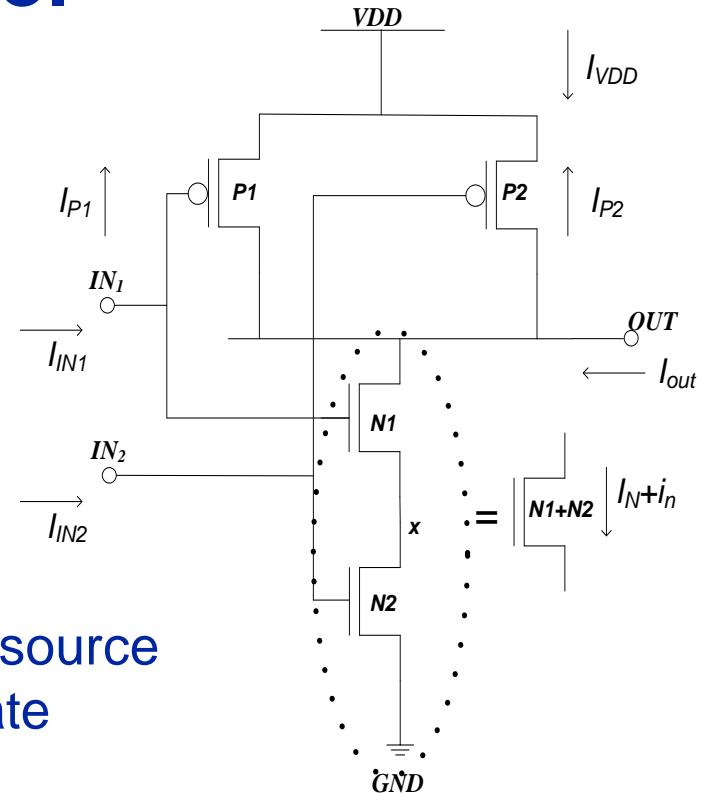
$$x[0] = VDD, x[1] = V_{in1}, x[2] = V_{in2}, \text{ and } x[3] = V_{out}$$

- Merge series transistors in single transistor

- Represent bulk referenced drain, gate and source voltages of PMOS transistors in terms of state variables

$$V_{db_P1} = x[3] - x[0], V_{gb_P1} = x[1] - x[0], \text{ and } V_{sb_P1} = 0$$

$$V_{db_P2} = x[3] - x[0], V_{gb_P2} = x[2] - x[0], \text{ and } V_{sb_P2} = 0$$



CMOS NAND Macromodel

- Depending upon the input voltages, it is assumed that one of the NMOS transistors is conducting, i.e. acting as a resistance and current through the other NMOS transistor is flowing through the series chain

if $(x[1] > x[2])$

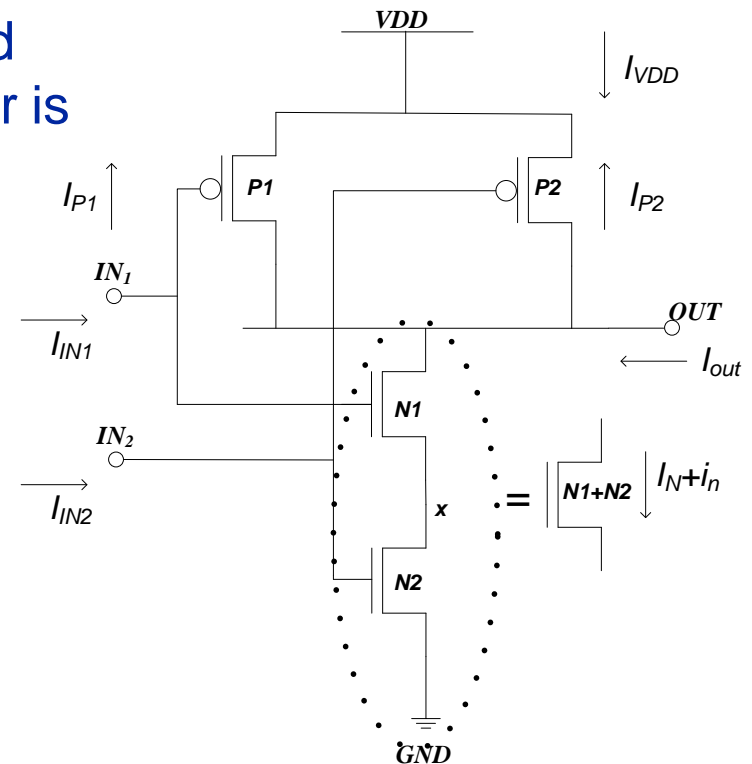
$V_{db_N2} = x[3]$, $V_{gb_N2} = x[2]$, and $V_{sb_N2} = 0$

$\beta_{eq} = \beta_1 \beta_2 / (\beta_1 + \beta_2)$, and $i_n = \frac{dQ_{dN2}}{dt}$

else

$V_{db_N1} = x[3]$, $V_{gb_N1} = x[1]$, and $V_{sb_N1} = 0$

$\beta_{eq} = \beta_1 \beta_2 / (\beta_1 + \beta_2)$, and $i_n = \frac{dQ_{dN1}}{dt}$



CMOS NAND Macromodel

- Formulate the current entering into each port

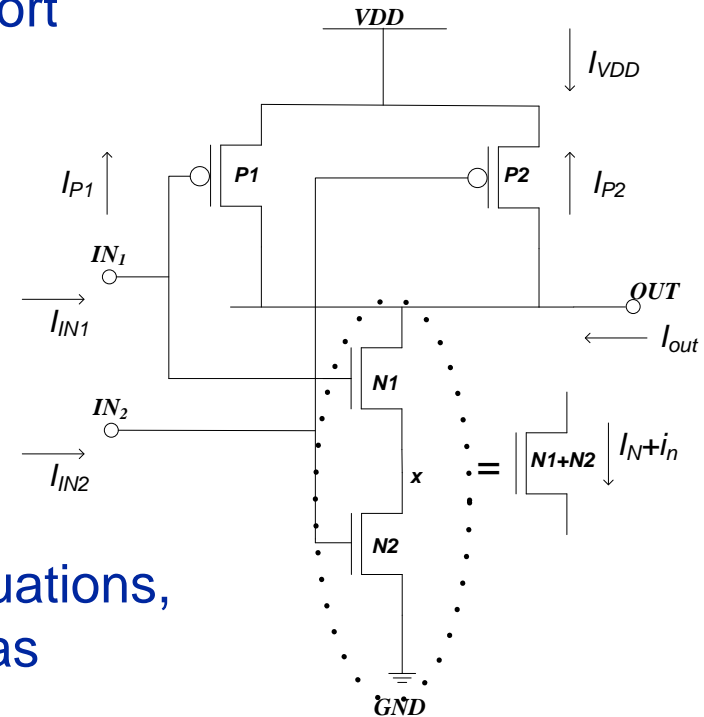
$$I_{out} = I_{P1} + I_{P2} + I_N + i_n + \frac{dQ_{dP1}}{dt} + \frac{dQ_{dP2}}{dt}$$

$$I_{VDD} = -\left(I_{P1} + I_{P2} + \frac{dQ_{VDD}}{dt} \right)$$

$$I_{IN1} = \frac{dQ_{gN1}}{dt} + \frac{dQ_{gP1}}{dt}, \text{ and } I_{IN2} = \frac{dQ_{gN2}}{dt} + \frac{dQ_{gP2}}{dt}$$

- Finally based on EKV MOSFET model equations, static currents and charges are formulated as functions of the state variables

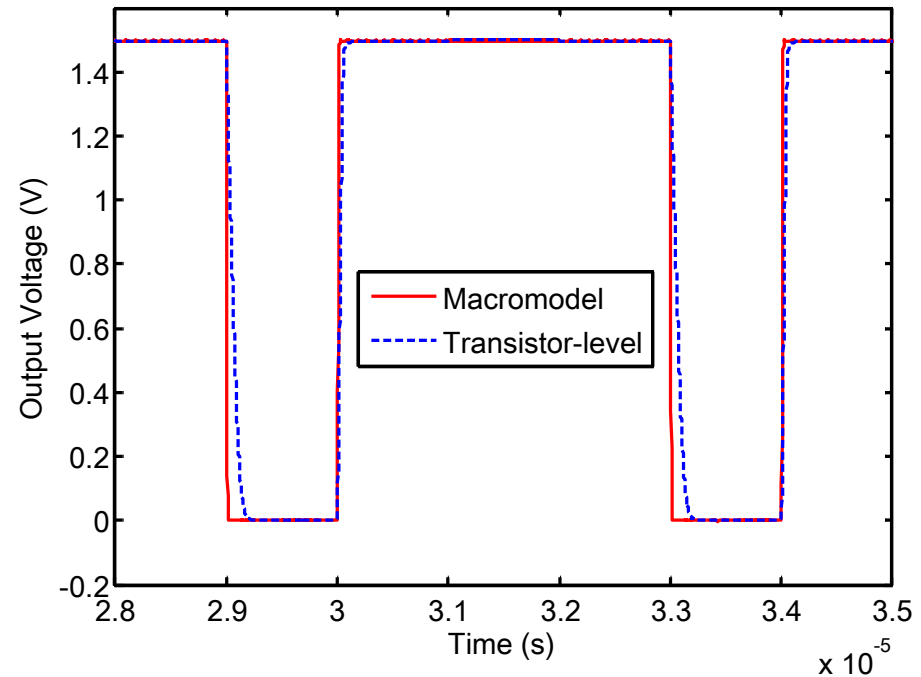
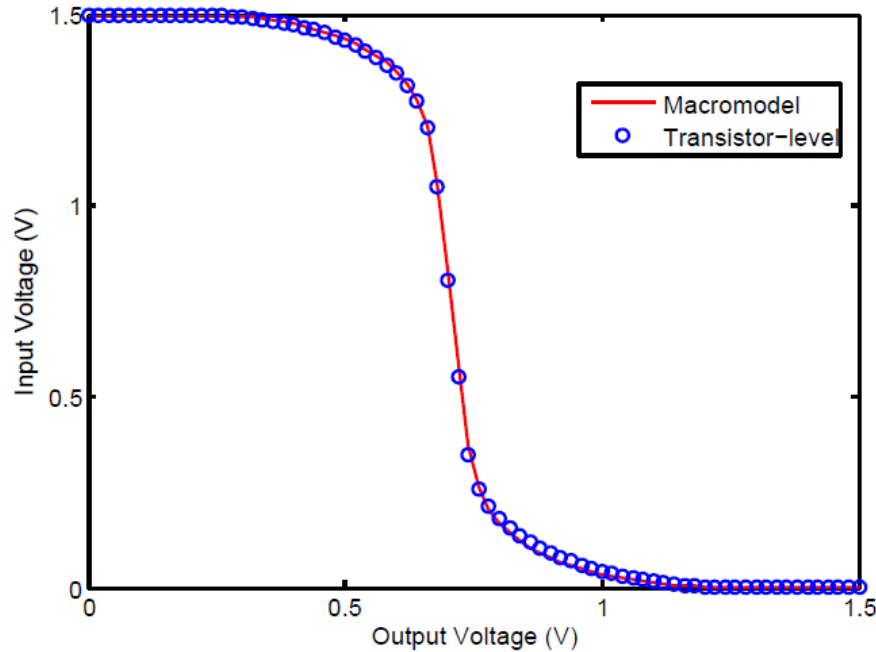
$$\{ I_{P1-P2}, I_N, Q_{VDD}, Q_{dP1-P2}, Q_{dN1-N2}, Q_{gP1-P2}, Q_{gN1-N2} \} = f(x[0-3])$$



CMOS NAND Macromodel

- Macromodel is completely parameterized in terms of process and geometry parameters such as oxide thickness, junction depth, effective channel length, width, and channel doping.
- All the temperature dependent device parameters are formulated as function of temperature
 - Can easily be transformed to an electro-thermal model by introducing temperature as an additional state variable
- Small geometry effects such as channel length modulation, source drain charge sharing, velocity saturation are modeled based on EKV MOSFET formulation

CMOS NAND Macromodel



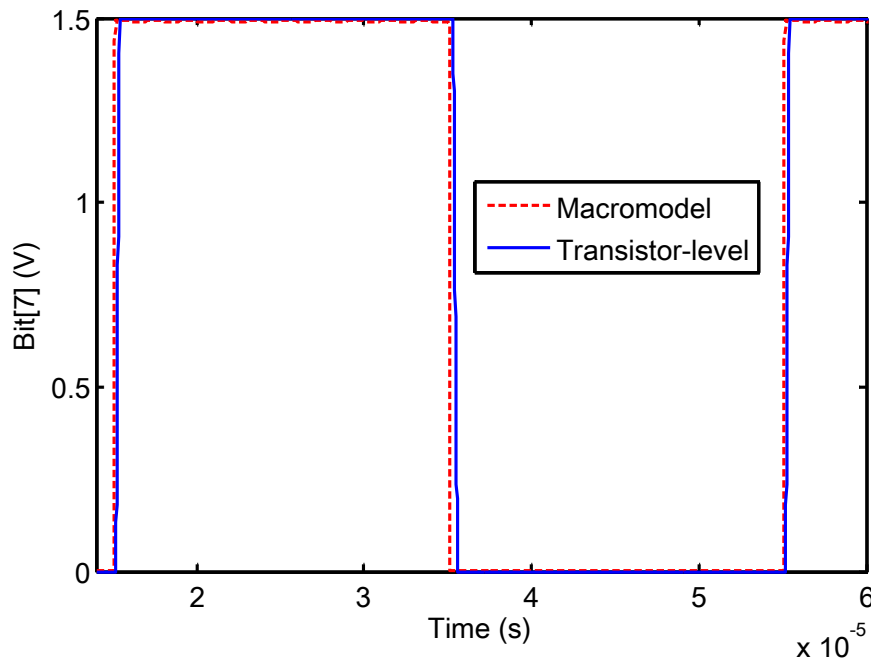
Comparison of DC transfer characteristic and transient characteristic of the NAND macromodel with transistor-level implementation on 150 nm CMOS process

Overview

- ▷ Motivation
- ▷ Reduced Order Macromodeling
- ▷ Macromodel Implementation examples
 - ◆ CMOS Inverter Macromodel
 - ◆ CMOS NAND Macromodel
- ▷ **Results and Discussion**
- ▷ Conclusion

Results and Discussion

- ▷ Reduced order macromodels of more complex cells such as XOR, Latch, Adder and D-Flip-Flop are built.
- ▷ Macromodels are based on device equations
 - ◆ Produce results which are in excellent agreement with transistor-level simulation



Standard Cells	Delay error (%)
Inverter	0.01
2-Input NAND	0.11
SR Latch	0.18
D-Flip-Flop	0.33
8-bit Shift Register	0.80

Transient simulation result of 8-bit shift register

Results and Discussion

- ▷ **The simulation time and memory usage of the macromodel and equivalent transistor-level implementations are compared by running the transient simulation in fREEDA**
- ▷ **State variable based fixed time step time marching transient analysis method is used**
- ▷ **The simulations are performed on a 3 GHz Intel Xeon server with 32 GB of RAM**
- ▷ **Both kinds of circuits, combinational and sequential, are considered for comparison.**

Results and Discussion

Design (Stop time for transient)	Macromodel (fREEDA)		Transistor-level (fREEDA)	
	# State variable	Runtime	# State variable	Runtime
Inverter(100 μ s)	3	6s	6	9s
2-Input NAND(100 μ s)	4	9s	12	37s
SR Latch(100 μ s)	8	19s	24	1m 6s
D-Flip-Flop(100 μ s)	35	3m 16s	99	28m 2s
8-bit Shift Register(10 μ s)	280	17m 55s	792	9h 5m
Freq. multiplier-divider chain (10 μ s)	620	3h 48m	1740	380h 5m

Freq. multiplier-divider chain –

- 15 frequency multipliers followed by 10 frequency dividers
- Represents a heat spot of 3DIC chip presented in [7]

Design	Reduction in runtime	Reduction in memory usage
Inverter	1.50x	1.50x
2-Input NAND/NOR	4.11x	1.62x
SR Latch	4.00x	1.75x
D-Flip-Flop	8.58x	2.00x
8-bit Shift Register	30.41x	2.75x
Freq. multiplier-divider chain	100.02x	2.80x

Results and Discussion

- General Trend : More speed-up over transistor–level simulation for large scale circuits
- Also dependent upon type of circuit
 - SR Latch and 2-Input NAND shows almost same speed-up
 - Feedback circuits tend to take more time to converge
- Comparison with Sparse matrix based simulation program (HSPICE)
 - The macromodel based simulation is 14 times faster than the HSPICE transistor-level simulation

Design	Reduction in runtime	Reduction in memory usage
Inverter	1.50x	1.50x
2-Input NAND/NOR	4.11x	1.62x
SR Latch	4.00x	1.75x
D-Flip-Flop	8.58x	2.00x
8-bit Shift Register	30.41x	2.75x
Freq. multiplier-divider chain	100.02x	2.80x

Design (Stop time)	Macromodel (fREEDA)	Transistor-level (HSPICE)
Freq. multiplier-divider chain (10 μ s)	3h 48m	53h

Conclusion

- ▷ **Reduced order macromodels of various standard cells are developed using which large scale circuits can be constructed**
 - ◆ Macromodels are implemented with lesser number of state variables compared to equivalent transistor level implementation
 - ◆ Results in significant speed-up over transistor-level simulation for large scale circuits. Examples showing 1.5x-100x speed-up are presented
 - ◆ Further speed improvements can be obtained by integrating fast spice techniques in fREEDA
 - ◆ Reduces memory usages. Examples showing 1.5x-2.8x reduction in memory usage are presented
- ▷ **Macromodels are physics based**
 - ◆ Produces results in excellent agreement with transistor-level simulation
- ▷ **Macromodels are suitable for long duration dynamic simulation of VLSI circuits**
 - ◆ Electro-thermal simulation
 - ◆ Functional Verification

References

- [1] L. Brocco, S. McCormick, and J. Allen, “Macromodeling CMOS circuits for timing simulation,” *IEEE Trans. Computer-Aided Design*, vol. 7, no. 12, pp. 1237 –1249, Dec 1988.
- [2] F.C. Chang, C.F. Chen, and P. Subramaniam, “An accurate and efficient gate level delay calculator for MOS circuits,” in *Proc. of 25th Design Automation Conference*, 1988, pp. 282 – 287.
- [3] C. Forzan, B. Franzini, and C. Guardiani, “Accurate And Efficient Macromodel Of Submicron Digital Standard Cells,” in *Proc. of the 34th Design Automation Conference*, 1997, pp. 633 – 637.
- [4] C. Amin, C. Kashyap, N. Menezes, K. Killpack, and E. Chiprout, “A multi-port current source model for multiple-input switching effects in CMOS library cells,” in *Proc. of 43rd Conference on Design Automation*, 2006, pp. 247 –252.
- [5] C. Kashyap, C. Amin, N. Menezes, and E. Chiprout, “A nonlinear cell macromodel for digital applications,” in *Proc. of IEEE/ACM International Conference on Computer-Aided Design*, 2007, pp. 678 –685.
- [6] C. Knoth, V. Kleeberger, P. Nordholz, and U. Schlichtmann, “Fast and waveform independent characterization of current source models,” in *Proc. of IEEE Behavioral Modeling and Simulation Workshop*, 2009, pp. 90 –95.
- [7] F. Akopyan, C. Otero, D. Fang, S. Jackson, and R. Manohar, “Variability in 3-D integrated circuits,” in *Proc. of IEEE Custom Integrated Circuits Conference*, 2008, pp. 659 –662.

Thank You