Runtime leakage power estimation technique for combinational circuits

Yu-Shiang Lin, Dennis Sylvester

EECS, University of Michigan

Acknowledgement

STARC

Motivation

- Subthreshold leakage accounts for nonnegligible portion of total energy consumption for advanced digital CMOS technologies
- Traditional methods that uses steady state leakage analysis is underestimating the leakage contribution due to the transient behavior
- Develop a runtime leakage estimation method to provide more accurate leakage information at synthesis level

Outline

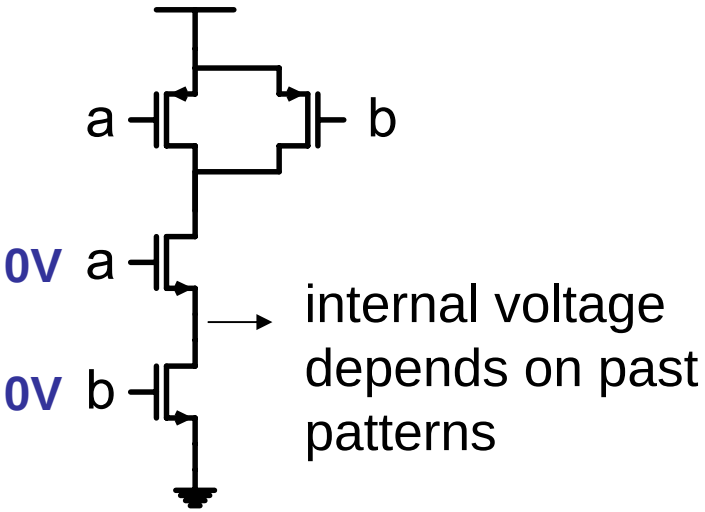
- Traditional approach and prior works
- Transient behavior of subthreshold leakage
- Proposed dynamic estimation approach
- Static estimation based on state probability
- Results
- Conclusions

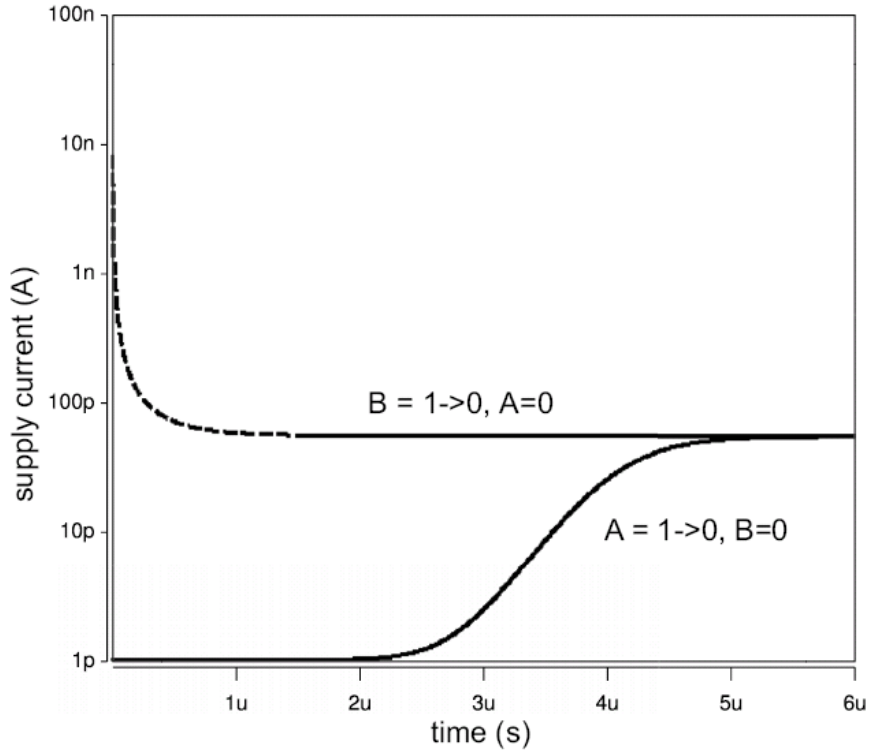
Prior works on subthresold leakage estimation

- Solving for transistor stacks
  - computes subthreshold leakage with stack transistors by 3D lookup table [Sirichotiyakul, VLSI 02]
  - Empirical modeling: estimating leakage by computing the total width of the circuit and effective scaling factor for input patterns [Jiang, VLSID 02]
- Circuit level estimation
  - Use linear regression model to find out the total leakage corresponding to a certain library [Kumar, VLSID 02]

Transient behavior of subthreshold leakage

- Internal voltage of transistor stack can be varied because of different switching patterns.





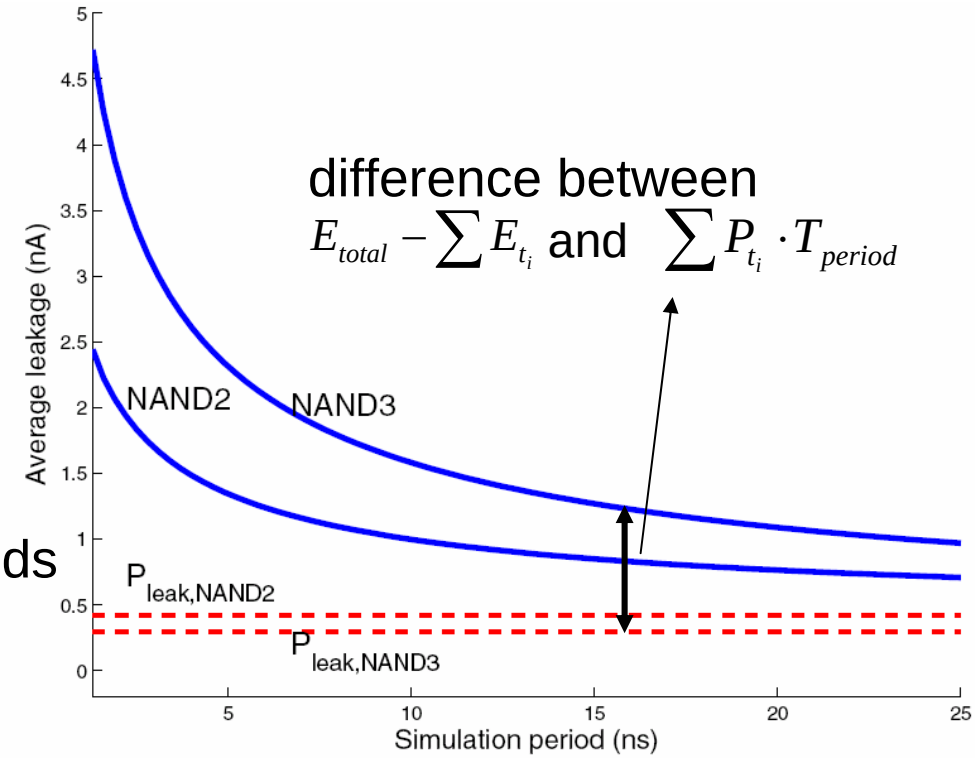
Subthresold leakage as a function of time

- High level synthesis tool estimates power by calculating:
  - Switching pattern based dynamic energy
  - State dependent leakage power

$$E_{total} = \sum E_{t_i} + \sum P_{t_i} \cdot T_{period}$$

dynamic energy for each input switching event

leakage power depends only on input state

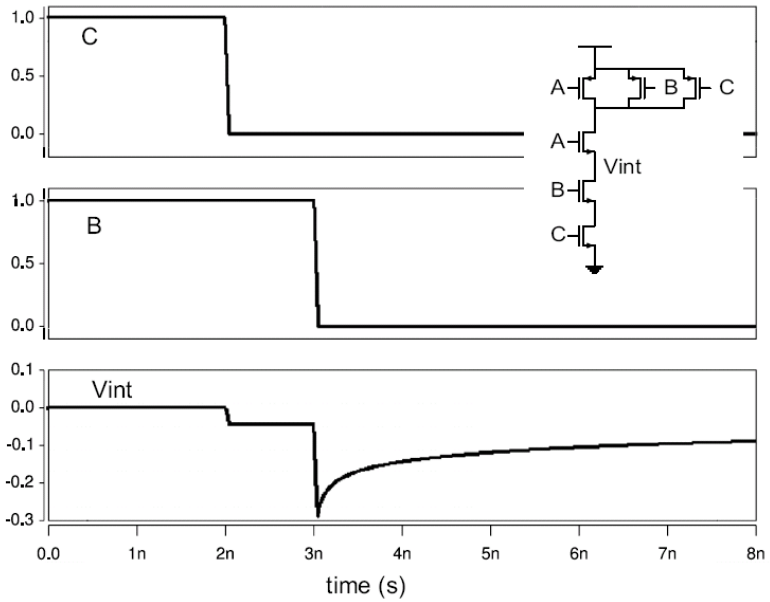


Transient behavior of internal node voltage for transistor stacks

- Internal node voltage of a transistor stack is dependent on all the past states on the gate, and the amount of time it stays at such state as well.
- General runtime leakage patterns can be expressed as an exponential function of time right after input switching events happens

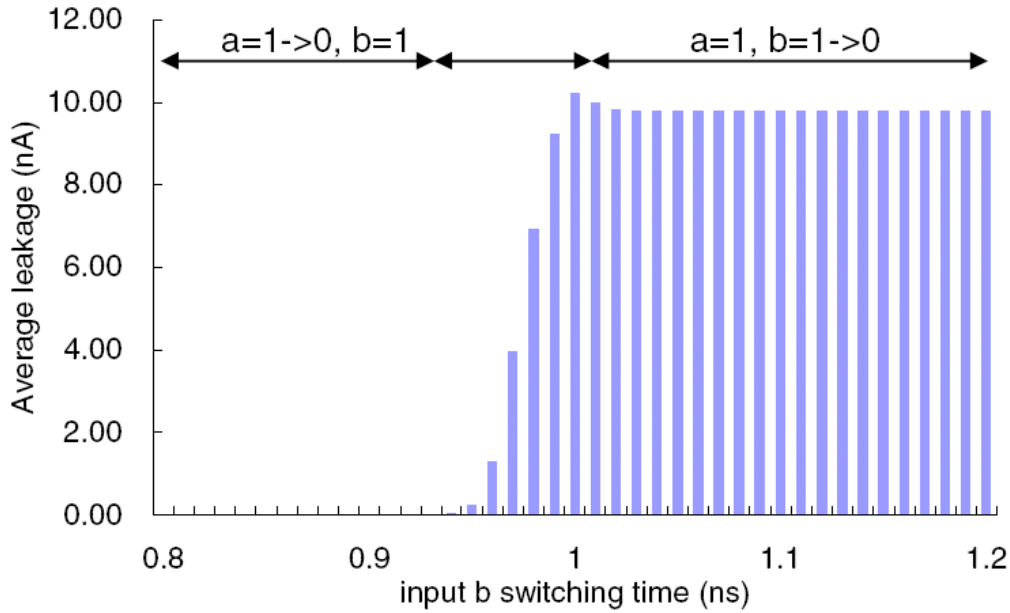
$$V(t) = nV_T \cdot \ln[\frac{k}{nV_T} \cdot t]$$

where V(t) is the internal voltage



Other factors that impact internal voltage estimation

- Considering simultaneous inputs switching, STA is required as well as inputs slope information
- Signal glitch from the inputs may further complicate the tracking of internal voltages



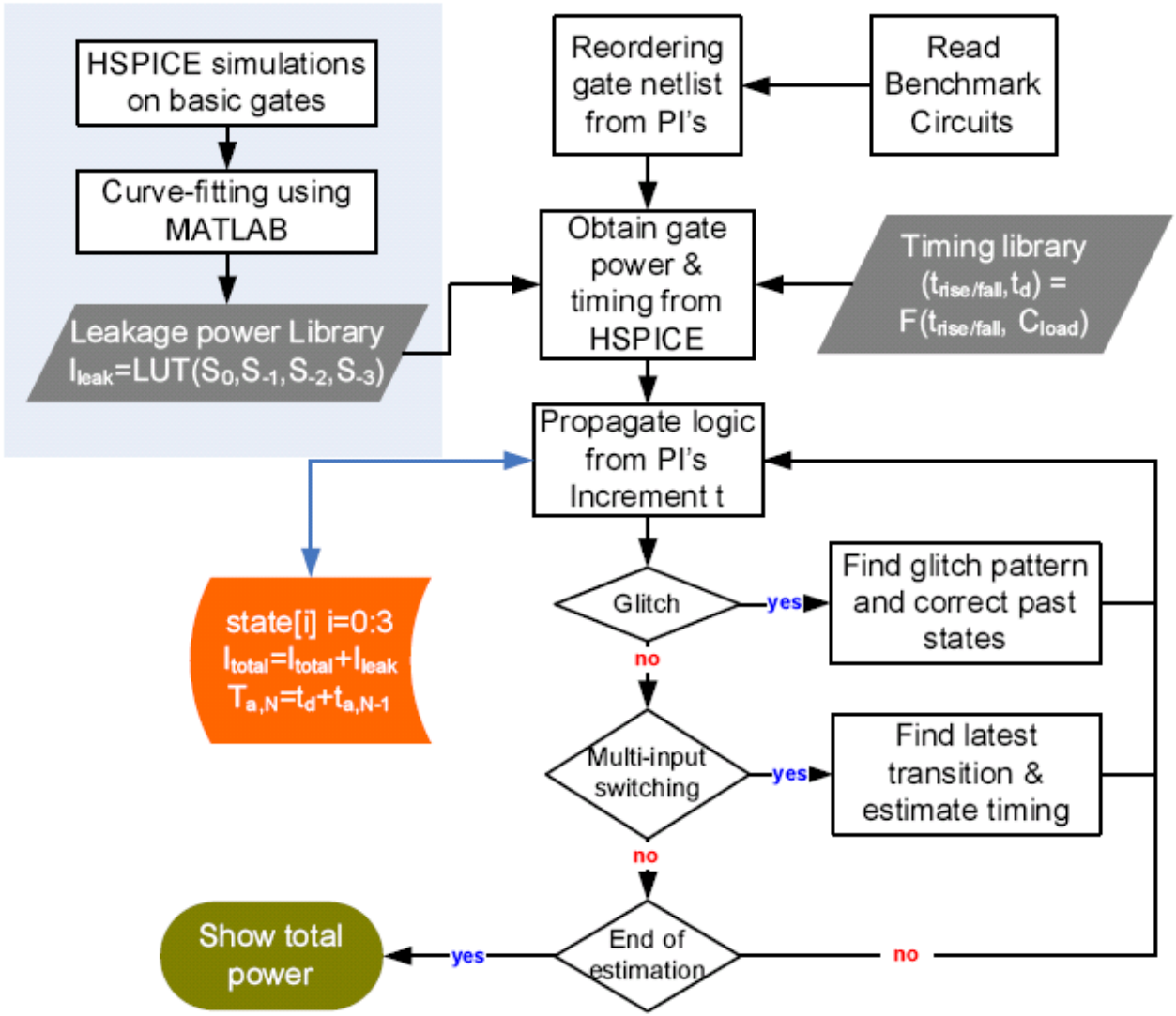
Dynamic runtime leakage estimation

- Keep tracking internal voltage of a gate is computational expensive and small error can greatly affect the results
- Our approach considers the following steps
  - Past N states of the gate: (S_{-1}, S_{-2}, …, S_{-N})
  - Time spent in the current state: t_0
  - Change past states accordingly for simultaneous input switching
  - Glitches will only change the last state if the arrival time of inputs that cause the glitch is over a certain value
- Each gate will be characterized will past states and be stored in a table
- Transient value of the leakage is curved fitted to exponential function

Library characterization

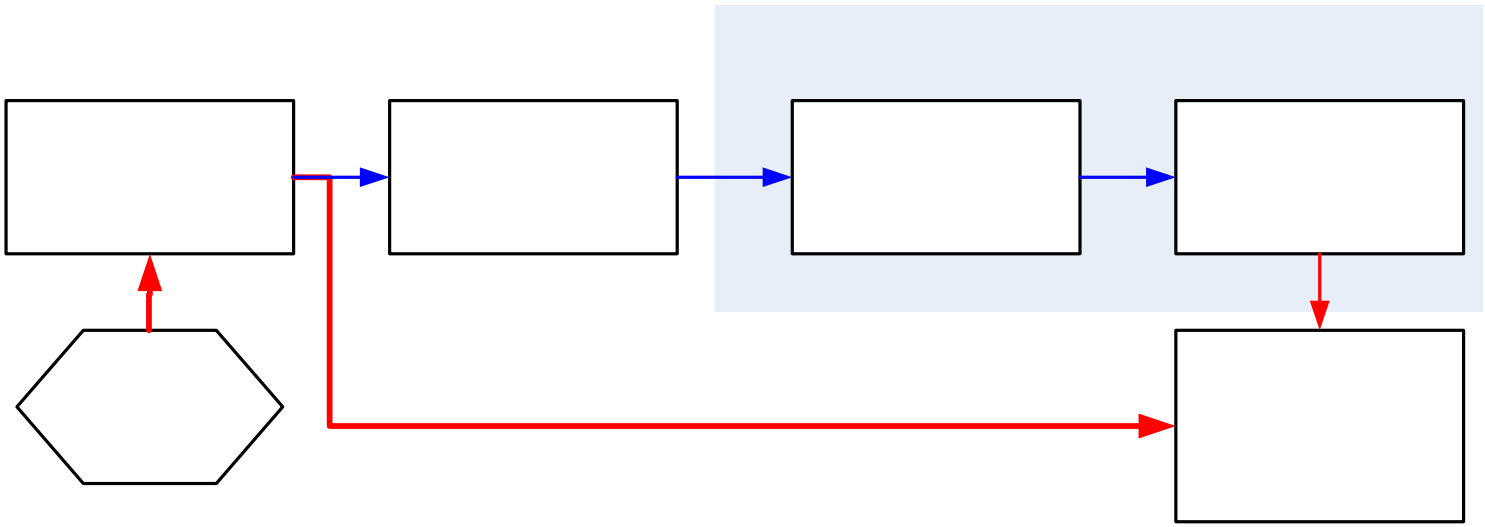
- In general, with N past states a M-input gate requires $2^{M(N+1)}$ simulation runs
- If the current input state shorts all its internal nodes to its rails, then the past states do not matter
  - For example, state (01, 11, 10, 00) gives the same leakage current for a NAND2 gate as (10, 00, 10, 00)

Dynamic leakage estimation flow



Static approach based on state probability analysis

- For simplicity, spatial correlation of gates are neglected so that the state probability can be independently calculated for each gate
- Dynamic estimation approach is used to generate a N-dimensional lookup table for leakage power, where N is the number of inputs of a gate



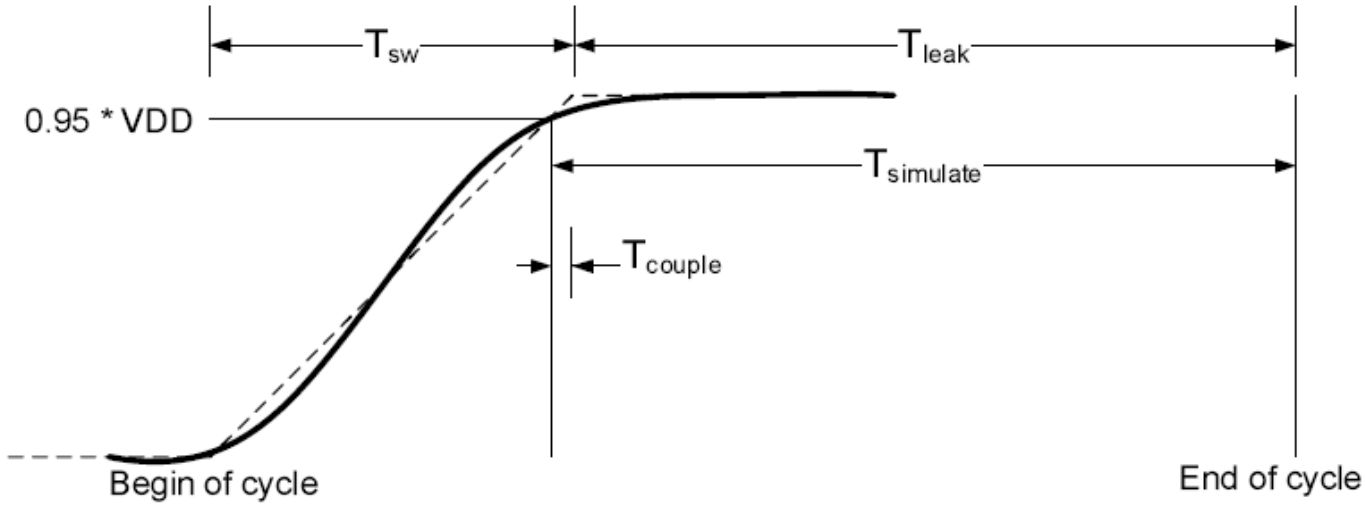
Simulation setup

SUMMARY OF SIMULATION SETUP

Technology	Industrial 90nm CMOS
Cell libraries	INV, NAND2, NAND3 NOR2, NOR3
Benchmark circuits	ICSAS '85
PI rise/fall time	50ps (0-100%)
Input pattern	Randomly generated with activity rate of 12.5%
Number of patterns	500 or 250

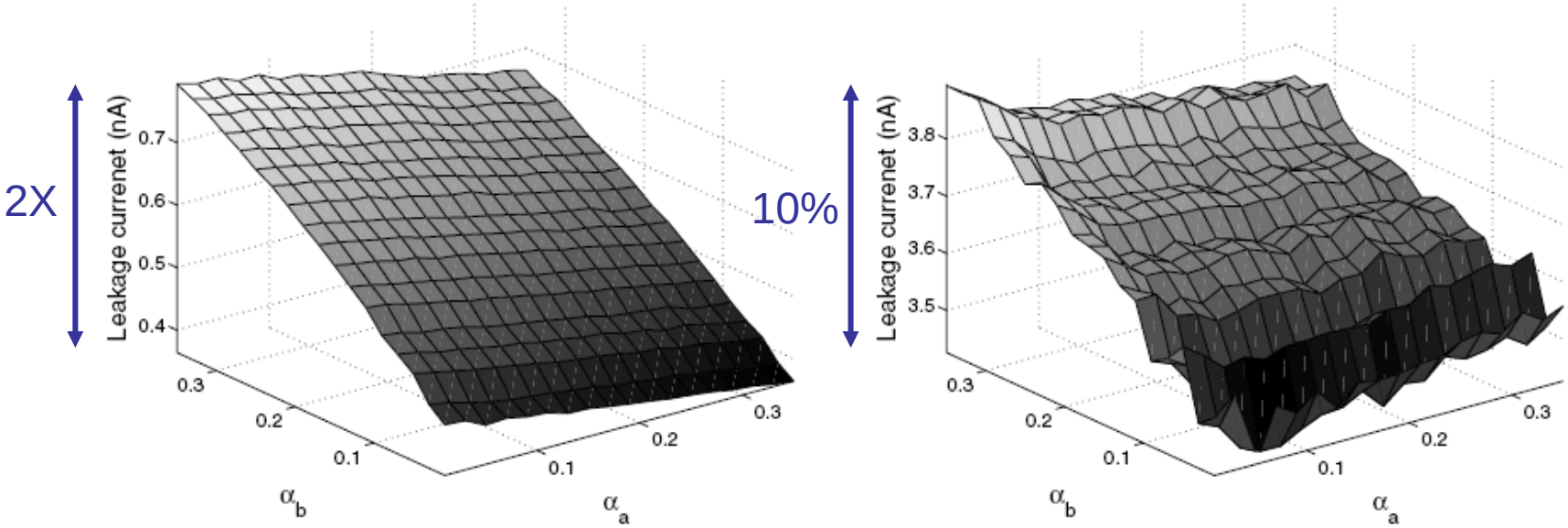
Determination of leakage measurement period for SPICE simulation

- To use SPICE as golden reference, leakage current is measured after the input finishes 95% of the switching
- The leakage between the beginning of the cycle and the first transition is negligible due to the short period and the low value compared to the leakage after transition



State probability dependency on leakage current

- For NAND2 gate, the upper transistor A has much lower impact on the leakage compared to lower transistor B
- Runtime leakage varies more from its steady state value for medium V_t transistors



medium Vt transistors

low Vt transistors

Simulation results: accuracy comparison

- Both dynamic and static approaches show about the same accuracy compared to HSPICE

	Clock period(ps)	HSPICE I_{avg}(nA)	Dynamic method I_{avg}(nA)	Error compared to HSPICE(%)	Static method I_{avg}(nA)	Error compared to HSPICE(%)	Steady state I_{avg}(nA)	Error compared to HSPICE(X)
c17	1000	8.52	7.57	-11.16	8.12	-4.66	3.51	2.43
c432	1000	236.97	210.69	-11.09	255.83	7.96	80.41	2.95
c499	1000	607.14	712.14	17.29	399.90	-34.13	230.48	2.63
c880	1000	604.54	583.74	-3.44	675.75	11.78	184.17	3.28
c1355	1000	527.10	641.82	21.76	595.92	13.06	258.16	2.04
c1908	1500	461.39	424.98	-7.89	440.46	-4.54	202.72	2.28
c2670	1500	930.5	721.2	-22.49	815.74	-12.33	426.68	2.18
c3540	1500	970.87	1075.62	10.79	1179.59	21.50	450.72	2.15
c5315*	1500	1407.61	1418.52	0.78	1466.14	4.16	809.65	1.53
c6288*	2500	2058.25	1894.82	-7.94	1938.39	-5.82	1177.33	1.75
c7552*	1500	1520.35	1650.13	8.54	1796.31	18.15	992.75	1.74
$\|Avg.\|$				11.2		12.6		

Simulation results: simulation speed

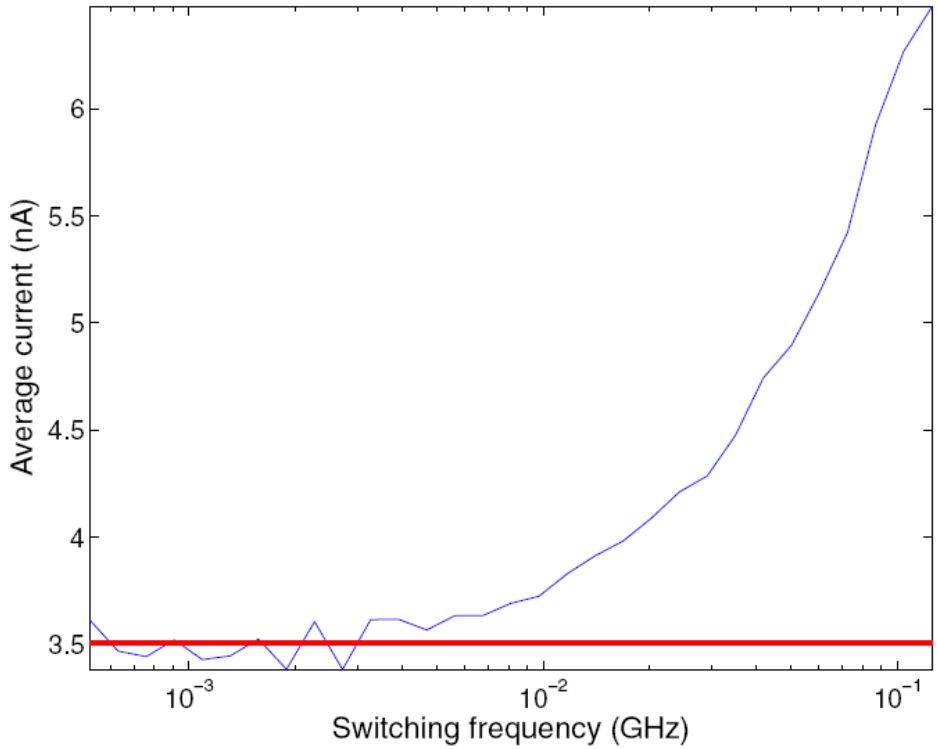
- Our estimation has much improved compared to HSPICE in terms of simulation speed

	HSPICE	dynamic method	static method
c17	691.23	0.88	0.17
c432	18793.67	16.6	0.16
c499	87338.57	49.28	0.18
c880	68730.12	40.54	0.17
c1355	109726.36	57.68	0.19
c1908	87812.18	45.45	0.24
c2670	434945.44	95.20	0.22
c3540	442573.41	103.49	0.24
c5315	553214.50	92.54	0.29
c6288	1452735.07	149.81	0.38
c7552	671991.82	211.85	0.33

Convergence to steady-state

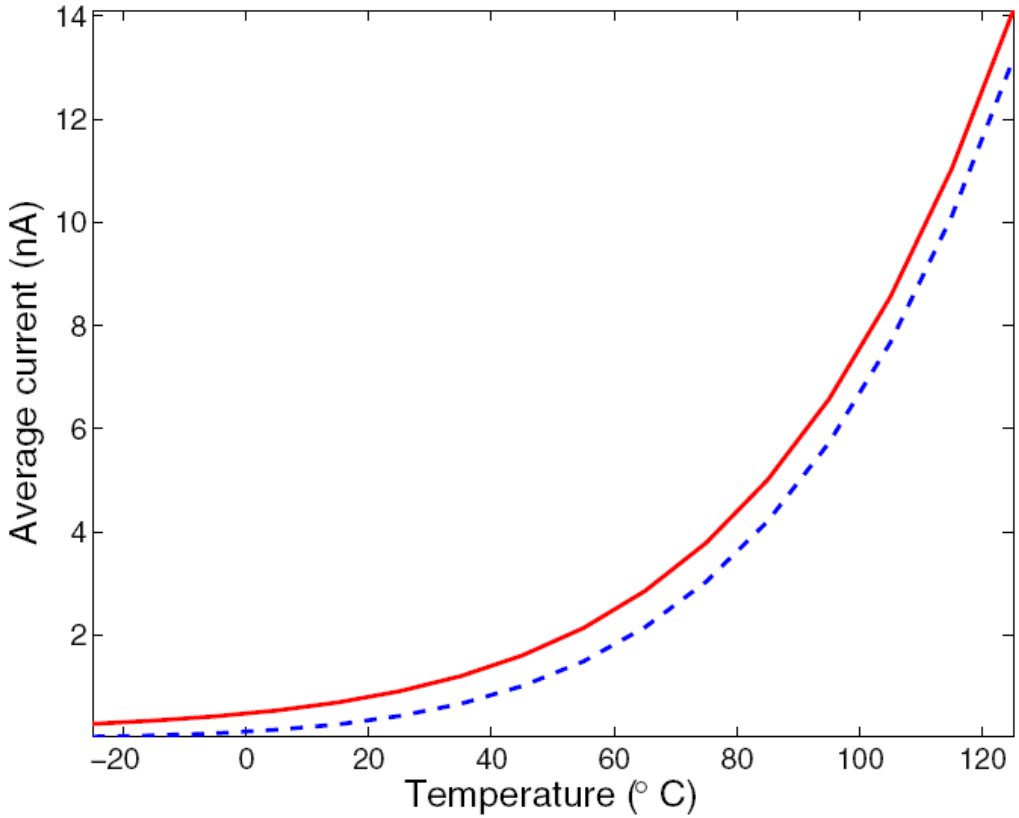
- By increasing the period between each input event, the dynamic behavior will finally converged its steady state

example: ISCAS c17



Temperature effect on leakage estimation

- Both runtime leakage (red solid line) and steady state leakage follows closely to each other with temperature variation



Conclusions

- We reinvestigate the impact of transient behavior on subthreshold leakage
- Both dynamic and static estimation methods are provided for roughly 12% error and improved simulation speed