Power Estimation

Sangyoung Park
Chair for Real-Time Computer Systems
Technical University of Munich
sangyoung.park@tum.de

Contents

- Why power estimation?
- SPICE power analysis
- Power estimation basics
- Signal probability calculation
- Switching Activity
- Leakage Estimation

Why Power Estimation?

- Two types of control
  - Open loop control
    - Set a control law and expect faithful following
  - Closed loop control
    - Set a control law
    - Read the output
    - Calculate the error
    - Reset the control law
- Purpose of estimation
  - When direct read of the output and/or internal state is not possible
  - Early-stage decision
    - To establish a strategy
    - To figure out potential gain
- Power measurement and estimation are crucial for high-level power saving practices

How can we estimate power consumption?

- Direct measurement
  - Using lab equipments such as DAQs, oscilloscopes
    - Only for offline characterization
    - Not suitable for final product
  - Using onboard shunt resistors and ADCs
    - Applicable for online usage
    - Uses precious space and resources in the final products
- Indirect estimation
  - Simulation
    - Slow estimation speed
    - Only for offline characterization
  - Estimation using performance parameters
    - Fast enough for online utilization
    - Not all necessary parameters maybe accessible

Estimation using Circuit Simulation

- SPICE is the de facto standard power analysis tool
  - Simulation program with integrated circuit emphasis
  - A lot of SPICE related literatures and simulators
    - HSPICE, PSPICE, and so on
  - The reference for the higher abstraction levels
  - Accurate but slow
    - Analytical models of MOSFET
- Recently, faster analysis tools were introduced
  - E.g. PowerMill, Spectre, and so on
  - Still accuracy is inferior to SPICE

SPICE basics

- Solving a large matrix of nodal current using Kirchhoff's Current Law (KCL)
- Primitive elements
  - Resistors, capacitors, inductors, current sources, and voltage sources
- More complex elements
  - Such as diodes and transistors
  - Constructed from the primitive elements
- Analysis modes
  - DC analysis
  - Transient analysis

SPICE power analysis

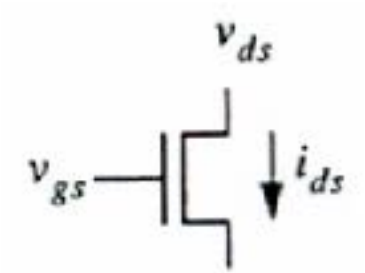
- Can estimate all types of power consumption
  - Dynamic/static/leakage
- Not feasible for the entire chip due to the computation complexity
  - Can be used as a characterization tool for higher abstraction level analysis
- Can consider process and other parameter's variation
  - BEST/TYPICAL/WORST

Discrete transistor modeling/analysis

- To speed up the analysis
  - Lose accuracy
- Typical methods
  - Circuit model
    - Approximate the complex equations into a linear equation
  - Tabular transistor model
    - Express the transistor models in tabular forms
  - Switch model
    - Consider a transistors as a two-state switch (on/ off)

Circuit model

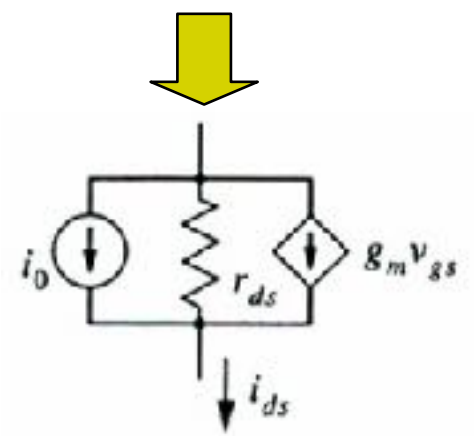
- Linear circuit model



$$I_{DS} = \frac{k}{2}\left(2(V_{GS} - V_T)V_{DS} - V_{DS}^2\right)$$

$$I_{DS} = \frac{k'}{2}(V_{GS} - V_T)^2(1 + \lambda V_{DS})$$

$$I_{DS} = W v_{sat} C_{ox} \frac{(V_{GS} - V_T)^2}{V_{GS} - V_T + E_c L}$$



$$i_{ds} \approx i_0 + g_m v_{gs} + r_{ds}$$

- The linear equation should be numerically evaluated whenever the operating points change

Tabular transistor model

- Pre-compute a current table
- Look up the table instead of solving an equation
- Table format

V_{gso}	V_{dso}	i_{ds}
0.1	0.1	1
...	...	...
5	5	10

- One-time characterization effort for each MOS
- Event-driven approach can be used for speed-up
- Nearly two orders of magnitude improvement (speed, size)

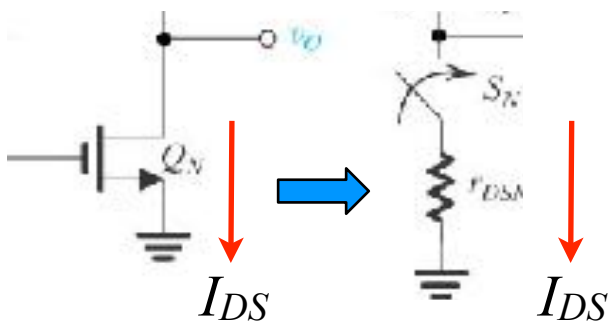
Switch model

- RC calculation for timing
- Power is estimated from the switching frequency and capacitance

$$I_{DS} = \frac{k}{2}\left(2(V_{GS} - V_T)V_{DS} - V_{DS}^2\right)$$

$$I_{DS} = \frac{k'}{2}(V_{GS} - V_T)^2(1 + \lambda V_{DS})$$

$$I_{DS} = W v_{sat} C_{ox} \frac{(V_{GS} - V_T)^2}{V_{GS} - V_T + E_c L}$$



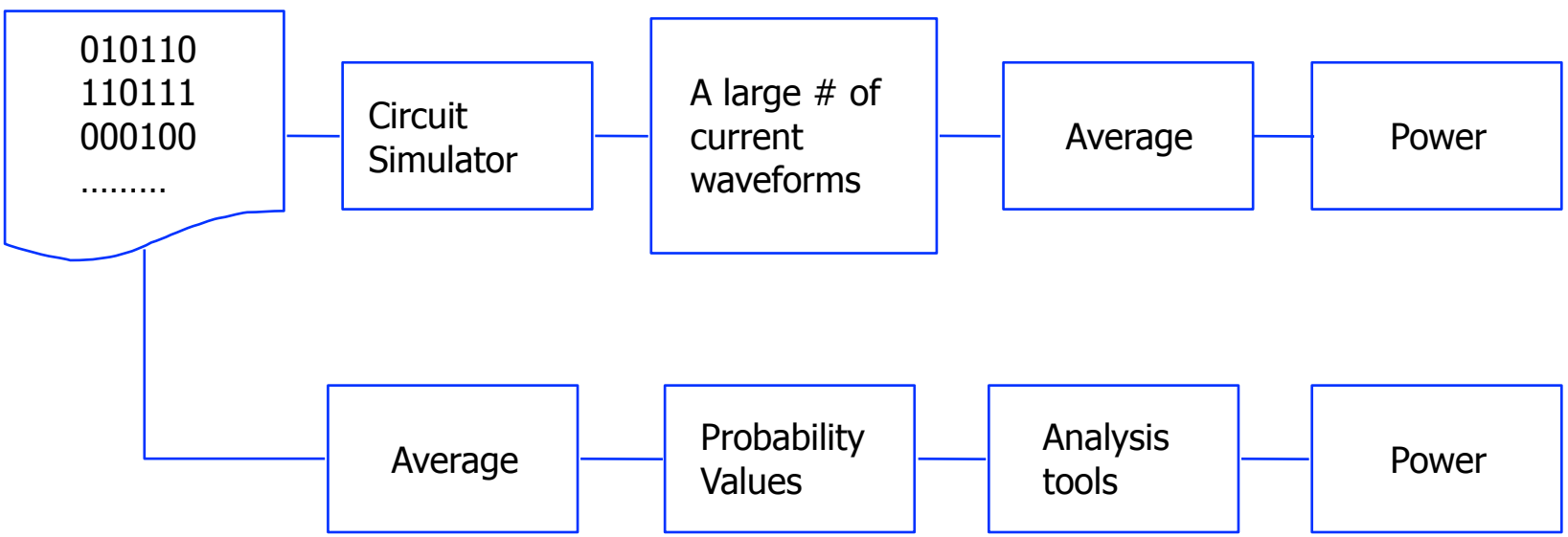
- Further speed-up, but less accuracy

Power characterization for cell library

- Circuit-level power analysis is time consuming
- Need to speed up with reasonable accuracy loss
- Estimation beyond the gate level will be discussed later
- Partially similar to delay characterization
- Dynamic power
  - Capacitive power dissipation
  - Internal switching power dissipation
- Leakage power
  - Accuracy depends on the model of circuit simulation
  - Iterative analytic estimation
  - Simulation based approach

Power characterization flow

- Accuracy vs. speed
  - Too many input patterns → too many simulation runs
  - Too many input patterns → probabilistic analysis



Simulation-based cell characterization

- Parameters
  - Input pattern (logical value)
  - Input slope
  - Output loading capacitance
  - Process condition
- Total # of runs of simulation is the multiplication of the possible number of values of each parameter
  - Some parameters are continuous
    - Input slope, output loading capacitance
    - Piece-wise linear approximation is widely used
  - Process/operation condition
    - BEST/TYPICAL/WORST

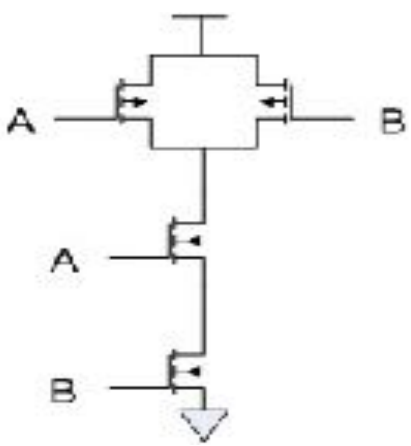
Example: 2-input NAND (I)

- Possible input patterns
  - Dynamic power
  - Static power

A	B	C	Power
1	r	f	?
1	f	r	?
r	1	f	?
f	1	r	?

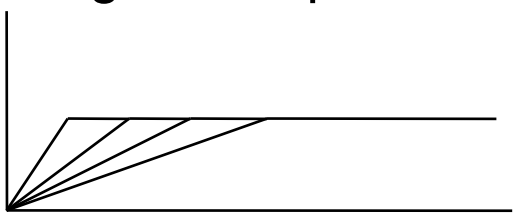
A	B	C	Power
0	0	1	?
0	1	1	?
1	0	1	?
1	1	0	?

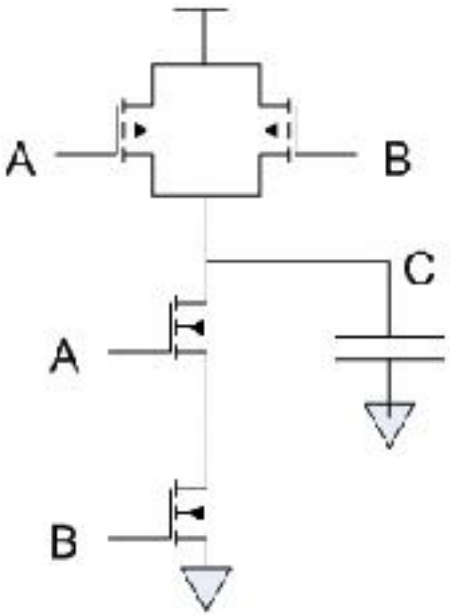
8 simulation runs!



Example: 2-input NAND (II)

- Input slope
  - Depending on the predecessor





- Capacitance
  - Depending on the successor
  - proportional to the # of fan-outs
  - If we consider four points for capacitance
- Total # of simulation runs for a single input
  - 2 (rise / fall) x 4 (# of input slopes) x 4 (# of capacitance points) = 32 points

Example: 2-input NAND (III)

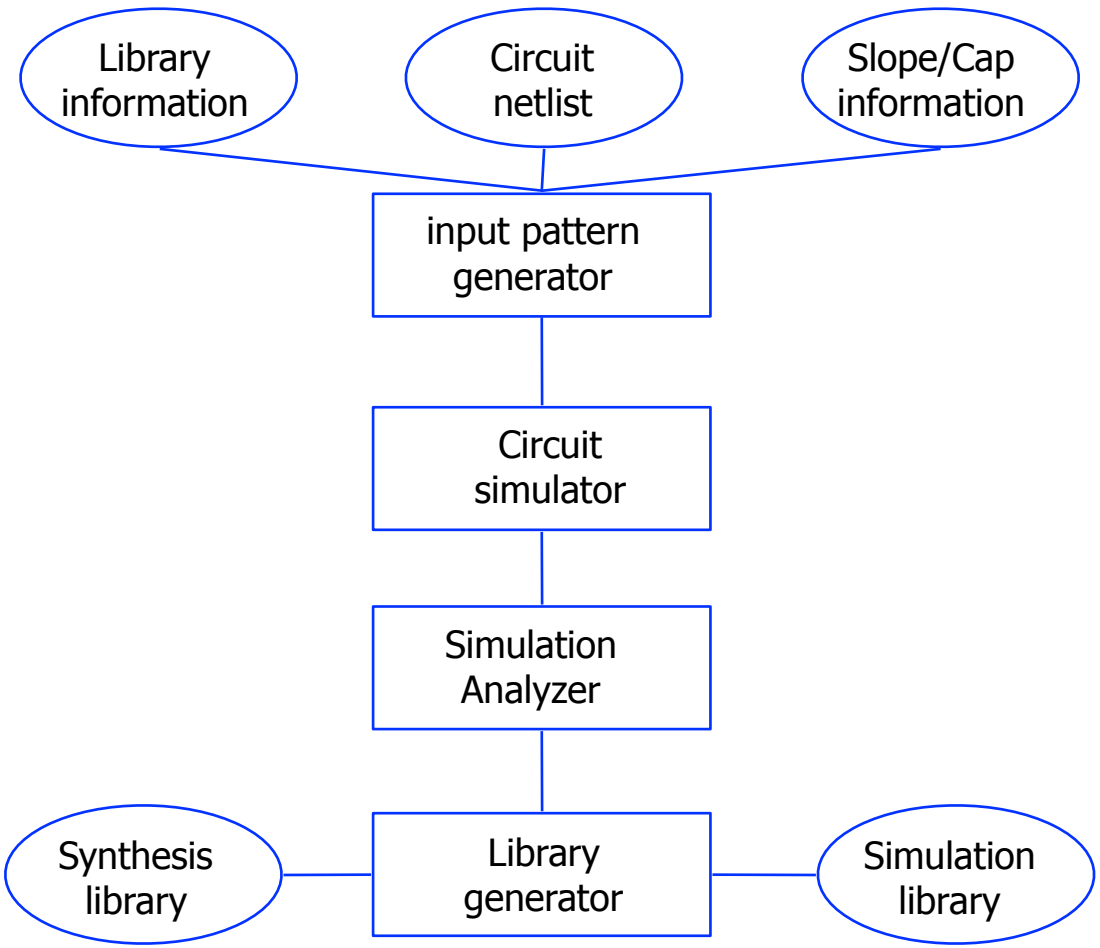
- Process/operation condition
  - Temperature
  - Process variations such as doping density
  - Typically use 3 conditions are widely used
    - BEST/TYPICAL/WORST

- Total # of simulations
  - For dynamic power
    - (2 x 2) x S x C x P
  - For static power
    - 22 x P

Additional factors to be characterized

- Output slope
  - Used as an input slope of the successor
  - Need to know for each simulation point
- Input capacitance
  - Used for computing the total output capacitance of the predecessor
  - Can be estimated by the area of gate (W/L) and T_{ox}
  - Parasitics: C_{gs}/C_{gd}
- All the information should be included in the library

Tool flow


Library information
Circuit netlist
Slope/Cap information
input pattern generator
Circuit simulator
Simulation Analyzer
Synthesis library
Library generator
Simulation library

Probability-based power estimation

- If we ignore internal capacitance of a logic gate
  - $$P_{avg} = \frac{1}{2} V_{dd}^2 C f$$
- Parameters
  - C: switched capacitance
  - f : the frequency of operation
    - For aperiodic signals: the average # of signal transitions per unit time
    - Called signal activity
- Our concern
  - How to estimate f in a probabilistic manner

Modeling of signals

- To model the digital signals, need to know
  - Signal probability
  - Signal activity
- $g(t)$, $t \in (-\infty, \infty)$
  - A stochastic process that takes the values of logical 0 or 1
  - Transitioning from one to the other at random times
  - SSS: Strict-Sense Stationary
  - Mean ergodic
    - Constant mean with a finite variance
    - $g(t)$ and $g(t+\tau)$ become uncorrelated as $\tau \rightarrow \infty$

Signal probability and activity

- Signal probability

$$P(g) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{+T} g(t)dt$$

  - P(g=1) : signal probability

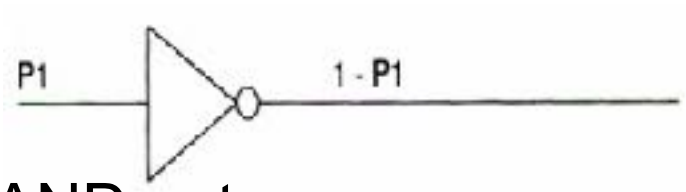
- Signal activity

$$A(g) = \lim_{T \to \infty} \frac{n_g(T)}{T}$$

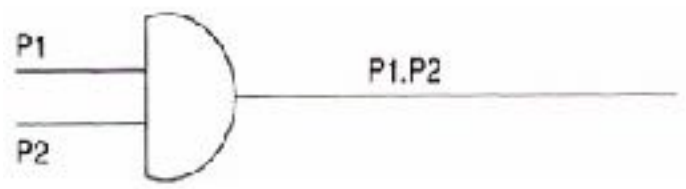
  - $n_g(t)$: # of transitions of g(t) in the time interval between –T/2 and +T/2

Signal probabilities of simple gates

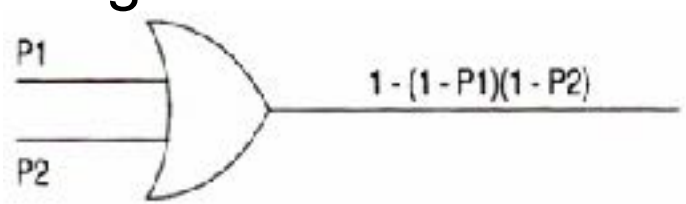
- Inverter



- AND gate



- OR gate



- Assumption
  - $g_1, g_2, \ldots, g_n$ are independent
- Output signal probability
  - Determined by the given boolean function
  - NOT: 1 –
  - AND: multiply
  - OR → NOT ((NOT) AND (NOT))

Signal probability calculation (I)

- By Parker and McClusky
- Algorithm: Compute signal probabilities
  - Input: Signal probabilities of all the inputs to the circuit
  - Output: Signal probabilities of all nodes of the circuit
  - Step1: For each input signal and gate output in the circuit, assign a unique variable
  - Step2: Starting at the inputs and proceeding to the outputs, write the expression for the output of each gate as a function (using standard expressions for each gate type for probability of its output signal in terms of its mutually independent primary input signals)
  - Step3: Suppress all exponents in a given expression to obtain the correct probability for that signal

Signal probability calculation (II)

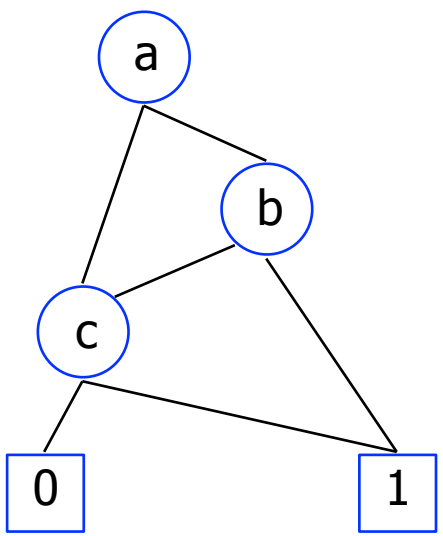
- Step 3 for protecting reconvergent fanout
  - W/o step 3, the reconvergent fanout node may have a signal probability higher than 1
- A boolean function f
  - $$P(f) = \sum_{i=1}^{p} \alpha_i \left(\prod_{k=1}^{n} P^{m_i,k}(x_k) \right)$$
  - n: # of independent inputs
  - p: # of products
  - α_i: some integer
  - Called as the sum of probability products of f

Signal probability calculation: Example

- $y = x_1x_2 + x_1x_3$, x_i, I = 1, 2, 3 are mutually independent
- $z = x_1x_2' + y$
- $P(y) = P(x_1x_2) + P(x_1x_3) - P(x_1x_2)P(x_1x_3)$
  $= P(x_1)P(x_2) + P(x_1)P(x_3) - P(x_1)P(x_2)P(x_3)$
- $P(z) = P(x_1x_2') + P(y) - P(x_1x_2')P(y)$
  $= P(x_1)P'(x_2) + P(x_1)P(x_2) + P(x_1)P(x_3) - P(x_1)P(x_2)p(x_3)$
  $- P(x_1)P'(x_2)(P(x_1)P(x_2) + P(x_1)P(x_3) - P(x_1)P(x_2)P(x_3))$
- $P(x_2)P'(x_2) = P(x2)(1 - P(x_2)) = 0$
- $P(z) = P(x_1)P'(x_2) + P(x_1)P(x_2) + P(x_1)P(x_3) - P(x_1)P(x_2)p(x_3)$
  $- P(x_1)P'(x_2)P(x_3)$

Signal probability using BDD (I)

- BDD: Binary Decision Diagram
- Shannon's expansion
  - $f = x_i \bullet f(x_1,...,1,x_{i+1},...,x_n) + \overline{x_i} f(x_1,...,0,x_{i+1},...,x_n)$
- Cofactors w.r.t. x_i and x'_i
  - $f_{x_i} = f(x_1,...,1,x_{i+1},...,x_n)$
  - $f_{\overline{x_i}} = f(x_1,...,0,x_{i+1},...,x_n)$
- Example
  - f = ab + c



Signal probability using BDD (II)

- P(f)
  - $P(x_1 \bullet f_{x_1} + \overline{x_1} \bullet f_{\overline{x_1}})$
  - $P(x_1 \bullet f_{x_1}) + P(\overline{x_1} \bullet f_{\overline{x_1}})$
  - $P(x_1) \bullet P(f_{x_1}) + P(\overline{x_1}) \bullet P(f_{\overline{x_1}})$
- A depth first traversal of BDD, with a post order evaluation of P(.) at every node is required for evaluation of P(f)

References

- http://public.itrs.net
- Gary K. Yeap, “Practical Low Power Digital VLSI Design”, Kluwer Academic Publishers, 1997
- Kaushik Roy and Sharat C. Prasad, “Low Power CMOS VLSI: Circuit Design”, Wiley Interscience, 2000
- Kiat-Seng Yeo, Kaushik Roy, “Low Voltage, Low Power VLSI Subsystems”, McGraw-Hill, 2004

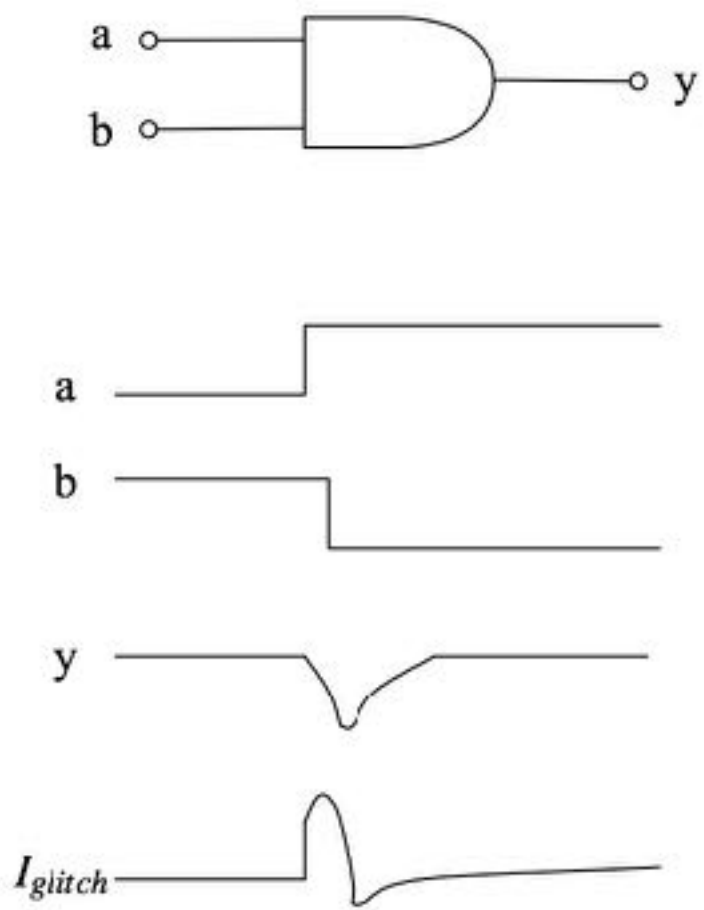
Switching Activity

- Activity Factor:
  - System clock frequency = f
  - Let f_{sw} = αf, where α = activity factor
    - If the signal is a clock, α = 1
    - If the signal switches once per cycle, α = ½
    - Dynamic gates: switch either 0 or 2 times per cycle, α = ½
    - Static gates: depending on design, but typically α = 0.1

- Switching power: $P_{sw} = \alpha f V_{DD}^2 C_L$

Switching Activity

- Abnormal switching activity
  - Glitch power
    - Power dissipated in intermediate transitions during the evaluation of the logic function
    - Unbalanced delay paths are principle cause
    - Usually 8% -25% of dynamic power



Switching Activity

- Transition Probability
  - Dynamic power is data dependent
  - Activity factor is dependent on the run-time data
  - Switching activity, $P_{0\rightarrow1}$, has two components
    - A static component: function of the logic topology
    - A dynamic component: function of the timing behavior (glitch)
- Static transition probability
  - $P_{0\rightarrow1} = P_{out=0}\, P_{out=1} = P_0(1-P_0)$
- With input signal probabilities $P_{A=1} = 1/2$ and $P_{B=1} = 1/2$
  - NOR static transition probability = 3/4 x 1/4 = 3/16

2-input NOR Gate

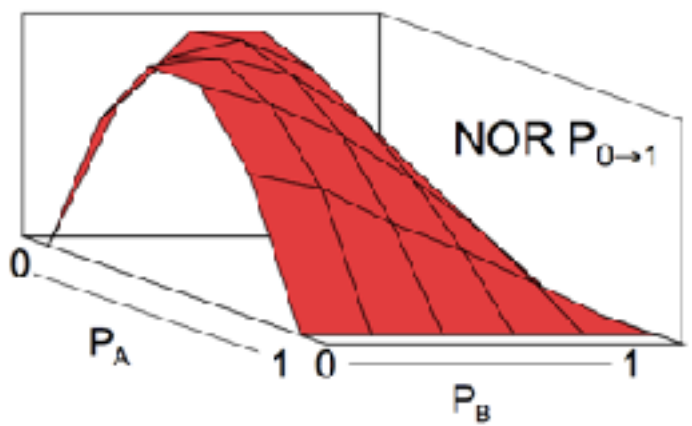
A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Switching Activity

- Transition Probability
  - Switching activity is a strong function of the input signal statistics
  - Generalized switching activity of a 2 input NOR gate
    - $P_{0\rightarrow1} = P_0P_1 = (1-(1-P_A)(1-P_B))\,(1-P_A)(1-P_B)$

Transition probability for basic gates

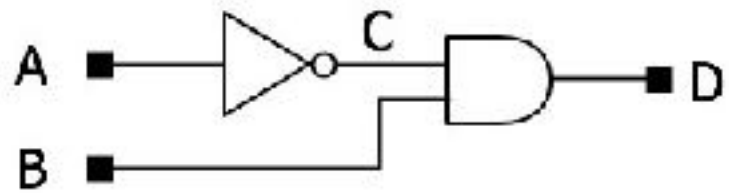
	$P_{0\rightarrow1} = P_{out=0} \times P_{out=1}$
NOR	$(1-(1-P_A)(1-P_B)) \times (1-P_A)(1-P_B)$
OR	$(1-P_A)(1-P_B) \times (1-(1-P_A)(1-P_B))$
NAND	$P_AP_B \times (1-P_AP_B)$
AND	$(1-P_AP_B) \times P_AP_B$
XOR	$(1-(P_A+P_B-2P_AP_B)) \times (P_A+P_B-2P_AP_B)$



Transition probability for 2 input NOR gates

Switching Activity

- Transition Probability
  - Transition probability propagation



- C: $P_{0\rightarrow1} = P_0P_1 = (1\text{-}P_A)\,P_A = 1/2 \times 1/2 = 1/4$
- D: $P_{0\rightarrow1} = P_0P_1 = (1\text{-}P_CP_B)\,P_CP_B$
  $= (1 - (1/2 \times 1/2)) \times (1/2 \times 1/2) = 3/16$

Switching Activity

- Signal Probability (advanced)
  - Generalized switching activity in combinational logic
  - Boolean difference:

$$\frac{\partial f_j}{\partial x_i} = f_j|_{x_i=1} \oplus f_j|_{x_i=0}$$

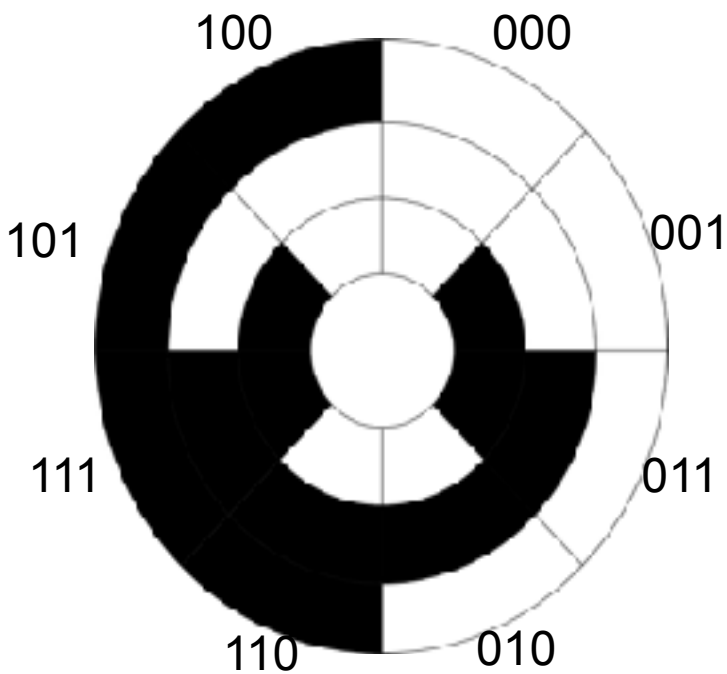
  - Switching activity in sequential logic
  - Estimation of glitch power

Switching Activity

- Decreasing the switching activities
  - No or little performance and/or functional degradation
  - Different coding techniques
    - Fewer bit transitions between two states
  - Boolean expressions simplification
    - Gate minimization
  - Avoid glitches
    - Get rid off unnecessary transitions
  - Power down modes
    - Turn off parts of that are not in use

Switching Activity

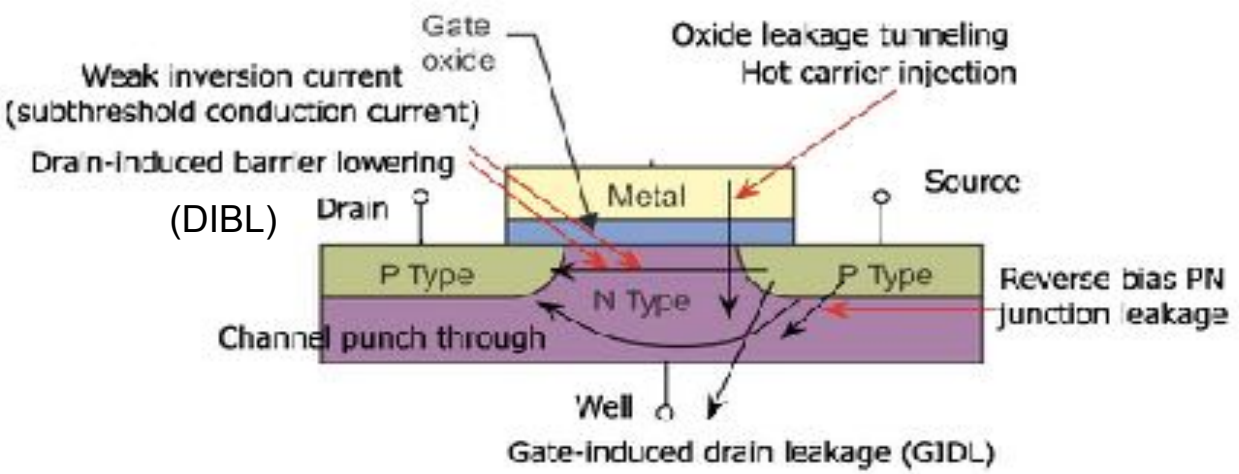
- Decreasing the switching activities
  - Example: gray coding
    - Hamming distance of one
    - Used when a sequence is predictable
      - FSMs
      - Address busses
    - Makes full use of the bit-width



Leakage Estimation

- Transistor leakage estimation
  - Leakage power components
    - Subthreshold leakage is the focus in leakage current modeling

$$I_{subthreshold} = \frac{\mu W C_{ox}}{L} V_T^2 e^{\frac{|V_{GS}| - |V_t|}{nV_T}} \left(1 - e^{\frac{-|V_{DS}|}{V_T}}\right)$$



Leakage Estimation

- Transistors in a circuit
  - Leakage current is strongly dependent on the relative position of on and off devices in a transistor network
- Position of devices
  - If transistors are connected in parallel and turned off, VDS and VS are similar for each other
    - Leakage current can be calculated independently and summed up
  - If transistors are connected in series and turned off
    - Subthreshold current though each transistor must be the same

$$V_{DS_2} = \frac{nkT}{q(1+2\eta+\gamma)} ln\left(\frac{A_1}{A_2} e^{\frac{\eta V_{DD}}{nkT}} + 1\right)$$

    - Voltage of the I-th transistor

$$V_{DS_i} = \frac{nkT}{q(1+\gamma)} ln\left(1 + \frac{A_{i-1}}{A_i}\left(1 - e^{\frac{-qV_{DS_{i-1}}}{kT}}\right)\right)$$

Leakage Estimation

- Large-circuit leakage current computation
  - Stack-based leakage estimation
    - On transistors are considered as a short circuit
    - Ignorance of the on resistance of transistors
  - The leakage current of a transistor in parallel with an on transistor is ignored
    - V_{DS} is estimated for the remaining transistors using

$$V_{DS_2} = \frac{nkT}{q(1+2\eta+\gamma)} ln\left(\frac{A_1}{A_2} e^{\frac{\eta V_{DD}}{nkT}} + 1\right)$$

- Leakage power

$$V_{DS_i} = \frac{nkT}{q(1+\gamma)} ln\left(1 + \frac{A_{i-1}}{A_i}\left(1 - e^{\frac{q}{kT} V_{DS_{i-1}}}\right)\right)$$

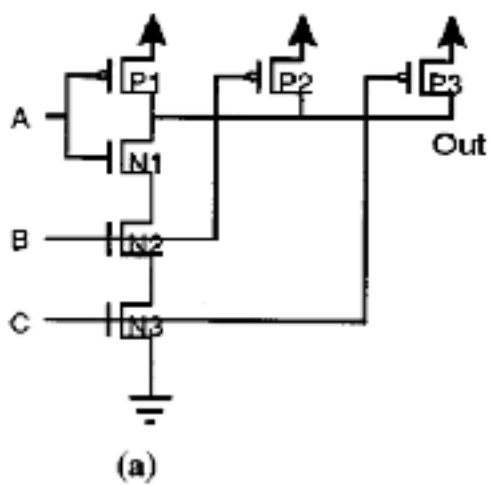
$$P_{leakage} = \sum_i I_{DS_{q_i}} V_{DS_{q_i}}$$

Leakage Estimation

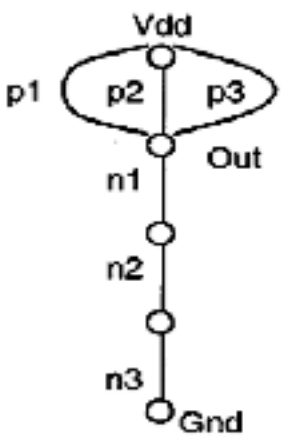
- Very large-circuit leakage estimation
- Probabilistic approach
  - Hurdles of large-circuit leakage current calculation
    - Calculation of the leakage current is complicated due to highly nonlinear behavior of the drain current with respect to source/drain voltage
    - SPICE simulation by using nonlinear model is still very expensive
      - Not feasible for the repeated evaluation of large circuits
    - Leakage current of a circuit is highly dependent on the circuit state
      - State probability must be considered

Leakage Estimation

- State probability
  - Three-input NAND SPICE leakage simulation



(a)



(b)

State (ABC)	Leakage Current (nA)	Leaking Transistors
000	0.095	N1, N2, N3
001	0.195	N1, N2
010	0.195	N1, N3
011	1.874	N1
100	0.185	N2, N3
101	1.220	N2
110	1.140	N3
111	9.410	P1, P2, P3

Leakage Estimation

- Gate state estimation
  - Necessary to simulate a substantial portion of the gates' states to obtain accurate average leakage of each gate
  - Requires extremely large number of random global circuit vectors
  - Complexity reduction method
    - Probabilistic approach eliminates the need to do simulation over all 2n
    - A small subset of all the possible states is evaluated, based on the notion of dominant-leakage states

Leakage Estimation

- Calculation of state probability
  - Statistical simulation to measure the average leakage of an entire circuit
    - Monte Carlo experiments
      - In each iteration, a randomly chosen circuit state is applied
  - Probabilistic approach is more effective than statistical simulation for optimization purpose
    - Leakage optimization relies on accurate estimation rather than the estimation of the total leakage

Leakage Estimation

- Further simplification of the leakage calculation
  - Dominant leakage states
    - Leakage current in some states is significantly smaller than other states
      - A state with more than one off transistor in a path from VDD to GND results in far less leakage than a state with one off transistor (dominant leakage state)
      - A set of dominant leakage states is generally small
    - Example: three-input NAND gate SPICE simulation
      - Average leakage is 1.78925 nA
      - Set of dominant leakage D={011, 101, 110, 111}
      - Only consideration of D, the average leakage is 1.7055 nA with 4.68% error

Note

- Slides are modified from lecture notes of “Advanced Computer System Design” from Seoul National University (Lecturer: Prof. Naehyuck Chang)