Power Estimation

Naehyuck Chang
Dept. of EECS/CSE
Seoul National University
naehyuck@snu.ac.kr
Contents

- SPICE power analysis
- Power estimation basics
- Signal probability calculation
- Switching Activity
- Leakage Estimation
Circuit-level power analysis

- 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 Krichoff’s Current Law (KCL)
- Primitive elements
  - Registers, 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

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

<table>
<thead>
<tr>
<th>$V_{gso}$</th>
<th>$V_{dso}$</th>
<th>$i_{ds}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

- 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

\[
I_{DS} = \frac{k}{2} (2(V_{GS} - V_T)V_{DS} - V_{DS}^2)
\]

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

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

- 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
- Levels beyond 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 $\rightarrow$ too many simulation runs
  - Too many input patterns $\rightarrow$ probabilistic analysis

```
010110
110111
000100
```

- Circuit Simulator
- A large # of current waveforms
- Average
- Power

- Average
- Probability Values
- Analysis tools
- Power
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

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>r</td>
<td>f</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>f</td>
<td>r</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>r</td>
<td>1</td>
<td>f</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>1</td>
<td>r</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>

- Static power

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>

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**
  - \(2 \text{ (rise / fall)} \times 4 \text{ (# of input slopes)} \times 4 \text{ (# of capacitance points)} = 32 \text{ points}\)
Example: 2-input NAND (III)

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

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

- Pre-requisite to move to module 8
- If we ignore internal capacitance of a logic gate
  \[ P_{\text{avg}} = \frac{1}{2} V_{dd}^2 C_f \]
- 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 $\tau \to \infty$
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**
  - ![Inverter Diagram]
  - Assumption
    - $g_1, g_2, ..., g_n$ are independent
  - Output signal probability
    - Determined by the given boolean function
    - NOT: $1 - P$
    - AND: multiply
    - OR $\rightarrow$ NOT ((NOT) AND (NOT))

- **AND gate**
  - ![AND Gate Diagram]

- **OR gate**
  - ![OR Gate Diagram]
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 $f$
  
  $$P(f) = \sum_{i=1}^{n} \alpha_i \left( \prod_{k=1}^{n} P^{m_{i,k}}(x_k) \right)$$

  - $n$: # of independent inputs
  - $p$: # of products
  - $\alpha_i$: 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
- $z = x_1x_2' + y$
- $P(y) = P(x_1x_2) + P(x_1x_3) - P(x_1x_2)P(x_1x_3)$
  \[= P(x_1)P(x_2) + P(x_1)P(x_3) - P(x_1)P(x_2)P(x_3)\]
- $P(z) = P(x_1x_2') + P(y) - P(x_1x_2')P(y)$
  \[= P(x_1)P'(x_2) + P(x_1)P(x_2) + P(x_1)P(x_3) - P(x_1)P(x_2)p(x_3)
  - P(x_1)P'(x_2)(P(x_1)P(x_2) + P(x_1)P(x_3) - P(x_1)P(x_2)P(x_3))\]
- $P(x_2)P'(x_2) = P(x_2)(1 - P(x_2)) = 0$
- $P(z) = P(x_1)P'(x_2) + P(x_1)P(x_2) + P(x_1)P(x_3) - P(x_1)P(x_2)p(x_3)
  - P(x_1)P'(x_2)P(x_3)$
Signal probability using BDD (I)

- BDD: Binary Decision Diagram
- Shannon’s expansion
  - \[ f = x_i \cdot f(x_1, \ldots, 1, x_{i+1}, \ldots, x_n) + \bar{x}_i f(x_1, \ldots, 0, x_{i+1}, \ldots, x_n) \]
- Cofactors w.r.t. \( x_i \) and \( x'_i \)
  - \( f_{x_i} = f(x_1, \ldots, 1, x_{i+1}, \ldots, x_n) \)
  - \( f_{\bar{x}_i} = f(x_1, \ldots, 0, x_{i+1}, \ldots, x_n) \)
- Example
  - \( f = ab + c \)
Signal probability using BDD (II)

- $P(f)$
  - $P(x_1 \cdot f_{x_1} + \overline{x_1} \cdot f_{\overline{x_1}})$
  - $P(x_1 \cdot f_{x_1}) + P(x_1 \cdot f_{\overline{x_1}})$
  - $P(x_1) \cdot P(f_{x_1}) + P(x_1) \cdot P(f_{\overline{x_1}})$

- 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
Switching Activity

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

- **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, $P_{0 \rightarrow 1}$, has two components
    - A static component: function of the logic topology
    - A dynamic component: function of the timing behavior (glitch)

- Static transition probability
  - $P_{0 \rightarrow 1} = P_{\text{out}=0} P_{\text{out}=1} = P_0(1-P_0)$
  - With input signal probabilities $P_A=1/2$ and $P_B=1/2$
  - NOR static transition probability
    - $= 3/4 \times 1/4 = 3/16$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Gate</th>
<th>Transition Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOR</td>
<td>( (1 - (1 - P_A)(1 - P_B)) \times (1 - P_A)(1 - P_B) )</td>
</tr>
<tr>
<td>OR</td>
<td>( (1 - P_A)(1 - P_B) \times (1 - (1 - P_A)(1 - P_B)) )</td>
</tr>
<tr>
<td>NAND</td>
<td>( P_A P_B \times (1 - P_A P_B) )</td>
</tr>
<tr>
<td>AND</td>
<td>( (1 - P_A P_B) \times P_A P_B )</td>
</tr>
<tr>
<td>XOR</td>
<td>( (1 - (P_A + P_B - 2P_A P_B)) \times (P_A + P_B - 2P_A P_B) )</td>
</tr>
</tbody>
</table>

Transition probability for 2 input NOR gates
Switching Activity

• Transition Probability
  • Transition probability propagation

\[ C: \, P_{0\rightarrow 1} = P_0P_1 = (1-P_A) \, P_A = 1/2 \times 1/2 = 1/4 \]

\[ D: \, P_{0\rightarrow 1} = P_0P_1 = (1-P_C \, P_B) \, P_C \, P_B \]
\[ = (1 -(1/2 \times 1/2)) \times (1/2 \times 1/2) = 3/16 \]
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

\[
I_{\text{subthreshold}} = \frac{\mu W C_{ox}}{L} V_T^2 e^{\frac{|V_{GS}| - |V_t|}{nV_T}} \left(1 - e^{-\frac{|V_{DS}|}{V_T}}\right)
\]
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, $V_{DS}$ and $V_S$ 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
      
      $$V_{DS_2} = \frac{nkT}{q(1 + 2\eta + \gamma)} \ln \left( \frac{A_1}{A_2} e^{\frac{qV_{DD}}{nkT}} + 1 \right)$$

  - Voltage of the I-th transistor
    
    $$V_{DS_i} = \frac{nkT}{q(1 + \gamma)} \ln \left( 1 + \frac{A_{i-1}}{A_i} \left( 1 - e^{\frac{qV_{DS_{i-1}}}{nkT}} \right) \right)$$
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
      
      \[
      V_{DS_2} = \frac{nkT}{q(1 + 2\eta + \gamma)} \ln \left( \frac{A_1}{A_2} e^{\frac{\eta V_{DD}}{nkT}} + 1 \right)
      \]
      \[
      V_{DS_i} = \frac{nkT}{q(1 + \gamma)} \ln \left( 1 + \frac{A_{i-1}}{A_i} \left( 1 - e^{\frac{qV_{DS_{i-1}}}{kT}} \right) \right)
      \]

- Leakage power
  
  \[
  P_{leakage} = \sum_i I_{DS_{qi}} V_{DS_{qi}}
  \]
Leakage Estimation

• Very large-circuit leakage estimation
• Probabilistic approach
  • Huddles of large-circuit leakage current calculation
    • Calculation of the leakage current is complicated due to highly nonlinear behavior of the drain current wrt 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

<table>
<thead>
<tr>
<th>State (ABC)</th>
<th>Leakage Current (nA)</th>
<th>Leaking Transistors</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0.095</td>
<td>N1, N2, N3</td>
</tr>
<tr>
<td>001</td>
<td>0.195</td>
<td>N1, N2</td>
</tr>
<tr>
<td>010</td>
<td>0.195</td>
<td>N1, N3</td>
</tr>
<tr>
<td>011</td>
<td>1.874</td>
<td>N1</td>
</tr>
<tr>
<td>100</td>
<td>0.185</td>
<td>N2, N3</td>
</tr>
<tr>
<td>101</td>
<td>1.220</td>
<td>N2</td>
</tr>
<tr>
<td>110</td>
<td>1.140</td>
<td>N3</td>
</tr>
<tr>
<td>111</td>
<td>9.410</td>
<td>P1, P2, P3</td>
</tr>
</tbody>
</table>

(a) (b)
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 $2^n$
  - 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 \( V_{DD} \) 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