Outline

- Motivation and Objectives
- Power Estimation Methodology
- Example Analysis/Estimation Tools
- Power Optimization Flow
- Example Minimization Techniques
- Summary
Opportunities for Power Savings

- System: 50-90%
- Behavioral: 30-60%
- RT-Level: 25-40%
- Logic: 10-25%
- Physical: 5-10%

- HW/SW Co-design
- Custom ISA
- Algorithm Design
- Communication Synthesis

- Scheduling, Allocation
- Pipelining
- Behavioral Transformations

- Clock Gating, Precomputation
- Operand Isolation
- State Assignment, Retiming

- Logic Restructuring
- Technology Mapping
- Pin Ordering & Phase Assignment

- Fanout Optimization, Buffering
- Transistor Sizing, Placement
- Partitioning, Clock Tree Design
- Glitch Elimination

Realistic Estimation Expectations

- System
- Architectural
- RT-Level
- Gate
- Transistor

- Speed: 70-90%
- Minute
- Minutes: 40-70%
- Hour
- Hours: 30-50%
- Day
- Days: 10-30%
- Year
- Years: 5-10%

- Instruction-Level Models
- IP Core Models
- Program Complexity
- Program Simulation

- Entropic Bounds
- Architectural Simulation
- I/O and Memory Accesses

- RT-Level Macromodels
- HDL Simulation
- Quick Synthesis

- Probabilistic Simulation
- Gate-Level Simulation
- Sampling and Compaction
- ASIC Library Models

- Parasitic Extraction
- Accurate Timing Analysis
- Circuit-Level Simulation

© M. Pedram
Nov 1997
Example Applications

- Portable Electronics (PC, PDA, Wireless)
- Ultra-Low-Power Circuits (Pacemaker)
- Space Missions (Miniaturized Satellites)
- IC Cost (Packaging and Cooling)
- Reliability (Electromigration, Latch-up)
- Signal Integrity (Switching Noise, DC Voltage Drop)
- Thermal Design

Digital Camera Circuit

Source: LSI Logic

© M. Pedram
Nov 1997
A Power Estimation Methodology

Quick Synthesis

RTL Source Code

Quick RTL Synthesis

RTL Netlist

Quick Logic Synthesis

Mapped Netlist

© M. Pedram

University of Southern California

Nov 1997
Dynamic (Simulative) Techniques

Input Sequence

Probabilistic Compaction

Compacted Sequence

Event or Cycle-Based Simulation

Report Power

Statistical Sampling

Sampled Sequence

Library Data

Simple Random Sampling (SRS)

0% error
very time consuming

< 5% error
> 1000X speedup
Monte Carlo Simulation

- Input vectors
  - Generate one sample of k units
  - Do circuit level simulation
  - Calculate mean power & confidence interval

Efficiency: depends on population characteristics and sampling procedure

Converged?

- NO
- YES

“Difficult” distributions

Report Power

SRS Results

- Biased Sequence
- Random Sequence

C7552
C6288
C5315
C3540
C2670
C1908
C1355
C880
C432

0 2000 4000 6000 8000
Stratified Random Sampling (StRS)

Estimate the average weight, assuming that gender and age of individuals are readily available

Lower sample variance leads to faster convergence

Application to Power Estimation

Age & Gender → Zero-delay power estimate
Weight → PowerMill power estimate

Input vectors (population) → Zero-Delay Simulation of the Entire population → Population Stratification → Random Sampling and PowerMill Simulation → Convergent?

NO

YES

Report Power
Regression Estimation

Estimate the average height, assuming that weight of individuals is readily available

\[ H_2 = H_1 + \text{Slope} \times (W_2 - W_1) \]

H1: Avg. Height of Samples
W1: Avg. Weight of Samples
W2: Avg. Weight of Population

StRS Results

Speedup

USC M. Pedram
Sampling on FSMs

Method 1: Functional simulation + sampling on comb. circuit

Method 2: Do machine warm-up for every sampling unit

Probabilistic Compaction (PC)

Sequence compaction: Generate a new, but shorter, sequence which exhibits nearly the same spatio-temporal correlations as the initial sequence
**PC by Example**

Initial sequence:
01101010011111010
11100000110111000
10001101011110101
Avg. bit activity: 23/16

Compacted sequence:
001010
011000
010001
Avg. bit activity: 8/5

**Stochastic State Machine Model**

**Dynamic Markov Trees**

\[ S_1 = (0000, 0001, 1001, 1100, 1001, 1100, 1001, 1100) \]
Comparison with SRS

L = 4,000
Compaction Ratio 10

<table>
<thead>
<tr>
<th>Circuit</th>
<th>SRS</th>
<th>Hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td>C6288</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3540</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1908</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1355</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C880</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C432</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compaction for FSMs

\[
\begin{align*}
\text{Input Sequence} & \quad \text{Target Circuit} \\
L = 4,000 & \quad \text{Compaction Ratio 10} \\
\text{Logic} & \\
\text{FFs} & \\
\end{align*}
\]

Markov Chain

\[
\begin{align*}
\text{Logic} & : z_n = \text{out} (x_n, s_n) \\
\text{FFs} & : s_{n+1} = \text{next} (x_m, s_n) \\
\end{align*}
\]
Higher Order DMTs

A lag-k Markov chain which correctly models the input sequence, also models the joint k-step conditional probabilities of the primary inputs and state lines.

High Order DMT Results

Compaction Ratio 10

Order 1
Order 2

USC  M. Pedram

USC  M. Pedram
Instruction Level Compaction

Original Program

Architectural Simulation

Characteristic Profile

New Program

Program Synthesis

CPU’s RTL Model

RTL Simulation

Power Estimation

Characteristic Profile

- Instruction mix
- Average instruction size (if applicable)
- Clocks per instruction
- Branch prediction miss rate
- Instruction cache miss rate
- Data cache read/write miss rate
- Pipeline stall rate
- Speculative execution and register renaming are not considered
- Target micro-processor: A super-scalar pipelined CPU with branch prediction (Intel’s Pentium chip)
Program Synthesis Procedure

1. Block Allocation
   - add op1, op2
   - xor op1, op2
   - mul op1, op2
   - mov op1, mem1
   - branch

2. Instruction Allocation
   - add op1, op2
   - xor op1, op2
   - mul op1, op2
   - mov op1, mem1
   - branch

3. Memory Allocation
   - mov op1, mem1
   - memory space

4. Operand Allocation/Instruction Scheduling
   - reorder the sequence

Results: Compression Ratio

- GO
- M88
- GCC
- COMP
- LISP
- IJPE
- PERL
- VORTEX

 USC  M. Pedram

© M. Pedram
Nov 1997
Results: Accuracy

Energy per Instruction

GO
M88
GCC
COMP
LISP
IJPEG
PERL
VORTEX

UCS
M. Pedram

Power Macromodeling

Training Set
Macromodel Equation Form
Low - Level Simulation
Model Variables
Regression Analysis
Regression Coefficients
Analytic Model Reduction
Macro Models

UCS
M. Pedram
Dual Bit Type Model

Consider a data path block:

\[ Pwr = C_0 + C_1 \cdot S_1 + C_2 \cdot S_2 + C_3 \cdot S_3 + C_4 \cdot S_4 \]

- **Sign bit**
- **MSB region**
- **LSB region**

\( S_1, S_2 \): avg. switching activity of LSB (MSB) region of operand 1
\( S_3, S_4 \): avg. switching activity of LSB (MSB) region of operand 2

---

Input/Output Data Model

Consider a data path block:

\[ Pwr = C_0 + C_1 \cdot S_1 + C_2 \cdot S_2 + C_3 \cdot S_3 \]

- **Module**

\( S_1, S_2 \): avg. switching activity of operands 1 and 2
\( S_3 \): avg. switching activity of output
Bitwise Data Model

Consider a random logic block:

\[ Pwr = C_0 + \sum_{\text{inputs}} C_i \cdot S_i \]

\( S_i \): avg. switching activity of input signal \( i \)

More parameters lead to a higher degree of accuracy, but increase the computational overhead.

Cycle-Accurate Macro-Models

- Average Power
- Peak Power
- DC Voltage Drop
- Signal Integrity
- Switching Noise
- Power Distribution

Cycle-Accurate Macro-Models
Statistical Macromodel Construction

Power Optimization Flow

© M. Pedram

© M. Pedram

Jan 1998

Nov 1997
System and Behavioral Synthesis

- Resource Allocation and Sharing
- Interconnect Synthesis
- Multiple Supply Voltage Scheduling
- Selective Shut-off of System Modules
- HW-SW Partitioning and Mapping
- Communication Synthesis

RT-Level and Logic Synthesis

- Logic Restructuring and Minimization
- Technology Mapping
- Precomputation or Bypass Logic
- State Assignment and Bus Encoding
- Gated Clocking
- Retiming
Physical Design

- Transistor Re-ordering under a DSM Delay/Power Model
- Gate Sizing under a DSM Delay/Power Model
- Floorplanning with Integrated Power Plane Design
- Fanout Optimization under a DSM Delay/Power Model
- Bounded-Skew Gated Clock Routing
- P/G Network Design for Ground Bounce Control

Summary

- CAD flows and tools can reduce power dissipation in VLSI circuits and systems by a factor of 5 - 8 X over the next three years
- Process and voltage scaling can provide another factor of 8 - 12 X
- Commercial tools for gate-level and circuit-level power estimation and optimization exist
- High level power analysis and estimation tools are a key enabling technology
- Early Design planning and system-level design and power optimization tools are needed