2003 ASPDAC Tutorial:

Power, Timing and Signal Integrity in SoC Designs

Speakers:
Dr. Eli Chiprout, Intel Labs
Dr. Howard H. Chen, IBM Research
Dr. Lei He, Univ. of California, Los Angeles

Outline

- Section I: Interconnect circuit model generation
  - Dr. Lei He
- Section II: Detailed models for noise and timing
  - Dr. Eli Chiprout
- Section III: Current and power modeling
  - Dr. Lei He
- Section IV: Chip and package power supply noise analysis
  - Dr. Howard Chen

Current and Power Estimation

Professor Lei He
lhe@ee.ucla.edu

EE Department, UCLA
http://eda.ee.ucla.edu

Outline

- Maximum simultaneous switching current
  - Related to IR and L di/dt noise
  - Circuit MSSC
  - Cluster MSSC
- Power-up current with power gating
  - Related to P/G reliability
- High-Level estimation
  - MSSC
  - Gate count and power
  - Power-up current

Circuit MSSC

- **Circuit MSSC problem:**
  - **Given a circuit, find the Maximum Simultaneous Switching Current (MSSC) and the input vector that leads to such value**
- **MSSC estimation methods:**
  - **Genetic Algorithm(GA) based algorithm**
    - **[Jiang, et al, 97], [Jiang, et al, 00]**
  - **Timed Automatic Test Pattern Generation (Timed-ATPG) based algorithm**
    - **[Krstic-Cheng97]**

Genetic Algorithm

```
GnerationNumber = 1;
Randomly generate the initial generation;
While ( GenerationNumber <= MAX_GENERATION )
{
   Evaluate the fitness (current value) of each string;
   Apply selection to obtain parent-strings;
   Apply crossover and mutation to obtain child-strings;
   Produce the new generation using the child-strings;
   GenerationNumber ++;
}
Return MSSC and the corresponding input vector
```

ATPG based Algorithm

Definition:

$I_s(g_i)$: Max Switching Current of a single gate g_i with an output switch

***Justify_switch*(g_i): Justification of an output switch assignment to gate g_i by backtracing and implication**

Algorithm:

Sort all the gates g_i ($1<i<N$) in non-increasing order of $I_s(g_i)$;

While (exits unprocessed gates)

{

Select an unprocessed gate g_j with the largest $I_s(g_j)$;

If (*Justify_switch*(g_j) == SUCCESS)

{

Assign an output switch to gate g_j;

}

Mark g_j as a processed gate;

}

Return the input vector that generates MSSC;

- **Timing constraints can be added to become Timed-ATPG**

GA and Timed-ATPG based algorithm

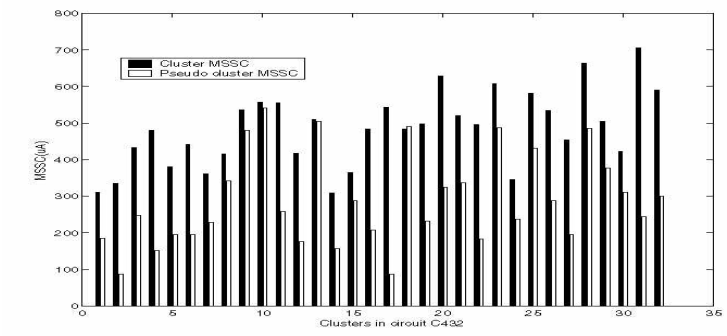
	GA based algorithm	Timed-ATPG based Algorithm
Computational Complexity	Low	High
Computational Scalability	High	Low
Estimation Quality	High	High
Flexibility for different abstract level	High	Low

Cluster MSSC-Problem formulation

- **Problem formulation**
  - **Given a circuit and a cluster, the cluster MSSC problem finds the MSSC value for the cluster and the input vector that leads to such value**
    - **Cluster size varies from several gates to a large module, depending on applications**
    - **For large cluster size, circuit MSSC estimation methods is applicable**
    - **For small cluster size, accuracy can be achieved with less efforts and efficiency is more important**
    - **New methods rather than GA and ATPG may be needed**

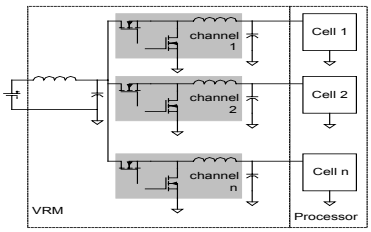
Experiments

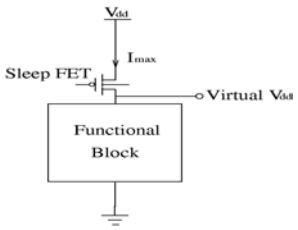
- **MSSC for small cluster size, gate number less than 20.**
  - **Psuedo Cluster based MSSC: MSSC value for cluster under vectors that maximizes circuit MSC**
  - **It leads to underestimation by up to 64%**



Power Gating

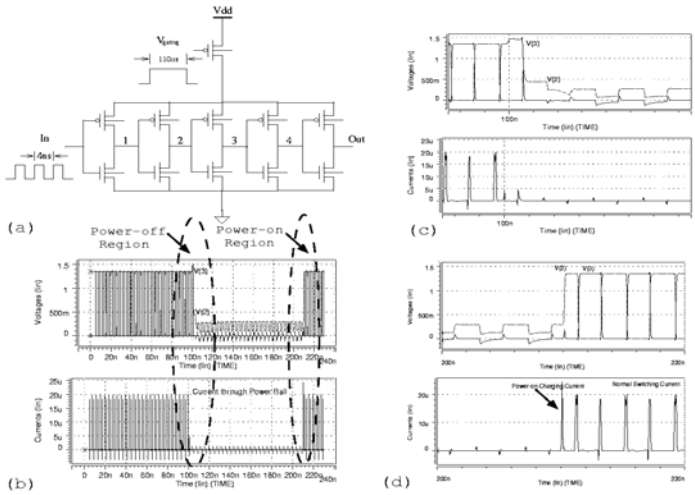
- Power-up (surge) current occurs when
  - Power up a module with individual VRM for dynamic power management
  - Turn on a macro-cell with sleep transistor for leakage power reduction





Power-up Current

- Power-up current may be much bigger than MSSC



Gate-Level Algorithm [Li-He01]

- Power-up current depends on one input vector, and MSSC depends on two input vectors
  - For combinational circuits
- ATPG, genetic algorithm, and random simulation may be used to find the vector leading to maximum power-up current

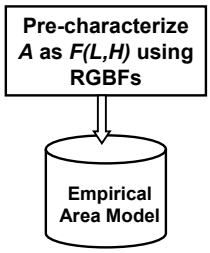
Circuits	Random Simulation	Timed-ATPG algorithm	Genetic Algorithm
Adder8	1.80E-03	1.81E-03	2.26E-03

Challenges of High-Level Gate Count Estimation

- Fast synthesis leads to 11x difference
  - Same logic function but with different granularity

Circuits	Gate Count	
	Smaller Granularity	Larger Granularity
1bit adder	3	3
4bit adder	20	16
8bit adder	42	490

Pre-characterization of Gate-Count Library [Nemani-Najm, TCAD'99]



- *L(f)*: linear measure
- *H(f)*: output entropy

$L(f) = L_1(f) + L_0(f)$

$L_1(f)$: the linear measure of on-set

$L_0(f)$: the linear measure of off-set

$$L_1(f) = \sum_{i=1}^{N} c_i p_i$$

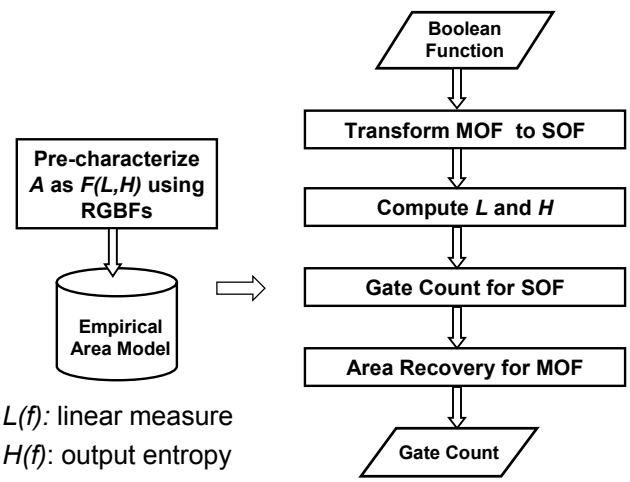
c_i is the size of the Primary Implicant PI_i and p_i is the probability that PI_i can be satisfied.

$L_0(f)$ is defined in a similar way.

$$H(f) = p\log_2(\frac{1}{p}) + (1-p)\log_2(\frac{1}{1-p})$$

p is the output probability of function f

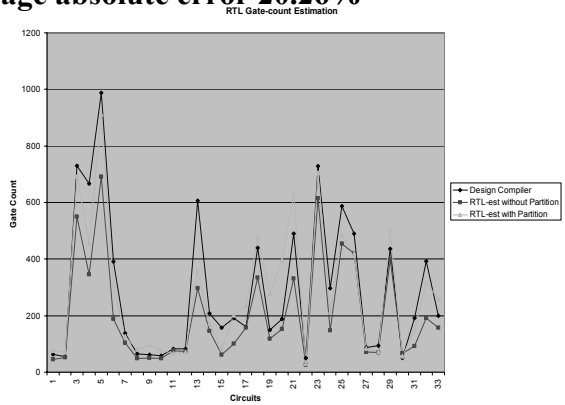
Gate-Count Estimation Flow



- *L(f):* linear measure
- *H(f)*: output entropy

Experiments of Gate Count Estimation

- Use the cell library and synthesis script in a CPU project
- Average absolute error 20.26%



High-Level Power Estimation

- Metric for average power [NemaniNajm99]
  - $P_{avg} = D_{avg} * A * C_{avg}$
- D_{avg}: average node transition density
  - Equivalent to average node entropy
  - Assuming quadratic decrease of node entropy as logic level increases
  - Average node entropy is deducted as

$$H = \frac{1}{N}\sum_{i=1}^{K}\sum_{g_i} H(g_i) = \frac{2/3}{n_{PI}+n_{PO}}(H_{PI}+2H_{PO})$$

- A: number of gates (gate count) in the circuit
- C_{avg}: average node capacitance

Experimental Results

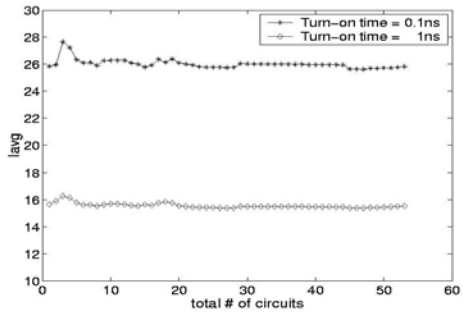
Circuit	Act. Power	Pred. Power	Abs. Err(%)
C8	318.60	397.40	24.73
S298	308.61	244.12	20.89
B9	321.13	212.58	33.80
S1196	618.52	810.52	23.5
S1238	996.75	672.07	32.58
S1494	1202.54	649.41	46.0
S1488	1328.34	712.99	46.32
Average	-	-	31.91

RT-level Power-up Current Estimation

- Total maximum power-up current is
  - $I_p = A \cdot I_{avg}$
- A: number of gates (gate count) in the circuit
- I_{avg} : weighted average power-up current
  - Weight: how frequently a cell is used
  - Different from the average of current for all cells in library

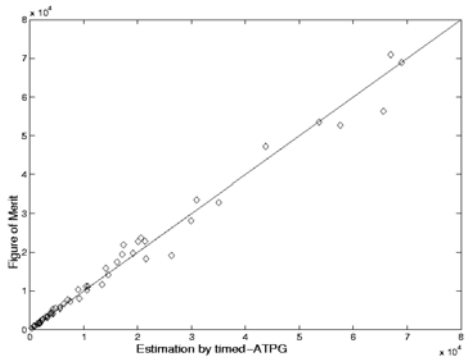
Computation of Iavg [LiHe02]

- I_{avg} computed by gate-level estimating of no more than 20 small RGBFs (or circuits similar to applications)
  - Based on regressive average
- One-time characterization for each turn-on time and each cell library



Verification of Metric Ip = A * Iavg

- Accurate gate count assumed
- MCNC and ISCAS benchmarks used (up to thousands of gates)



Experiments of RT-Level Estimation

- Use estimated A and I_{avg}
- Average absolute error 22.64%

circuit	Gate-level	High-level	Abs. Error(%)
con1	761.93	689.10	9.56
misex1	1672.34	1355.23	18.96
squar5	1656.82	1263.35	23.75
b9	3772.84	3950.84	4.72
5xp1	2954.62	2986.10	1.07
ttt2	4858.79	3790.05	24.71
apex7	6330.88	5283.10	16.55
x1	7443.08	5926.26	20.38
i6	14486.83	10612.14	26.75
x4	10840.71	11875.49	9.55
apex6	18879.98	12679.44	32.84
i8	23973.35	24394.14	1.76
apex3	26969.87	17916.60	33.56

References

- [Jiang, et al, 97] "Estimation of maximum power and instantaneous current using a genetic algorithm", *Proc. IEEE CICC*, pp. 135-138, May 1997
- [Jiang, et al, 00] "Estimation for maximum instantaneous current through supply lines for CMOS circuits", *IEEE Trans on VLSI*, vol. 8, pp. 61-73, Feb 2000
- [Krstic-Cheng97] "Vector generation for maximum instantaneous current through supply lines for CMOS circuits", *Proc Design Automation Conf.*, pp.383-388, Jue 1997
- [Li-He01] "Maximum Current Estimation Considering Power Gating", ISPD 2001
- [Nemani-Najm96] "Towards a High-Level Power Estimation Capability", *Trans on CAD* 1996
- [Nemani-Najm, TCAD'99] "High-Level Area and Power Estimation for VLSI Circuits", Trans on CAD 1999
- [Gupta-Najm02] "Energy and peak-current per-cycle estimation at RTL", *to appear, IEEE Transactions on VLSI*, 2002