Variability-Driven Module Selection with Joint Design Time Optimization and Post-Silicon Tuning

Feng Wang, Xiaoxia Wu, Yuan Xie

The Pennsylvania State University
Department of Computer Science & Engineering
Outline

- **Introduction**
  - Process Variation and its impact on HLS
  - Related work
- **Variability-Driven Module Selection**
  - Performance/Power yield
  - Design Time Approach
  - Post-silicon Tuning Approach
  - The combined approach
- **Experimental Results**
- **Conclusion**
What is the problem?

- Process variation has become a prominent concern as technology scales.
- Device and interconnect process variations increase with shrinking feature sizes.

(Source: K. Roy DAC05)

(Source: Intel)
Impact on High-Level Synthesis

- HLS schedules operations at different clock cycle and maps them to function units (FU).
- Traditionally, each FU has a fixed latency value.

However, under process variation....

(Source: K. Bernstein, IBM)
Worst-case analysis:
-- much larger variation -- very pessimistic

Require a shift in the design paradigm, from today's deterministic to probabilistic design.
Probabilistic Design Paradigm

A holistic design paradigm shift to statistical design

Variation-aware architecture
Variation-aware high level synthesis?
Statistical timing analysis
Statistical gate level optimization
Statistical technology mapping
Process variation modeling
Related work

- High-level synthesis is a well-studied problem
  - Low power: T. Kim TVLSI03, J. Cong ASPDAC08
  - Thermal: Seda ICCAD 06
- Physical information can also be integrated into HLS
  - H. Zhou DAC05
- Industry success story:
  - HLS tool “Catapult” (Mentor Graphics)
  - BlueSpec inc.
  - AutoESL
- Variation-aware HLS
  - W. Huang ICCAD06, T. Kim ICCAD07, S. P. Mohanty VLSID 07

variation-aware high level synthesis is still in its infancy
Outline

- Introduction
  - Process Variation and its impact on HLS
  - Related work
- Variability-Driven Module Selection
  - Performance/Power yield
  - Design Time Approach
  - Post-silicon Tuning Approach
  - The combined approach
- Experimental Results
- Conclusion
Performance Analysis/Yield

Performance yield: The probability that the synthesis hardware can work at a particular clock rate

A functional unit: \[ T_i = a_{0_i} + a_{1_i} \Delta V_{th} + a_{2_i} \Delta l + a_{3_i} V_{SB} \]

Synthesized DFG: Sum operation and Max operation

Performance Yield of the DFG:

\[
Yield_{\text{delay}}(DFG) = \text{Prob}(T_{\text{max}} \leq T_{\text{clock}} | \text{constraint } s)
\]

\[
Yield_{\text{delay}} = \prod_{i=1}^{M} Yield_{\text{delay}}(b_i)
\]

\[
\Delta Yield_{\text{delay}} = \prod_{i=1, i \neq j}^{M} \text{Yield}_{\text{delay}}(b_i) \times \Delta \text{Yield}_{\text{delay}}(b_j)
\]
**Power Analysis/Yield**

- **Power yield**: The probability that the total power less than the power limit

A functional unit: \[ P_i = \exp(b_0 i + b_1 \Delta V_{th} + b_2 \Delta l + b_3 V_{SB}) \]

Synthesized DFG: **Sum** of the random variables

Power Yield of the DFG:

\[
Yield_{\text{power}}(DFG) = \Pr(\text{prob} (P_{\text{tot}} \leq P_{\text{target}} | \text{constraints}))
\]

\[
P_{DFG}^{\text{new}} = P_{DFG}^{\text{old}} - P_{opt_k}^{\text{old}} + P_{opt_k}^{\text{new}}
\]

\[
\Delta \text{Yield} = \text{Yield}(P_{DFG}^{\text{new}}) - \text{Yield}(P_{DFG}^{\text{old}})
\]
Design Time Approach - example

**Worst case analysis:** Adder 2 is faster

- **CCT=T1:** Adder 1 is better
- **CCT=T2:** Adder 2 is better
- **CCT=T3:** Both Adders have the same yield (100%)

\[ \text{Yield} = \int_0^T p(t) \, dt \]
Design Time Approach - algorithm

- **Input**: initial scheduled DFG, constraints, module library
- **Output**: a synthesized DFG with optimized power and satisfied performance constraints

Evaluate the gain of each possible move, insert the move with highest gain to to_move_list

Meet constraint and $\Delta Yield \geq \varepsilon$

**Generate_multiple_moves**

**to_move_list**: find k moves to maximize the total gain $G_k$

If ($G_k > 0$)

Apply the moves, evaluate the power and performance yield

2 iterative steps:
Performance yield maximization and power yield improvement under performance yield constraint
Post Silicon Tuning

- Tuning chips after manufacturing, body biasing techniques by controlling threshold voltage
  - Reverse body biasing (RBB) reduces leakage power at the expense of slowing down circuits
  - Forward body biasing (FBB) improves performance at the expense of higher leakage power

- Adaptive body biasing (ABB) can tighten distribution of the performance and power, minimizing the yield loss due to process variation
Post Silicon tuning Approach

- Decide the optimal body biasing for a module selection decision such that the power yield is maximized under the performance constraints.

\[
\begin{align*}
\text{minimize:} & \quad P_{\text{tot}} \\
\text{subject to:} & \quad P(T_{\text{max}} \leq T_{\text{clock}} | \text{constraint } s) \geq \alpha
\end{align*}
\]

second order conic program

\[
\begin{align*}
\text{minimize:} & \quad (a1 + b \cdot a2)^T s \\
\text{subject to:} & \quad b^T s + \phi^{-1}(\alpha)(s^T \sum s)^{1/2} \leq T_{\text{limit}} \\
& \quad c^T (s - s_{\text{ini}}) \leq \varepsilon
\end{align*}
\]

vector \( s \) is to be determined, then Vsb
Joint optimization Approach

JointOpt (ISDFG, constraints, Library)
1. While ($\Delta \text{Yield} > \epsilon$ and meet constraints){
2. Design time module selection under current body bias;
3. Sequential Conic Optimization;
4. }

- The initial body bias is zero
- Maximize the power yield under performance yield constraints
- Iterates until no improvement can be obtained
- Output a synthesized DFG with optimal body bias
Outline

- Introduction
  - Process Variation and its impact on HLS
  - Related work
- Variability-Driven Module Selection
  - Performance/Power yield
  - Design Time Approach
  - Post-silicon Tuning Approach
  - The combined approach
- Experimental Results
- Conclusion
Experiment set up

- Algorithms in C++
- 90nm technology
- Six high level synthesis benchmarks:
  - A 16-point symmetric FIR filter (FF)
  - A 16-point elliptic wave filter (EWF)
  - An autoregressive lattice filter (ARF)
  - An algorithm for computing discrete cosine transform (DCT)
  - A differential equation solver (DES)
  - An IIR filter (IIR)
Power Yield Gain

- Design Time Approach vs. worst case
  - 90% performance yield constraint

Power Yield Gain for Different Benchmarks

34% power yield improvement
Power Yield Results

- Joint Approach vs. Design time only
  - 99% performance yield constraint

38% power yield improvement
Conclusion

- As technology scales, process variation has increasing impact on performance and power variations.
- Traditional synthesis techniques belong to design time approaches.
- We propose a yield driven module selection with joint design time optimization and post-silicon tuning.
Thank you!
Compare with Previous Works

- Only consider timing variability
- Every step is still deterministic
- Design time approach