Performance, Power & Energy

ELEC8106/ELEC6102
Spring 2010
Hayden Kwok-Hay So

What are the Options?

\[ P_{\text{total}} = \alpha \cdot C_L \cdot V_{dd} \cdot f_{clk} + I_{sc} \cdot V_{dd} + I_{\text{leakage}} \cdot V_{dd} \]

Dynamic Energy per operation:

\[ E_{\text{dyn}} = P_{\text{dyn}} / f_{clk} = \alpha \cdot C_L \cdot V_{sw} \cdot V_{dd} \]

Static Energy consumption:

\[ E_{\text{static}} = E_{\text{dyn}} \times \text{no. of operations} \]

Total Run Time:

\[ T_{\text{total}} = \text{no. of operations} \times CPI / f_{clk} \]

Power/Energy Reduction Techniques

- Smart CAD
  - Packing
  - Place and Route
- Turn off unused part of circuits
  - Clock gating
  - Multi-threshold devices
  - Precomputation/Guarded Evaluation
- Dynamic Voltage/Frequency Scaling
  - Does lowering clock speed work?

Capacitance Distributions

- Interconnect:
  - IXbar – input crossbar into a CLB
  - OXbar – output crossbar from CLB
  - Double – travel two CLBs
  - Hex – travel six CLBs
  - Long – travel entire chip
  - Long and global clock routing a factor of chip size

FPGA Power Consumption

Most power consumed in interconnect
Hex and Long power due to high capacitance
Double due to high activities

Low-power interconnect switches

- 3 operating modes
  - High-speed, low-power, sleep
- Leakage
  - 36-40% reduction from high-speed to low-power
  - 61% reduction in sleep mode
- Dynamic: up to 28% reduction
- Most existing routes have enough slack to operate in low-power mode

Clock Power

- Varied among designs
  - Depending on number of FFs?
- Clock network in FPGA is fixed
  - Usually global H-tree + local routes
- Clock power:
  - Activity = constant
  - Reduce capacitance? Swing?
- Modern FPGA divided fabric into clock regions
  - Local buffer can be disabled
  - Reduced overall effective capacitance
  - Fewer active clock region → lower clock power

Clock Gating – Example

- Consider 4 architectures
  - Fine Gran vs Coarse Gran
  - Region-based vs Column Based

Save...

- Dynamic power from gated clock regions
- Capacitance
- Clock enable signal (?)

But...

- Need to group related logic into gated region
- Fine-grain:
  - More opportunities to disable clock
  - But high buffer overhead
- Coarse-grain:
  - Low overhead
  - Less opportunities
- Module-based clock gating?
Clock Gating Example (2)

- Works best when the original designs have low activity factor
  - More opportunities for clock gating
- Clock power reduced > 50% in some cases
- Overall power reduction ~6.5%
- -ve reduction in high activity circuits
  - Increased in logic power

Precomputation

- Dynamically turns off part of the circuit depending on input data at run-time
- Up to 80% reduction in dynamic power consumption

Adder with precomputation

Normalized dynamic power consumption of adders with precomputation

Ripple Carry Adders  Carry Lookahead Adders

Multiplier with precomputation

Normalized dynamic power consumption for multipliers

Virtex-5 FPGA DSP block
Multi-Vth, Vdd Devices
- Signals for clock gating (or similarly generated EN signals) can be used to turn off/slow down part of FPGA
- High Vth transistors used to stop leakage current
- Varying Vdd provides different energy-delay tradeoff
  - Spatial: Multi-Vdd devices
  - Temporal: DVS, DVFS

Multi-Vdd devices
- CAD tools map the parts of circuit with enough slack to low Vdd tiles
- ~14% reduction in power consumption in practice

But...

Dynamic Voltage Scaling
- Energy/Power consumption decreases when Vdd decreases
- But delay also increases
- Need to find the correct balancing point
- DVS find the best operating Voltage dynamically
- Save energy AND power

DVS
- Computer doesn't need to be "too" fast
- Save energy by reducing Vdd when not in high demand
  - Between events

---

**Figure 8:** Pre-defined dual Vdd layout patterns for dual Vdd logic block (left).

**Figure 9:** Power vs. delay for dual Vdd.

**Figure 10:** Power vs. delay for dual Vdd.

---

**Table 1:** Power saving obtained by predefined dual-Vdd (dash Vdd) blocks at the same target clock frequency. The clock frequency is chosen so the product clock frequency achieved by each Vdd is:

<table>
<thead>
<tr>
<th>Vdd</th>
<th>Delay (ns)</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0V</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>0.9V</td>
<td>15</td>
<td>75</td>
</tr>
<tr>
<td>0.8V</td>
<td>20</td>
<td>50</td>
</tr>
</tbody>
</table>

---

DVS results

- Optimal saving up > 80%
- Real scheduling algorithm reduces saving

In Conclusion...

- Power and Performance are closely related
- Decreasing power consumption almost always imply lower performance
- Architectural changes provide extra degree of freedom to tradeoff power-performance
- Reconfigurable computing systems are capable of exploiting these architectural choices, even on a per-application basis