Performance Metrics & Architectural Adaptivity

ELEC8106/ELEC6102
Spring 2010
Hayden Kwok-Hay So
What are the Options?

$$P_{\text{total}} = \alpha(C_L) \cdot V_{sw} \cdot V_{dd} \cdot f_{clk} + I_{sc} \cdot V_{dd} + I_{\text{leakage}} \cdot V_{dd}$$

**Power Consumption**
- Activity factor (amount of circuit switching)
- Load Capacitance (size of circuit)
- Voltage Swing
- Supply Voltage
- Clock frequency

**Dynamic**
$$E_{op} \approx \frac{P_{\text{dyn}}}{f_{clk}} = \alpha \cdot C_L \cdot V_{sw} \cdot V_{dd}$$

**Static**

**Energy per “operation”**

**Total Energy Consumption**

$$E_{total} = E_{op} \times \text{no. of operations}$$

**Total Run Time**

$$T_{total} = \frac{\text{no. of operations} \times CPI}{f_{clk}}$$
Metrics

- How do we quantify Performance, Power and Energy?
- How do we quantify Energy Efficiency?
- Many different quantities:
  - Power-delay product
  - Energy-delay product
  - Performance/Power
  - Performance^2/Power
**Absolute wall clock time**

- Measures the absolute wall clock time to finish a task
  - Compared to virtual OS time, etc
- Measure in seconds
- Includes all overhead from hardware, software, as well as the OS

**Cons**

- Doesn’t tell you the portion of time spent because of time sharing machine, OS overhead, etc
  - Solution: run in single-user mode
IPS, kIPS, MIPS, bIPS

- IPS = instruction per second
- Measures throughput of a computer
- A very simple first-order estimation of how much work a computer can perform per unit time.
  - A machine that completes 1 instruction per cycle running at 1 Megahertz = 1 MIPS
- Varies GREATLY depending on the input benchmark
  - IPS = IPC * Cycle per second (Hz)
- Very often used, but can be very misleading
FLOPS

- Floating Point Operations per Second
- Similarly calculated as MIPS
- A machine that performs 1 floating point operation per cycle running at 1 MHz is running at 1 FLOPS
- Similar to MIPS, FLOPS rating of a system depends highly on the input workload
  - Also, system performances, e.g. memory, bus
- E.g. top500 list use FLOPS to measure super computers using LINPACK
  - The sustained FLOPS of a system is used to rank, not the theoretical peak.
Energy and Power

- Power: Watt (W)
- Energy: Joules (J)
  - kWh, (mAh?)
- Power = Energy consumed per unit time
Energy efficiency

- How much performance can you get per unit of energy spent

- Three possible measurements:
  - 1. Power/Throughput
  - 2. Power/Throughput^2
  - 3. BIPS/w
Power/Throughput

- Good for **fixed throughput** systems
  - Such as DSP systems
  - Data stream into system and processed continuously

\[
\frac{\text{Power}}{\text{Throughput}} = \frac{\text{Energy/time}}{\text{Operation/time}} = \frac{\text{Energy}}{\text{Operation}}
\]

- In such fixed throughput systems, number of operation per unit time is constant

- Drawback: Doesn’t take performance into account
  - System A may be better than System B in terms of Energy/Op, but System B may be much faster

- Similar to Power-Delay-Product in circuit design
Power/Throughput$^2$

$$ETR = \frac{E_{\text{MAX}}}{T_{\text{MAX}}} = \frac{\text{Power}}{\text{Throughput}^2}$$

- Balances both Energy/Op and Throughput
- If system A has lower ETR than System B, then either
  - 1. A has lower energy/operation than B at the same throughput, OR
  - 2. A has higher throughput than B at the same energy/operation
- Similar to Energy-Delay-Product in circuit designs,
  - But hasn’t taken into account architectural effects
Optimizing ETR

- Need to apply power-delay tradeoff
- Vdd scaling only works within a limited range
- Frequency scaling alone doesn’t work
  - Frequency doesn’t affect E/op → lowered frequency decrease throughput → higher ETR

Figure 2: ETR as a function of $V_{DD}$. Inefficient to operate at high supply voltages (greater than 3.3 V)
Optimizing ETR

Algorithm and Architectural changes most effective!

Architecture:

- If a machine can perform S operations at the same time, throughput is increased by S while energy/op is constant $\Rightarrow$ ETR decreased by S

Algorithm:

- If the length of an application is reduced by a factor of S
- Then, switching capacitance is reduced by S, throughput increased by S, so ETR is reduced by $S^2$
FPGA coprocessing on ETR

- Offloading computation to FPGA very likely improve ETR

  1. Decreases code size in sw ($S^2$ factor)
  2. Decreases overall run time (overall E)
  3. Increases operation/cycle

But

- Increased seconds/cycle $\Rightarrow$ decreases operation per second
Efficiency Trends and Limits from Comprehensive Microarchitectural Adaptivity

Benjamin C. Lee and David Brooks, ASPLOS’08
Overview

- Studied the effectiveness of architectural adaptation
- 15 different architectural parameters are adapted to the running application both temporally and spatially

<table>
<thead>
<tr>
<th>Set</th>
<th>Parameters</th>
<th>Measure</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>Depth</td>
<td>depth</td>
<td>FO4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>width</td>
<td>9:3:36</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Width</td>
<td>issue b/w</td>
<td>2.4, 8</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Branch Predictor</td>
<td>BTB associativity</td>
<td>1, 2, 4, 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTB size, $log_2$</td>
<td>12::1::15</td>
</tr>
<tr>
<td>$S_4$</td>
<td>Load/Store</td>
<td>load/store queue</td>
<td>9::5::54</td>
</tr>
<tr>
<td>$S_5$</td>
<td>Physical Registers</td>
<td>general purpose (GP)</td>
<td>40::10::130</td>
</tr>
<tr>
<td></td>
<td></td>
<td>floating-point (FP)</td>
<td>40::8::112</td>
</tr>
<tr>
<td></td>
<td></td>
<td>special purpose (SP)</td>
<td>42::6::96</td>
</tr>
<tr>
<td>$S_6$</td>
<td>Reservation Stations</td>
<td>branch</td>
<td>count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fixed-point/memory floating-point</td>
<td>count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>entries</td>
<td>6::1::15</td>
</tr>
<tr>
<td>$S_7$</td>
<td>I-L1 Cache</td>
<td>i-L1 cache size</td>
<td>16::2x::256</td>
</tr>
<tr>
<td>$S_8$</td>
<td></td>
<td>i-L1 cache assoc.</td>
<td>1, 2, 4, 8</td>
</tr>
<tr>
<td>$S_9$</td>
<td>D-L1 Cache</td>
<td>d-L1 cache size</td>
<td>8::2x::128</td>
</tr>
<tr>
<td>$S_{10}$</td>
<td></td>
<td>d-L1 cache assoc.</td>
<td>1, 2, 4, 8</td>
</tr>
<tr>
<td>$S_{11}$</td>
<td></td>
<td>load/store latency</td>
<td>1::1::5</td>
</tr>
<tr>
<td>$S_{12}$</td>
<td></td>
<td>L2 cache size</td>
<td>MB 0.25::2x::4</td>
</tr>
<tr>
<td>$S_{13}$</td>
<td></td>
<td>L2 cache assoc.</td>
<td>1, 2, 4, 8</td>
</tr>
<tr>
<td>$S_{14}$</td>
<td></td>
<td>L2 cache latency</td>
<td>8::2::16</td>
</tr>
<tr>
<td>$S_{15}$</td>
<td></td>
<td>Memory</td>
<td>memory latency</td>
</tr>
</tbody>
</table>

Table 2. Design space parameters where $i::j::k$ denotes a set of values from $i$ to $k$ in steps of $j$. 
Overview

- Temporal and Spatial sampling to reduce exploration space
- Genetic algorithm is used to find optimal configuration for an application at a given time

<table>
<thead>
<tr>
<th></th>
<th>amm</th>
<th>app</th>
<th>equ</th>
<th>gcc</th>
<th>gzi</th>
<th>jbb</th>
<th>mcf</th>
<th>mes</th>
<th>cho</th>
<th>oce</th>
<th>rad</th>
<th>ray</th>
<th>bla</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1</td>
<td>depth</td>
<td>9</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>33</td>
<td>9</td>
<td>18</td>
<td>36</td>
<td>30</td>
<td>27</td>
<td>30</td>
<td>24</td>
</tr>
<tr>
<td>S_2</td>
<td>width</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>S_3</td>
<td>bp</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>S_4</td>
<td>lsq</td>
<td>31</td>
<td>36</td>
<td>31</td>
<td>31</td>
<td>11</td>
<td>26</td>
<td>11</td>
<td>11</td>
<td>41</td>
<td>41</td>
<td>26</td>
<td>56</td>
</tr>
<tr>
<td>S_5</td>
<td>reg</td>
<td>80</td>
<td>130</td>
<td>70</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>130</td>
<td>130</td>
</tr>
<tr>
<td>S_6</td>
<td>resv</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>S_7</td>
<td>i1Size(KB)</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>S_8</td>
<td>i1Assoc</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S_9</td>
<td>d1Size(KB)</td>
<td>8</td>
<td>16</td>
<td>8</td>
<td>64</td>
<td>32</td>
<td>8</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>8</td>
</tr>
<tr>
<td>S_10</td>
<td>d1Assoc</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S_11</td>
<td>d1Lat</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>S_12</td>
<td>l2Size(MB)</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.25</td>
</tr>
<tr>
<td>S_13</td>
<td>l2Assoc</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>S_14</td>
<td>l2Lat</td>
<td>8</td>
<td>8</td>
<td>14</td>
<td>14</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>S_15</td>
<td>memLat</td>
<td>90</td>
<td>85</td>
<td>70</td>
<td>70</td>
<td>115</td>
<td>115</td>
<td>70</td>
<td>115</td>
<td>115</td>
<td>70</td>
<td>115</td>
<td>115</td>
</tr>
</tbody>
</table>

Table 1. Static baseline configurations that maximize each application’s efficiency.
BIPS$^3$ per Watt

- Lee and Brooks used bips$^3$/w as a metrics

- Argument:
  - voltage invariant

\[ \therefore w \propto V^2 f \quad \text{and} \quad f \propto V \]
\[ \therefore w \propto f^3 \]
\[ \therefore \text{bips} \propto f \]
\[ \therefore \frac{\text{bips}^3}{w} = k \]

- Similar to \( \frac{\text{Power}}{\text{Throughput}^3} \)

- What’s the catch?
Temporal Adaptivity

- Optimal architecture evaluated after a block of instructions have been executed
- Any or all of the 15 parameters can be adapted to their optimal values.
- Higher temporal adaptivity \(\Rightarrow\) higher overall efficiency

Improved performance

Improved power
Parameter Adaptation

- High temporal adaptivity
  - ➔ smaller # of parameters changed
  - ➔ smaller parameter delta values
  - ➔ more high value changes
Spatial Adapativity

- What if not all 15 parameters can be changed at the same time?
- Pick the 3 optimal parameters? Which 3?

<table>
<thead>
<tr>
<th>$S_i$</th>
<th>amm</th>
<th>app</th>
<th>equ</th>
<th>gcc</th>
<th>gzi</th>
<th>jjb</th>
<th>mcf</th>
<th>mes</th>
<th>cho</th>
<th>oce</th>
<th>rad</th>
<th>ray</th>
<th>bla</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>depth</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$S_2$</td>
<td>width</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_3$</td>
<td>bp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_4$</td>
<td>lsq</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>$S_5$</td>
<td>reg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_6$</td>
<td>resv</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2*</td>
</tr>
<tr>
<td>$S_7$</td>
<td>l1Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_8$</td>
<td>l1Assoc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_9$</td>
<td>d1Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{10}$</td>
<td>d1Assoc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{11}$</td>
<td>d1Lat</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{12}$</td>
<td>l2Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{13}$</td>
<td>l2Assoc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{14}$</td>
<td>l2Lat</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{15}$</td>
<td>memLat</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Limited spatial adaptivity

- Most benefits come from the 3 most important parameters

But...

- Some benchmarks require adapting all 15 parameters to be effective
### DVFS as adaptation

- During each iteration, after spatial parameters are changed, adjust voltage and the corresponding frequency to maximize $m3pw$

- Since all bars relative to the

---

H. So, Sp10  

Lecture 4 - ELEC8106/6102
DVFS as adaptation

- During each iteration, after spatial parameters are changed, adjust voltage and the corresponding frequency to maximize $m_{3pw}$

- DVFS only useful in a few cases in static (non-adapting) cases
  - Why?
In conclusion…

- Metrics for energy efficiency must be carefully constructed to suit actual need
  - Energy/Throughput seems reasonable for this class
  - Throughput$^3$/Power also used

- Offloading computation to reconfigurable hardware accelerators can significantly improve energy efficiency metrics

- Reconfigurable systems allow flexible tradeoffs in energy efficiency tradeoff

- Reconfigurable systems allow run time architectural adaptation that can improve energy efficiency by factor of 5