Computer Structure

Power and Power Management

Lihu Rappoport and Adi Yoaz
Thermal Design Point

**TDP power**
- Maximum amount of power the thermal solution of the platform is required to dissipate
- Calculated as the rolling average of 5sec power of the highest real existing applications
- Not including Viruses
  - Some virus like application are ignored as well

**TDP power impacts**
- Guaranteed frequency
- Affects form factor
- Cooling solution cost
- Acoustic noise

How much power can this fan can dissipate?
Average Power

- **Average Power**
  - Average power = Total Energy / Total time
  - Including low-activity and idle-time (~90% idle time for client)

- **Average Power determines**
  - Battery life – for mobile devices
  - Electricity bill
  - Air-condition bill
  - Air pollution

Battery Life
Continuous Web surfing over wireless

<table>
<thead>
<tr>
<th>Laptop Model</th>
<th>Battery Life</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenovo ThinkPad T450s (extended battery)</td>
<td>15:26</td>
</tr>
<tr>
<td>Lenovo ThinkPad X250 (extended battery)</td>
<td>15:12</td>
</tr>
<tr>
<td>Acer Aspire One Cloudbook (14-inch)</td>
<td>14:43</td>
</tr>
<tr>
<td>Toshiba Portege Z20t (with keyboard dock)</td>
<td>14:37</td>
</tr>
<tr>
<td>Apple MacBook Air (13-inch, early 2015)</td>
<td>14:00</td>
</tr>
<tr>
<td>Dell Chromebook 13 - 7310</td>
<td>13:25</td>
</tr>
<tr>
<td>Surface Book (Core i5, Integrated GPU)</td>
<td>12:29</td>
</tr>
<tr>
<td>ASUS X205</td>
<td>12:05</td>
</tr>
<tr>
<td>Apple MacBook Pro with Retina Display (13-inch, 2015)</td>
<td>12:04</td>
</tr>
<tr>
<td>Dell XPS 13 (2015, Nontouch)</td>
<td>11:54</td>
</tr>
<tr>
<td>Lenovo 100S Chromebook</td>
<td>11:19</td>
</tr>
</tbody>
</table>

1 web search
- Same energy as an 11W light bulb for 1 hour
- Emits 7gr CO₂
- 4B web searches daily
Platform Power

- Processor average power is <10% of the platform
CPU Power Components

- **Dynamic power:** $P_{dyn} = CV^2f$
  - $C$ – total electrical capacitance charged/discharged per cycle
    - The sum of all the capacities of all transistors and wires which are charged/discharged (toggle from 0→1 or from 1→0) per cycle
      - Transistors which maintain their value do not spend dynamic power
    - Application dependent
      - E.g., an application with no floating point calculation will not toggle the transistors in the floating point execution unit
    - Typically, a bigger CPU has a bigger $C_{dyn}$
  - $V$ – voltage
  - $f$ – frequency: increasing $f$ also requires increasing $V \sim \text{linearly}$
    - $CV^2f \sim f^3 \Rightarrow X\%$ frequency costs $\sim 3X\%$ power

- **Leakage power**
  - Leakage of transistors under voltage, which is a function of
    - $Z$ – total size of all transistors, $V$ – voltage, $t$ – temperature
Performance per Watt

- In small form-factor devices thermal budget limits performance
  - Old target: get max performance
  - New target: get max performance at a given power envelope
    - Performance per Watt

- Increasing f also requires increasing V (≈linearly)
  - Dynamic Power = αCV^2f = Kf^3 \Rightarrow X\% \text{ performance costs } \sim 3X\% \text{ power}
  - A power efficient feature – better than 1:3 performance : power
    - Otherwise it is better to just increase frequency (and voltage)

- Vmin is the minimal operation voltage
  - Once at Vmin, reducing frequency no longer reduces voltage
  - At this point a feature is power efficient only if it is 1:1 performance : power

- Active energy efficiency tradeoff
  - \( Energy_{active} = Power_{active} \times Time_{active} \approx Power_{active} / Perf_{active} \)
  - Energy efficient feature: 1:1 performance : power
Managing Power

• Typical CPU usage varies over time
  - Bursts of high utilization & long idle periods (~90% of time in client)

• Optimize power and energy consumption
  - High power when high performance is needed
  - Low power at low activity or idle

• Enhanced Intel SpeedStep® Technology
  - Multi voltage/frequency operating points
  - OS changes frequency to meet performance needs and minimize power
  - Referred to as processor Performance states = P-States

• OS notifies CPU when no tasks are ready for execution
  - CPU enters sleep state, called C-state
  - Using MWAIT instruction, with C-state level as an argument
  - Tradeoff between power and latency
    ➢ Deeper sleep → more power savings → longer to wake
P-states

- **Operation frequencies are called P-states = Performance states**
  - P0 is the highest frequency
  - P1,2,3... are lower frequencies
  - Pn is the min Vcc point = Energy efficient point

- **DVFS = Dynamic Voltage and Frequency Scaling**
  - Power = CV^2f ; f = KV \implies \text{Power } \sim f^3
  - Program execution time \sim 1/f
  - E = P \times t \implies E \sim f^2
    \implies \text{Pn is the most energy efficient point}

- Going up/down the cubic curve of power
  - High cost to achieve frequency
  - large power savings for some small frequency reduction
C-States: C0

- **C0**: CPU active state
C-States: C1

- **C0**: CPU active state
- **C1**: Halt state:
  - Stop core pipeline
  - Stop most core clocks
  - No instructions are executed
  - Caches respond to external snoops
C-States: C3

- C0: CPU active state
- C1: Halt state:
  - Stop core pipeline
  - Stop most core clocks
  - No instructions are executed
  - Caches respond to external snoops
- C3 state:
  - Stop remaining core clocks
  - Flush internal core caches
C-States: C6

- **C0: CPU active state**
- **C1: Halt state:**
  - Stop core pipeline
  - Stop most core clocks
  - No instructions are executed
  - Caches respond to external snoops
- **C3 state:**
  - Stop remaining core clocks
  - Flush internal core caches
- **C6 state:**
  - Processor saves architectural state
  - Turn off power gate, eliminating leakage

Core power goes to ~0
Putting it all together

- CPU running at max power and frequency
- Periodically enters C1
Putting it all together

- **Going into idle period**
  - Gradually enters deeper C states
  - Controlled by OS
Putting it all together

- Tracking CPU utilization history
  - OS identifies low activity
  - Switches CPU to lower P state
Putting it all together

- CPU enters Idle state again
Putting it all together

- Further lowering the P state
- DVD play runs at lowest P state
Voltage and Frequency Domains

- **Two Independent Variable Power Planes**
  - CPU cores, ring and LLC
    - Embedded power gates – each core can be turned off individually
    - Cache power gating – turn off portions or all cache at deeper sleep states
  - Graphics processor
    - Can be varied or turned off when not active

- **Shared frequency for all IA32 cores and ring**

- **Independent frequency for PG**

- **Fixed Programmable power plane for System Agent**
  - Optimize SA power consumption
  - System On Chip functionality and PCU logic
  - Periphery: DDR, PCIe, Display
• **P1 is guaranteed frequency**
  - CPU and GFX simultaneous heavy load at worst case conditions
  - Actual power has high dynamic range

• **P0 is max possible frequency – the Turbo frequency**
  - P1-P0 has significant frequency range (GHz)
    - Single thread or lightly loaded applications
    - GFX <>CPU balancing
  - OS treats P0 as any other P-state
    - Requesting is when it needs more performance
  - P1 to P0 range is fully H/W controlled
    - Frequency transitions handled completely in HW
    - PCU keeps silicon within existing operating limits
  - Systems designed to same specs, with or without Turbo Mode

• **Pn is the energy efficient state**
  - Lower than Pn is controlled by Thermal-State
Turbo Mode

Power Gating
Zero power for inactive cores

No Turbo

Workload Lightly Threaded
Turbo Mode

Power Gating
Zero power for inactive cores

Turbo Mode
Use thermal budget of inactive core to increase frequency of active cores

No Turbo

Workload Lightly Threaded
Turbo Mode

No Turbo

Power Gating
Zero power for inactive cores

Turbo Mode
Use thermal budget of inactive core to increase frequency of active cores

Workload Lightly Threaded
Turbo Mode

Increase frequency within thermal headroom

Active cores running workloads < TDP

Turbo Mode

No Turbo

Frequency (F)

Core 0  Core 1  Core 2  Core 3

Frequency (F)

Core 0  Core 1  Core 2  Core 3
Turbo Mode

No Turbo

Power Gating
Zero power for inactive cores

Turbo Mode
Increase frequency within thermal headroom

Workload Lightly Threaded
And active cores < TDP
Thermal Capacitance

**Classic Model**
Steady-State Thermal Resistance

Design guide for steady state

**New Model**
Steady-State Thermal Resistance AND Dynamic Thermal Capacitance

*Temperature rises as energy is delivered to thermal solution*

*Thermal solution response is calculated at real-time*

*More realistic response to power changes*
After idle periods, the system accumulates “energy budget” and can accommodate high power/performance for a few seconds.

In Steady State conditions the power stabilizes on TDP.

Use accumulated energy budget to enhance user experience.
Core and Graphic Power Budgeting

- Cores and Graphics integrated on the same die with separate voltage/frequency controls; tight HW control
- Full package power specifications available for sharing
- Power budget can shift between Cores and Graphics

### Core Power [W] vs. Graphics Power [W]

<table>
<thead>
<tr>
<th>Applications</th>
<th>Core Power Specification</th>
<th>Graphics Power Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heavy CPU workload</td>
<td>Sandy Bridge Next Gen Turbo for short periods</td>
<td></td>
</tr>
<tr>
<td>Heavy Graphics workload</td>
<td>Realistic concurrent max power</td>
<td></td>
</tr>
<tr>
<td>Total package power</td>
<td>Sum of max power</td>
<td></td>
</tr>
</tbody>
</table>

Foil taken from IDF 2011
Example – Power/Performance Exam Question

You have to design a Multi-Processor Micro Server and you have to compare two types of processors so that the hot applications (TDP) can run efficiently while still achieving the highest possible performance in the budget.

The power of the system is at most 60W (Uncore – Core’s). The power consumption of the whole system is 0.25W per square millimeter (Leakage Power).

The dynamic capacity of each processor is a function of the IPC of the application it is running and it is:

\[ C_{dyn} = IPC \times 750\text{pF} \]

You have to compare two processors, a large one and a small one, and decide which one to use:

- The large processor is 4 wide with IPC = 3
- The small processor is 3 wide with IPC = 2

The TDP of the large processor is 4W and the small processor is 3W.
### Example – Power/Performance Exam Question (Cont’)

לحلول נתונה טבלת המריאה את נקודת מעבדית מתח ותדר אפקטיבי לрабידת המעבדיים הגדול והקטן.

<table>
<thead>
<tr>
<th>Volt’s 범위</th>
<th>당연한 범위 (Ghz)</th>
<th>תדר בGhz מעבד גודל</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vmin=0.7</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>Vmin=0.7</td>
<td>1.25</td>
<td>1</td>
</tr>
<tr>
<td>Vmin=0.7</td>
<td>1.45</td>
<td>1.35</td>
</tr>
<tr>
<td>0.8</td>
<td>1.75</td>
<td>1.75</td>
</tr>
<tr>
<td>0.85</td>
<td>2</td>
<td>2.25</td>
</tr>
<tr>
<td>0.9</td>
<td>2.25</td>
<td>2.5</td>
</tr>
<tr>
<td>0.95</td>
<td>2.5</td>
<td>3</td>
</tr>
<tr>
<td>Vmax=1</td>
<td>2.75</td>
<td>3.5</td>
</tr>
</tbody>
</table>

יש לتكن שתי מערכות אחת עם מעבדים גדולים והשנייה עם מעבדים קטנים כדי לקום את האפליקציות הת目标任务ים (TDP) הפיזיקליים נוקט על קבלת ביצועים הובונים יותר במסגרת הקצרים הטיפוסיים. עבורי של מערכתי יש למציא את מספר המעבדי הדרושים.
**Solution**

Power = leakage + c×f×V²
Leakage of the core big Core = 4×0.25W= 1W
Leakage of the core small Core = 2×0.25W= 0.5W

To maximize Total MP performance need to work at Vmin for TDP app
Big Core: V= 0.7v  f=1.35Ghz
Small Core V= 0.7v  f=1.45Ghz

For each system need to calculate the per Core Power budget for most efficient operating point which is Vmin when Hot App is running, this will allow maximizing number of cores and overall performance:
For Big Core System: P = C×V²×f + 1 ⇒ 3×0.75×0.7²×1.35+1=2.49W
For Small Core System: P= C×V²×f + 0.5 ⇒ 2×0.75×0.7²×1.45+0.5=1.57W

Power budget for all cores=60×2/3=40W
Number of Cores: Big 40/2.49 = 16   Small 40/1.57= 25

When building a system with big Cores putting 16 big Cores will provide highest MP performance in the system power envelop

When building a system with small Cores putting 25 small Cores will provide highest MP performance in the system power envelop
MP Perf/Power Q&A

MP Performance = # of cores × IPC × f

For Big Core: f = 1.35Ghz; IPC_{tdp} = 3; # of cores = 16
TDP MP Performance = 16 × 3 × 1.35 = 64.8 × 10^9 Instructions per second

For small Core: f = 1.45Ghz; IPC_{tdp} = 2; # of cores = 25
TDP MP Performance = 25 × 2 × 1.45 = 72.5 × 10^9 Instructions per second

The system with the small cores provides the highest MP performance and takes less area (25×2=50mm^2 vs 16×4=64mm^2) so we recommend to build the system with the small cores provide highest MP performance in the system power envelop.