Presentation 11

Energy Efficient Architecture

מערכות "ירוקות" או מערכת חוסכות חשמל מתוכננות לצמצם את צריכת האנרגיה שלהן בשימוש השגרתי. בתחילת המאה המניע העיקרי לחסכון בחשמל במחשבים היה שמירה על הסוללה במחשבים ניידים. בשנים שעברו מאז הנושא תופס יותר חשיבות ככל שמבינים יותר את המשמעות של צריכת חשמל לא מבוקרת במערכות מחשוב גדולות. פיתוחים חדשים בפיסיקה והכימיה של סוללות האריכו את הזמן היעיל של מחשב נייד לפחות פי חמש ב- 15 השנים האחרונות. הגישה בתכנון חומרת מחשוב נשארת כמו תמיד — כשלא משתמשים ברכיב מסוים, כבו אותו.

Slide 2 **Energy Efficiency**

The initial motivation for energy efficient devices began with the development of mobile computing. A large market potential was identified in the 1990s with the standardization of WiFi: wireless networking would make laptop use more attractive and energy efficient laptops would make WiFi systems more attractive. Both systems were limited by battery life-time.

A subsequent insight is that electricity consumption is a large factor in computing cost. Even without the cost of cooling, the electric bill for a large server farm can be estimated as

(500 W / PC server) × 1000 servers = 500 kW

500 kW × 720 hours / month = 360,000 kW / month

292,000 kW-Hr / month × 0.54 shekels / kW-Hr = 157,680 ח"ש / month

Power consumption also limits the performance of computing systems. Since

Performance / rack = (performance / kW) × (kW / rack)

any improvement to performance / kW improves the performance of large systems. As mainframes have become more popular, the relative energy efficiency they provide through centralization has become a highly competitive factor.

Slide 3 – 4 **Power Consumption**

Power is consumed in digital circuits during $0 \rightarrow 1$ and $1 \rightarrow 0$ state transitions. Since power is defined as energy consumed per second, it is shown in slide 3 that power consumption for a given program equals the energy per task divided by the program run time. Therefore it is possible to reduce the power consumption by lowering the clock frequency (making the program run more slowly), or by reducing the energy transition (by raising the energy efficiency of each transistor or using fewer transistors). These relationships are summarized on slide 4 as

Energy per task = power × run time

Performance = instructions per clock cycle × clock frequency

Slide 5 – 6 **System Issues**

Research into power consumption in laptop systems found that the CPU only consumes about 10% of the power required by the system. The peripherals (LCD, hard disk, memory system, networking components, I/O system) consume about 90% of the power. Therefore, making the CPU run more slowly to save power leaves the peripherals running for a longer time, and

producing very little energy savings. This realization led to a major change in strategy for laptops — increase the CPU speed (even with a moderate increase in power consumption) in order to finish the computing task as quickly as possible, and then put the CPU and the peripherals into sleep mode.

The goal became to classify changes to processor architecture according to their effect on consumed power and run time. The most desirable changes would decrease power consumption and run time. A small increase in power consumption should be considered if it decreases run time, and a small increase in run time should be considered if it decreases consumed power. Increases in both categories must be avoided.

Slide 7 – 12 Choosing Architectural Changes

Slides 7 – 12 show the method for classifying architectural changes. The CPU is assumed to have an initial state defined by its default power consumption P_0 and its default performance S_0. Each parameter will shift under an architectural change by an amount ΔP and ΔS. The criteria for accepting and rejecting changes are expressed in terms of ΔP and ΔS.

Slide 8 shows certain relationships among power, energy, performance, system voltage, and clock frequency familiar to electrical engineers. They show that lowering the system voltage can reduce CPU power consumption, but this also decreases performance.

Slides 9 – 11 show how these relations can be combined to define a criterion for architectural changes. A relative system voltage produces relative changes in performance, power consumption and total energy consumption according to **voltage scaling**:

$$\frac{\Delta P}{P_0} = 3\frac{\Delta V}{V_0} \qquad \frac{\Delta E}{E_0} = 2\frac{\Delta V}{V_0} \qquad \frac{\Delta S}{S_0} = \frac{\Delta V}{V_0}$$

For example, a 1% drop in voltage will produce a 3% drop in power consumption, a 2% drop in total energy consumption, and a 1% drop in performance. The criterion for architectural changes is that it must be better than voltage scaling. Thus, any change that lowers power by 3% but also lowers performance by more than 1% is not worthwhile — it would be preferable to just lower the voltage. But a change that lowers power by 3% and lowers performance by less than 1% is worthwhile — it cannot be achieved by lowering the voltage.

These criteria are graphed on slide 12.

Slide 13 – 20 Pentium M

The Pentium M was designed at the Intel design center in Haifa as a mobile version of the Pentium 4. Design requirements set by Intel were:

1. Satisfy IA-32 Instruction Set Architecture (ISA) for integer, float, MMX and SSE/SSE2 vector extensions.
2. Implement the microarchitecture of the Pentium 4
3. Save power — maximize battery life in laptop while providing "close-to-desktop" performance (comparable to a Pentium 4 of the same generation)

The design strategy was to introduce architectural changes that meet the voltage scaling criterion. To the surprise of many, the result was a processor that out-performed the Pentium 4 under energy optimization and at maximum performance.

The main enhancements in the Pentium M are:

1. Improvements to branch prediction leads to fewer mispredicted speculative loads, a decrease in the number of instructions per task, and a decrease in run time. This combination increases performance and lowers power.
2. Decreasing the number of micro-ops per instruction by micro-op fusion lowers performance slightly while lowering total energy consumption.
3. Decreasing the number of transistor switches per micro-op lowers performance slightly while lowering total energy consumption.
4. Decreasing energy per transistor switch by building higher quality transistors raises performance and lowers power.

The improvement in branch prediction was achieved with three new features. The first is the loop stream detector (trace cache, discussed in presentation 5). The second is the indirect branch predictor (slide 32 in presentation 4). The third is called a **jump detector**. On the first encounter with a branch instruction, it misses in the branch predictor. It is predicted not taken if it jumps forward (usually and if-conditional) but predicted taken if it jumps backward (generally a loop). The Pentium-M misprediction rate is 20% lower than the rate for the Pentium 4.

The Pentium 4 is a superscalar with dynamic scheduling. IA-32 instructions are decoded to RISC type micro-ops that execute out-of-order for high throughput. But the required resources (decoders, instruction pool registers, registration station) grow with the number of micro-ops, so that the independent processing of micro-ops consumes more power. The Pentium M decodes certain IA-32 instructions to fewer micro-ops (fused micro-ops). Multiple inherently sequential micro-ops are stored as a single fused micro-op with independent sections. The dispatch unit acts on one independent section of a micro-op, executed once for each section. This reduces the number of micro-ops by about 10%.

Similarly, in the Pentium 4 stack operations are decoded as a read or write micro-op on the stack and a separate update of the stack pointer. In the Pentium M, the various changes to the stack pointer are accumulated in a dedicated counter, and the stack pointer is only updated on retirement of a stack operation.

Slides 19 – 20 show comparisons of the Pentium 4 and Pentium M under energy efficient operation (speed close to Pentium 4 and much higher efficiency) and under full speed operation (higher speed and efficiency).

Slide 21 Intel Core Duo

The Intel Core Duo (Intel's second generation dual core CPU) was designed as a chip containing two enhanced Pentium-M cores. The design requirements set by Intel were

1. Similar or better performance than Pentium M on a single thread code
2. Improved performance for multithreaded and multi-processs code
3. Average power consumption of dual core same as Pentium M
4. Meets thermal requirements of target systems

The Core Duo contains independent thermal control per core.

Slide 22 – 25 ACPI

Even with energy efficient processors, the most effective way to save power is to deactivate system components when they are not required. The **Advanced Configuration and Power Interface** (ACPI) is an industry standard for the OS interface to processor configuration, developed by HP, Intel, Microsoft, Phoenix, Toshiba. ACPI defines operating states for the global system and devices.

The global (system) states are:

G0 (working), G1 (sleeping), G2 (software off), G3 (mechanical off)

and the device states are:

D0 (on), D1/D2 (device dependent) , D3 (off)

Specific CPU power states are

C0 (active), C1 (halt), C2 (stop clock), C3(sleep), C4 (deep sleep), ...

The OS controls the CPU power state to save power. When the scheduler detects an empty task queue, it transitions the processor into an idle state with a BIOS instruction sequence. The processor remains idle until it receives an interrupt that indicates a new task to be executed. The OS returns the processor to the C0 state. The use of sleep states for the Core Duo is shown in slide 23.

The OS is given range of sleep options. One possible idle management transition diagram is shown in slide 24. The definitions of the sleep states for the Core Duo is given in slide 25.