Some Power Observations
Food for Thought

Ronny Ronen

Haifa Microprocessor Research Lab

ronny.ronen@intel.com

Education – Power & Energy considerations

Thermal – cannot afford exceeding operating temperature
Higher power ➔ Higher temperature

- **Max Power**
  - » Theoretical power value a processor may require
- **High Power test (~90% of Max)**
  - » Maximum known achievable required power
- **Thermal Design Power – TDP (~75% of Max)**
  - » High enough point – not exceeded by most practical application
  - » Processors are designed for that power. Throttled when exceeds
- **Power Envelop**
  - » A power value allowed for certain implementation
  - » Envelop depends on system (cooling), and processor (area, distribution).
- **"Measured" at peak time**

Energy – important mainly for battery life.

- **Measured over time**

Di/Dt – changes in current due to varying power demand

- **"Measured" at peak time**

Education (2)

Power = αCV^2f (Activity x capacitance x voltage^2 x frequency)

- **Affected by new process technology**
- **Affected by new μarch**

At certain range f = kV

- **In other words P is proportional to V^3 !**
- **Lower freq ➔ lower voltage ➔ lower power (20% freq = 50% power!)**

Thermal

- **Max Tj (Junction temperature) should not exceed 100C-110C**
- **Tj is proportional to the power dissipated by the processor**
- **$T_J = T_A + \Theta_{JA} \times P$ (e.g., 100 = 50 + 2.5* 20)**

For most segments, battery life (=energy) is more of a system issue

- **CPU is <10% of the overall energy.**

Efficiency – what is it?

Source:
Power-aware microarchitecture
Brroks, Bose et. Al
IBM IEEE-MICRO 11-12/2000

1. Perf/Power (➔ Energy)?
2. Perf^2/Power (➔ Energy*Delay)?
3. Perf^3/Power (➔ Energy*Delay^2)?

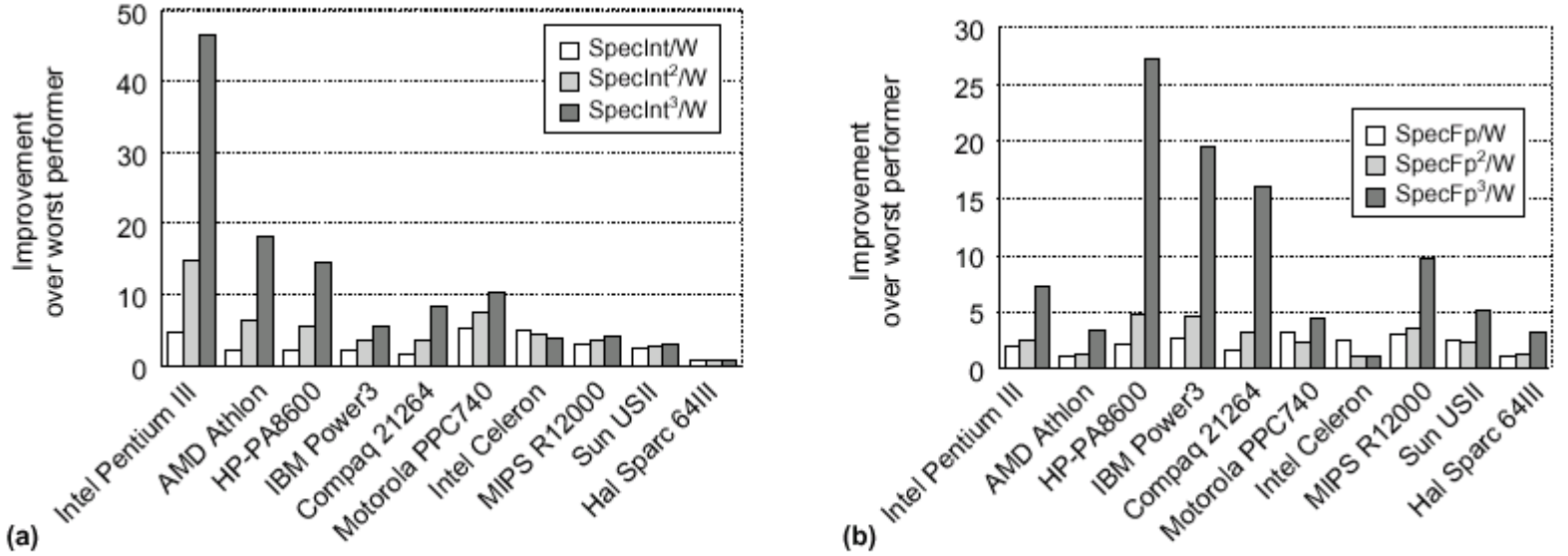


Figure 1. Performance-power efficiencies across commercial processor products. The newer processors appear on the left.

1. Appropriate if time is not a factor – battery life
2. Appropriate when time is factor
3. Appropriate as tradeoff for fixed power envelop

Process Technology - Ideal Scenarios...

Ideal "Shrink"

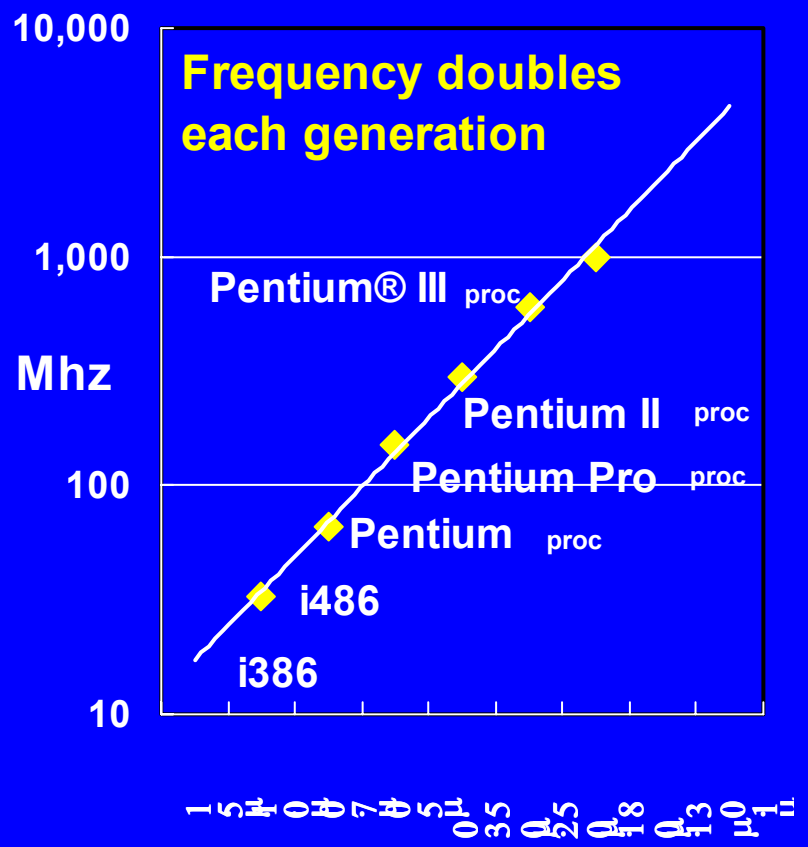
- *Same μarch*
- 1X #Xistors
- 0.5X size
- 1.5X frequency
- 0.5X power
- 1.5X performance
- 1X power density

Ideal New μarch

- *Same die size*
- 2X #Xistors
- 1X size
- 1.5X frequency
- 1X power
- 3X performance
- 1X power density

Looks good. Isn't it?

Real Trends and expectations



With Each Process Generation:

- Frequency increased by ~2X (not 1.5X)
- Vcc will scale by only ~0.8 (not 0.7)
- Active power will scale by ~0.9 (not 0.5)
- Leakage power will make it even worse, and
- Active power density will increase by ~30-80% (not stay constant)…

Performance Efficiency of µarchitectures

Tech	Old µArch	mm (linear)	New µArch	mm (linear)	Area
1.0µ	i386C	6.5	i486	11.5	3.1
0.7µ	i486C	9.5	Pentium® proc	17	3.2
0.5µ	Pentium® proc	12.2	Pentium Pro®	17.3	2.1
0.18µ	Pentium III® proc	10.3	Pentium® 4 proc	?	2--3

Implications: (in the same technology)

1. New µArch ~ 2-3X die area of the last µArch
2. Provides 1.5-1.7X integer performance of the last µArch

We are on the Wrong Side of a Square Law

Power & Performance efficiency: Qualitative look

Performance ("MIPS"): IPC * Frequency (IPC = Instruction Per Cycle)

Energy spent in processing an instruction: W_i

- Increases with the complexity of the processor.
  E.g., Out Of Order processor consumes more energy per instruction than an in-order processor (renaming, scheduling, retiring…)

Ratio of Useful to Total number of processed instructions: η

- Total Instruction Per Cycle (IPC) including speculated instructions: IPC/η.

Energy/second (=power) proportional to: (IPC/η)*Frequency*W_i

- Proportional to the amount of processed instructions per second and the amount of work consumed per instruction.

Energy efficiency (for a given instruction stream), measured in MIPS/Watt, proportional to: η/W_i

- This value deteriorates as speculation increases and complexity grows.

Modern processors are more performant
But are less efficient!

Five Non Linear Functions*

1. Microarchitecture vs performance

(Keeping process technology constant). Increasing single thread general purpose performance by 2X requires 4X the die area

2. Voltage vs. Power.

This is actually a cube law. Reducing voltage by 1/3, reduces frequency by 1/3, and power by 19/27th (i.e. 1 - (2/3)**3).

3. Transistor sizing vs. frequency.

Power is proportional to width of transistor (which determines speed of transistor). For example, 10% reduction in width produces 10% more power, but since it enables 10% more frequency and frequency is proportional to power. Thus, another square law.

4. Leakage Power vs. Junction temperature.

As junction temperature increases, leakage power increases exponentially.e.g. a reduction from 100 degrees C in Tj to 75 degrees will cut leakage power in half (approximately). Note that a lower Tj also results in somewhat higher frequency.

5. Voltage Threshold (Vt) vs. Leakage power.

Going from a high Vt transistor to a low Vt transistor will boost transistor frequency by 10-15%, but increase leakage by 10X. (And freq increase is directly proportional to % usage of low Vt, and exponentially depends on leakages of hi & lo Vt.)

*** Originated by Fred Pollack and Shekhar Borkar**

Working without Power Limits

"Traditional Desktop"

Perf = IPC * Freq

$P = \alpha C V^2 f$

V=const

Assumptions:

IO/OOO = same #gate delays per stage

Superpipline: 1.2X Cap, 1.9X freq, 0.85X IPC

Out of Order: 2X Cap, 1X freq, 1.5X IPC *(1.3X IPC)*

Type	C	V	Freq (@ same V)	IPC	Power	Perf	Perf/ Power
I In-Order	1X	1X	1X	1X	1P	1X	1X
O Out of Order	2X	1X	1X	1.5X *1.3X*	2P	1.5X *1.3X*	0.75X *0.65X*
Isp In-Order, Super Scalar	1.2X	1X	1.9X	0.85X	2.3P	1.6X	0.7X
Osp Out of Order Super Scalar	2.4X	1X	1.9X	1.3X *1.1X*	4.6P	2.4X *2.1X*	0.53X *0.46X*

Modern processors are more performant - But are less efficient!

Working within Power Envelope (same V)

"Ultra low Power"

Perf = IPC * Freq

P = αCV^2f

V=const, P=const

Assumptions:

IO/OOO = same #gate delays per stage

Superpipline: 1.2X Cap, vary. freq, 0.85X IPC

Out of Order: 2X Cap, vary. freq, 1.5X IPC *(1.3X IPC)*

Type	C	V	Freq (@ same V)	IPC	Power	Perf	Perf/ Power
I In-Order	1X	1X	1X	1X	1P	1X	1X
O Out of Order	2X	1X	0.5X	1.5X *1.3X*	1P	0.75X *0.65X*	0.75X *0.65X*
Isp In-Order, Super Scalar	1.2X	1X	0.83X	0.85X	1P	0.7X	0.7X
Osp Out of Order Super Scalar	2.4X	1X	0.42X	1.3X *1.1X*	1P	0.53X *0.46X*	0.53X *0.46X*

Reduced Performance via Frequency Scaling => Same Efficiency Picture

Working within Power Envelope (V/F scale) *"Mobile"*

Perf = IPC * Freq

P = αCV^2f

P=const

Assumptions:

IO/OOO = V=kf, same #gate delays per stage

Superpipline: V=½kf, 1.2X Cap, vary. freq, 0.85X IPC

Out of Order: 2X Cap, vary. freq, 1.5X IPC *(1.3X IPC)*

Type	C	V	Freq	IPC	Power	Perf	Perf/ Power
I In-Order	1X	1X	1X	1X	1P	1X	1X
O Out of Order	2X	0.8X	0.8X	1.5X *1.3X*	1P	1.2X *1.05X*	1.2X *1.05X*
Isp In-Order, Super Scalar	1.2X	0.75X	1.5X	0.85X	1P	1.27X	1.27X
Osp Out of Order Super Scalar	2.4X	0.6X	1.2X	1.3X *1.1X*	1P	1.5X *1.3X*	1.5X *1.3X*

High Performance may Win Power => Better Efficiency

Working within Power Envelope (Same C)

"Web Server"

Perf = IPC * Freq

$P = \alpha C V^2 f$

P, C=const, Via CMP

Assumptions:

IO/OOO = V=kf, same #gate delays per stage

Superpipline: V=½kf, 1.2X Cap, vary. freq, 0.85X IPC

Out of Order: 2X Cap, vary. freq, 1.5X IPC *(1.3X IPC)*

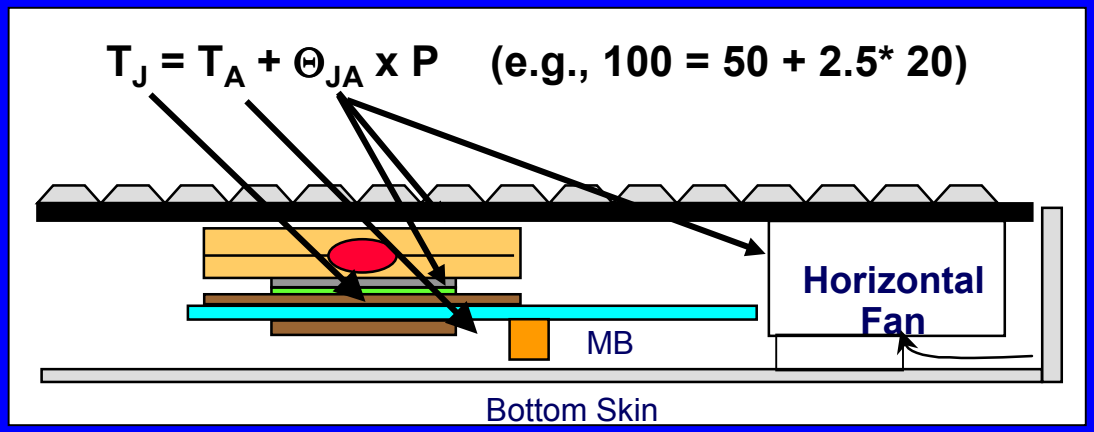
Type	C	V	Freq	IPC	Power	Perf	Perf/ Power
I In-Order	2*1X	0.8X	0.8X	2*1X	2*0.5P	2*0.8X	1.6X
O Out of Order	2X	0.8X	0.8X	1.5X *1.3X*	1P	1.2X *1.05X*	1.2X *1.05X*
Isp In-Order, Super Scalar	2*1.2X	0.6X	1.2X	2*0.85X	2*0.5P	2*1.0X	2.0X
Osp Out of Order Super Scalar	2.4X	0.6X	1.2X	1.3X *1.1X*	1P	1.5X *1.3X*	1.5X *1.3X*

Are Two Pipers Better than one F16?

The Mythical Power Envelop

CPU power envelop varies

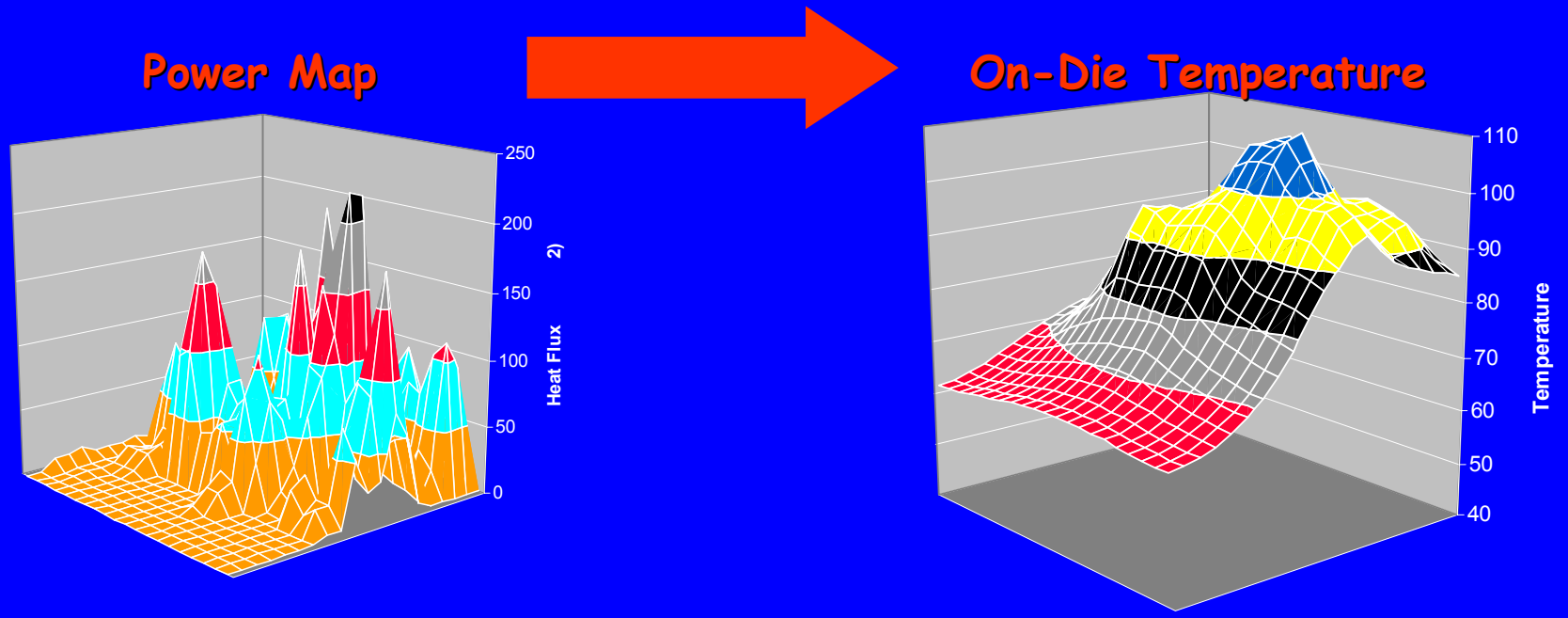
- Bigger systems can cool better and dissipate more power
- Average power density matters



Some observations:

- Interaction among hot spots reduces w/ distance
- Silicon is conductive – to a point. Adding low-activity area at distance do not absorbs much.

Power Density



Varying power density ➔ "Hot spots" - temperature is not uniform

- Hot spots limit power and cause CMOS logic to slow down

A more uniform power map would flatten the on-die temperature

➔ Can dissipate more power

What's the Lesson?

"Things Look Different from the Other Side of the Mirror"

Goal: Maximum performance for a given power envelope

Strategy: Optimum is not obvious to us

- *In-order vs Out-of-Order*
- *Narrow/Wide machine*

But some "back to basics" always work:

- *Save locally wherever you can (clock gating)*
- *Smartly trade performance – power (serial caches)*
- *Eliminate unnecessary work (e.g., use decoded Icache)*
- *Avoid excess speculation (Slow/stop/switch on low conf branches)*
- *Look for energy saving, even at cost of higher power (Better branch predictors)*

What Else?

And the power envelop is not fixed either…

- *And μarch can help here as well!*

And we did not even considered yet:

- *leakage, di/dt, wire delays*

Your Mission:
"Squeeze more performance, consume less power"