

# Generalization of Pollack's rule and alternative power equation

JUAN RAMÓN GONZÁLEZ ÁLVAREZ

C:/Simancas 26, E-36208 (Bouzas) Vigo, Pontevedra, Spain

<http://juanrga.com>

2014, Dec 01, 21:00

---

## Abstract

After showing that only one of the different versions of POLLACK's rule found on the literature agrees with the experimental behavior of a CPU running at stock frequency versus the same CPU overclocked, we introduce a formal simplified model of a CPU and derive a generalized POLLACK's rule also valid for multithread architectures, caches, clusters of processors, and other computational devices described by this model. A companion equation for power consumption is also proposed.

---

## 1 Introduction

POLLACK's rule [1] is often used to model the performance of multi and manycores. However, different versions of the rule are invoked in the literature. In the next section, we show that only one of the versions agrees with the experimental behavior of a CPU running at stock frequency versus the same CPU overclocked. Next, we introduce a formal simplified model of a CPU and derive a generalized POLLACK's rule also valid for multithread architectures. Finally, we show that the usual power equations utilized in the literature don't agree with the behavior of a CPU running at stock frequency versus the same CPU overclocked, and propose a new power equation that yields the expected scaling with frequency.

## 2 Which version of the rule?

Certain authors [2–6] affirm that «*processor performance is proportional to the square-root of its area*», whereas others state that only the instructions per cycle (IPC) are proportional to the square root of its area [7–10]. RAN GINOSAR also uses a variant where the frequency is proportional to the square root of area [6]. Thus, we have to evaluate three different possibilities for the relation between performance  $R$ , instructions per cycle  $I$ , frequency  $f$ , and area  $A$ :

$$R \propto \sqrt{A}, \quad (1)$$

$$I \propto \sqrt{A}, \quad (2)$$

and

$$f \propto \sqrt{A}. \quad (3)$$

*A priori*, we could use the relation between performance, IPC, and frequency

$$R = If \quad (4)$$

to conclude that (2) and (3) are special cases from (1) when frequency and IPC are held constant, respectively. This is too naive and shown to be wrong *a posteriori*.

The simplest way to test the three versions of the POLLACK's rule is comparing a CPU running at stock frequency (e.g., 3.5 GHz) against the same CPU but overclocked (e.g., 4.1 GHz). In both cases the CPU has the same area but different performance and frequency, which eliminates (1) and (3) and leaves (2) as the correct version of the rule [7–10].

However, rule (2) does not work when comparing two CPUs running at same frequency but built on different processor nodes (e.g. 28 nm vs 20 nm). The CPU built on the newest node has less area but identical performance. A generalization of (2) is found in the section 3.

### 3 Generalized Pollack's rule

We begin by constructing a simplified formal model of a CPU. This model consists of two interrelated sections [Figure 1]. The first section processes a serial stream of instructions and extracts the available instruction-level-parallelism (ILP). A second section executes the identified instructions in parallel.

The diagram (i) represents a CPU that can identify and execute up to two instructions in parallel, whereas the diagram (ii) represents a CPU that can identify and execute up to four instructions in parallel from the same instruction stream at same frequency. However, whereas the first CPU occupies an area  $A$ , the second occupies an area roughly  $4A$ ; i.e., doubling the performance requires quadrupling the area:  $2R \Leftrightarrow 4A$ . This nonlinear increase in the area is a consequence of the strong coupling between both sections, which have to be scaled in orthogonal ways.

This discussion assumes the same process node for all the CPUs. Now, a die shrink of a given CPU would maintain the performance –if frequencies and everything else remains the same–, whereas the area would be reduced to one half. Thus we finally obtain the generalized Pollack's rule

$$R_1 \approx \frac{f}{\lambda} \sqrt{A_1}, \quad (5)$$

with  $\lambda$  a node parameter with length dimensions and the subindex 1 denoting a single instruction stream. For instance, the performance ratio of two CPUs one built on 28 nm and other on 20 nm and both clocked at same frequency will be

$$\frac{R_1(28\text{nm})}{R_1(20\text{nm})} \approx \frac{20 \sqrt{A_1(28\text{nm})}}{28 \sqrt{A_1(20\text{nm})}} = 1. \quad (6)$$

All of this is valid for a single instruction stream. The diagram (iii) represents a CPU that can identify and execute up to four instructions in parallel from two instruction streams. Since the parallelism between the set of instructions 1–2 and the set 3–4 was identified by the compiler, which scheduled them to different streams, the CPU requires simpler logic to extract the remaining ILP and the area is only  $2A$ . We can finally obtain the generalized Pollack's rule for  $N$ -streams executed on a CPU with total area  $A_N$

$$R_N \approx \frac{Nf}{\lambda} \sqrt{A_1} = \frac{f}{\lambda \sqrt{A_1}} A_N. \quad (7)$$

This generalized rule shows that we can increase the performance of a CPU by increasing frequency  $f$ , by extracting more ILP ( $A_1 \uparrow$ ), and/or by exploiting more TLP ( $N \uparrow$ ).

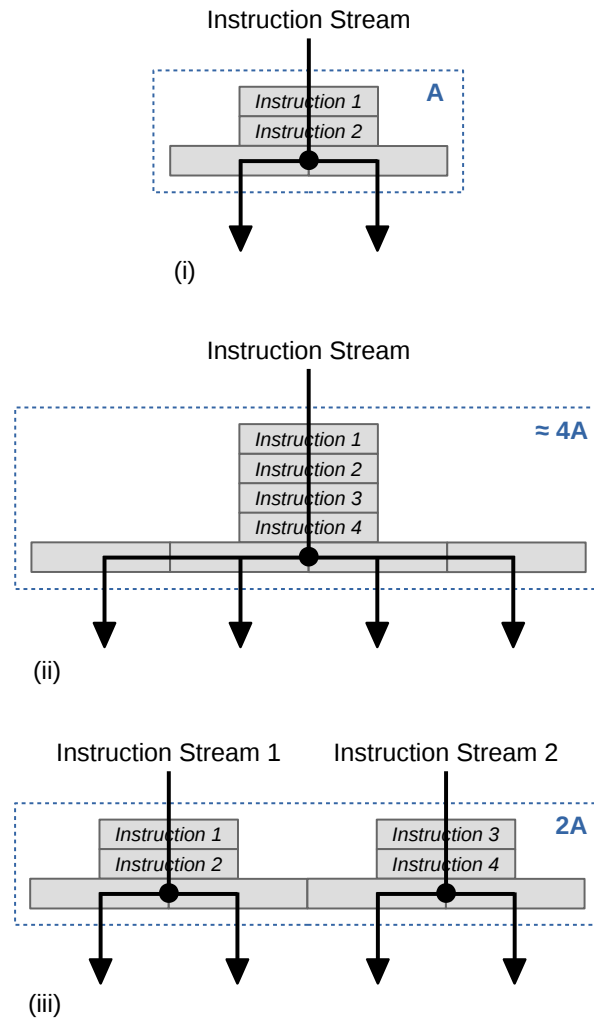


Figure 1: Simplified formal model of a CPU

This generalized rule has been derived from a simplified model of a CPU but is also valid for any other computational device involving a similar two-section coupled structure. For instance, the same square root performance law is valid for a cache [7], whose miss rate  $M$  is given by

$$M \propto \frac{1}{\sqrt{A_{\text{cache}}}}, \quad (8)$$

and is also valid for a cluster of processors or cores working together on a single stream of instructions via speculative execution. Applying the square law (7) to a cluster of 64 cores –each with area  $A_1$ – results on a speed up

$$\frac{R_1(64\text{core})}{R_1(1\text{core})} \approx \frac{\sqrt{64A_1}}{\sqrt{A_1}} = 8 \quad (9)$$

which agrees very well with measurements [11].

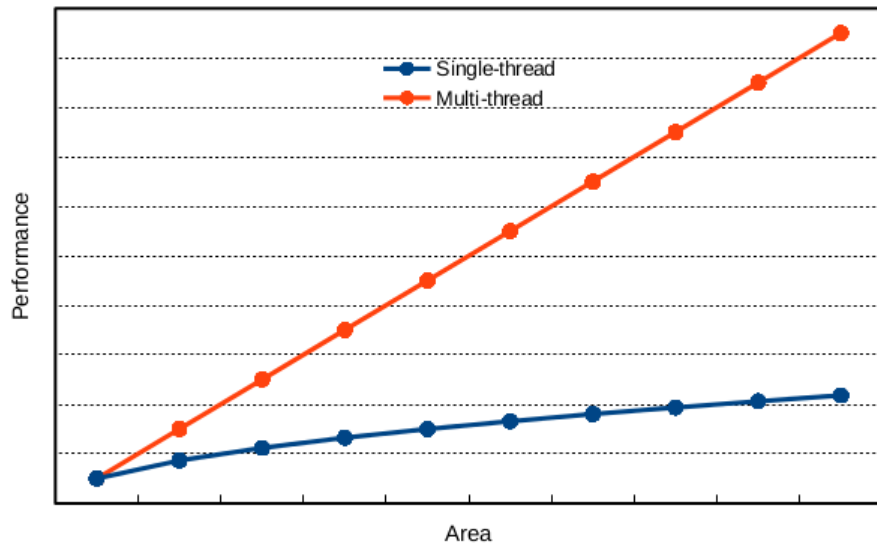


Figure 2: Single-thread and multi-thread performance as function of CPU area

## 4 Power equation

In the literature [1, 4, 6], the equations (1) and (3) are complemented with the next equation for power consumption  $P$

$$P \propto Af. \quad (10)$$

This equation is also suspicious, because predicts a linear increase in power consumption for an overclocked CPU compared to the same CPU at stock clocks, due to both CPUs having the same area.

Using (4) and combining the equations (1) and (10), the reference [4] derives

$$P \propto I^2 f^3, \quad (11)$$

whereas the reference [6] combines the equations (3) and (10) to obtain

$$P = A\sqrt{A} \propto f^3. \quad (12)$$

This cubic dependence on the frequency is about right, but it has been derived from combining two suspicious equations, whose deficiencies regarding the areas  $A$  luckily self-cancel in this case. In what follows we will derive an alternative power equation compatible with the generalized Pollack's rule (5).

We start with the next equation for the power consumption of an electric device

$$P = \beta CV^2 f, \quad (13)$$

here  $\beta$  is an utilization parameter,  $C$  is capacity, and  $V$  the working voltage of the electric circuit. The utilization parameter is proportional to the area of the device, and using  $V \approx f$ , we obtain

$$P \propto Af^3. \quad (14)$$

Unlike (10), this new equation predicts a cubic increase in power consumption for an overclocked CPU compared to the same CPU at stock clocks. Combining this new equation with (5), we can obtain (11)

$$P \propto Af^3 \propto I^2 f^3. \quad (15)$$

## References and notes

- [1] New microarchitecture challenges in the coming generations of CMOS process technologies 1999: *MICRO Keynote*. POLLACK, FRED.
- [2] Thousand Core Chips — A Technology Perspective 2007: *Proc. Design Automation Conference*, 746–749. BORKAR, SHEKHAR.
- [3] Importance of Single-Core Performance in the Multicore Era 2012: *Proceedings of the Thirty-Fifth Australasian Computer Science Conference*. SATO, TOSHINORI; MORI, HIDEKI; YANO, RIKIYA; HAYASHIDA, TAKANORI.
- [4] Mathematical modeling of many-cores 2013: *Technion*. GINOSAR, RAN.
- [5] The future of microprocessors 2011: *Communications of the ACM*, 54(5), 67–77. BORKAR, SHEKHAR; CHIEN, ANDREW A.
- [6] Many-cores: Supercomputer-on-chip How many? And how? (how not to?) 2009: *Technion*. GINOSAR, RAN.
- [7] Beyond Amdahl's Law: An Objective Function That Links Multiprocessor Performance Gains To Delay and Energy 2012: *IEEE Transactions on Computers*, 61(8), 1110–1126. CASSIDY, ANDREW S.; ANDREOU, ANDREAS G.
- [8] Fast and Accurate Techniques for Early Design Space Exploration and Dynamic Thermal Management of Multi-core Processors 2008: *ProQuest; Ann Arbor*. RAVISHANKAR, RAO.
- [9] High-performance and hardware-aware computing 2012: *KIT Scientific Publishing; Karlsruhe*. BUCHTY, RAINER; WEISS, JAN-PHILIPP (EDS.).
- [10] Microprocessors for the New Millennium – Challenges, Opportunities and New Frontiers 2001: *Digest of Technical Papers. ISSCC. IEEE International*, 22–25. GELSINGER, P. P.
- [11] Programming Many-Core Chips 2011: *Springer; New York*. VAJDA, ANDRÁS.