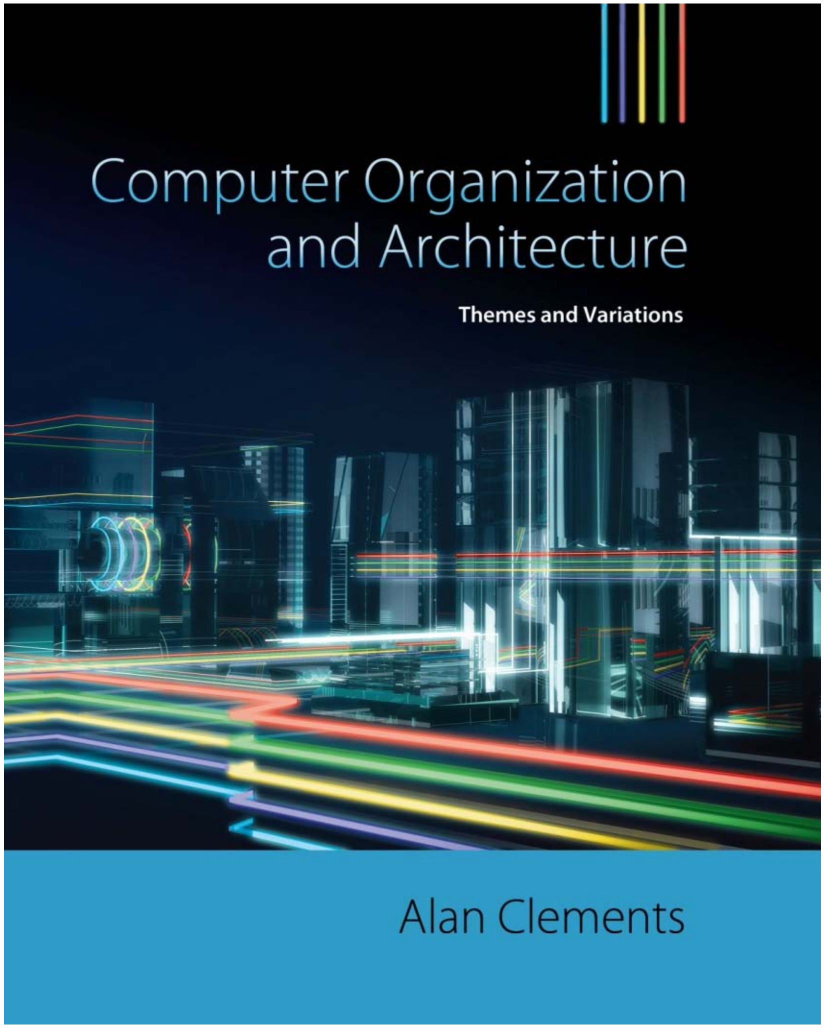
CHAPTER 6

Performance—Meaning and Metrics



CENGAGE Learning

Performance

- Performance is the most *interesting* and yet *problematic* topic in computer architecture.

    - It is ***interesting*** because few people seem to agree on
        - how to ***measure*** performance, and/or
        - how to ***interpret*** the results.

    - It is ***problematic*** because
        - its use can be ***meaningless*** in everyday situation
        - it may specify an aspect of a computer that might be of ***the least importance*** in many applications.

Performance

- According to a dictionary definition,
  "*performance*" is "*how successful someone is*".
- This is a meaningless definition.
  It is similar to say,
  "*performance*" is "*how someone perform*"

Performance

- Several groups of people are interested in computer performance (*each group looks at it from different prospective*)
  - *Corporate users*:
    - usually buy massive number of computers
    - need to obtain the best value for their money
    - require objective *criteria to compare* various systems
  - *Knowledgeable elite users*:
    - look for the best computer for game playing or multimedia applications
    - need to know *how well* various computers will handle their programs
  - *Chip manufactures*:
    - need an objective ways to allow them to *locate bottlenecks*; hence improving their future innovations
  - *Software and compiler developers*:
    - need ways to allow them to *understand the relationship between hardware and software* in order to create the most efficient codes
  - *Academics who teach computer design and carry out related research:*
    - need to have an objective ways of *quantitatively evaluate new designs*
  - *Marketing and sales teams*
    - need to have a *single number* that is higher than their competitors'
  - *Shareholders*
    - need to see the value of their *shares and dividend increase*

What is "Performance"?

- Do you agree that a computer performs better than another if it is faster?
  - Well, this might not be the case, as
    - performance *should not* be a single variable function.

- The *performance* depends on various weighted factors
- Factors include:
  - Cost
  - Speed
  - Memory capacity/access time
  - Weight
  - Connectivity
  - Power consumption
  - …
- Factors related to hard drives include:
  - Cost
  - Size
  - RPM (7200 rpm vs 5400 rpm)
  - Number of plates
  - Transfer rate
  - Weight
  - Power consumption
  - MTBF
  - …

Solid state hard drives dramatically changed all these factors.

What is "Performance"?

- It is easy to state what the designer should do to optimize the performance.
- In real world, the designer does not normally have a green light to create an entirely new system.
  - A computer, as a part of a manufacture's product line, has to be *backward combatable* with the codes written for previous products in the same family.
    - This means that the computer *must be able to run existing software*, which the designer has no control over it.
  - The processor has to *operate with other components*, e.g., memory and buses, that the designer can not optimize to suite the new CPU.

Progress and Computer Technology

- In a static world, there would be no need to worry about performance.
  - There would be little change from day to day to notice it.

- Technology world is not static.
  It is developing at an unbelievable rate
  - to improve the performance of existing systems, and
  - to develop new systems

- Before talking about performance, let us briefly talk about the march of progress in computer technology.

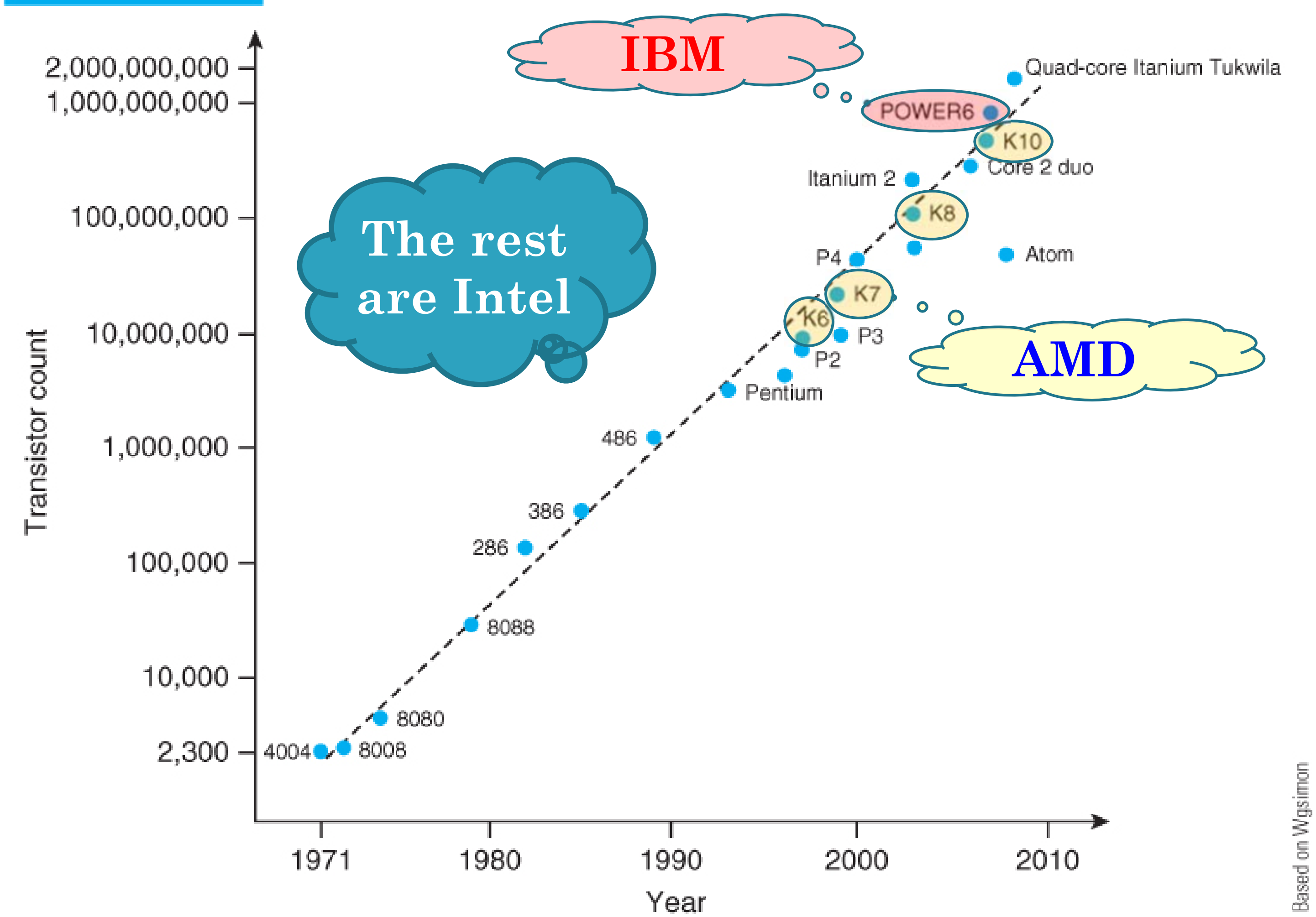
Moore's Law

- More than 50 years ago, *Moore's law* suggested an *exponential* increase in the maximum number of components on a chip with time.
  - In 1965, he predicted a doubling of the maximum number each year
  - In 1975, he revised it to doubling of the maximum number each two years
  - Year after year, the rate of increase has been empirically adjusted to doubling of the maximum number each 18 months
  - This trend that was observed by *Gordon Moore* in the 1960's has continued largely unbroken until 2010 (currently, it may be doubled in a longer period of time—3 years or so)
- FYI: If a baby in 1960 was crawling in about *1 meter/hour* (*average human speed is 5 km/hour*) and he/she increased his/her speed in accordance with Moore's law (*double the speed each two years*),
  - in 2018, he/she would be moving at more than *1,000,000 km/hour* i.e., about *298 km/second*;
- Moore's Law has *no precise formulation* or *basis in physics or natural laws*.
- In fact, *Moore's law is not a law!!*
  It is based on an *observation of the progress* of semiconductor technology.
- *Semiconductor companies, e.g., Intel, apply this law in their production line.*

Semiconductor Progress

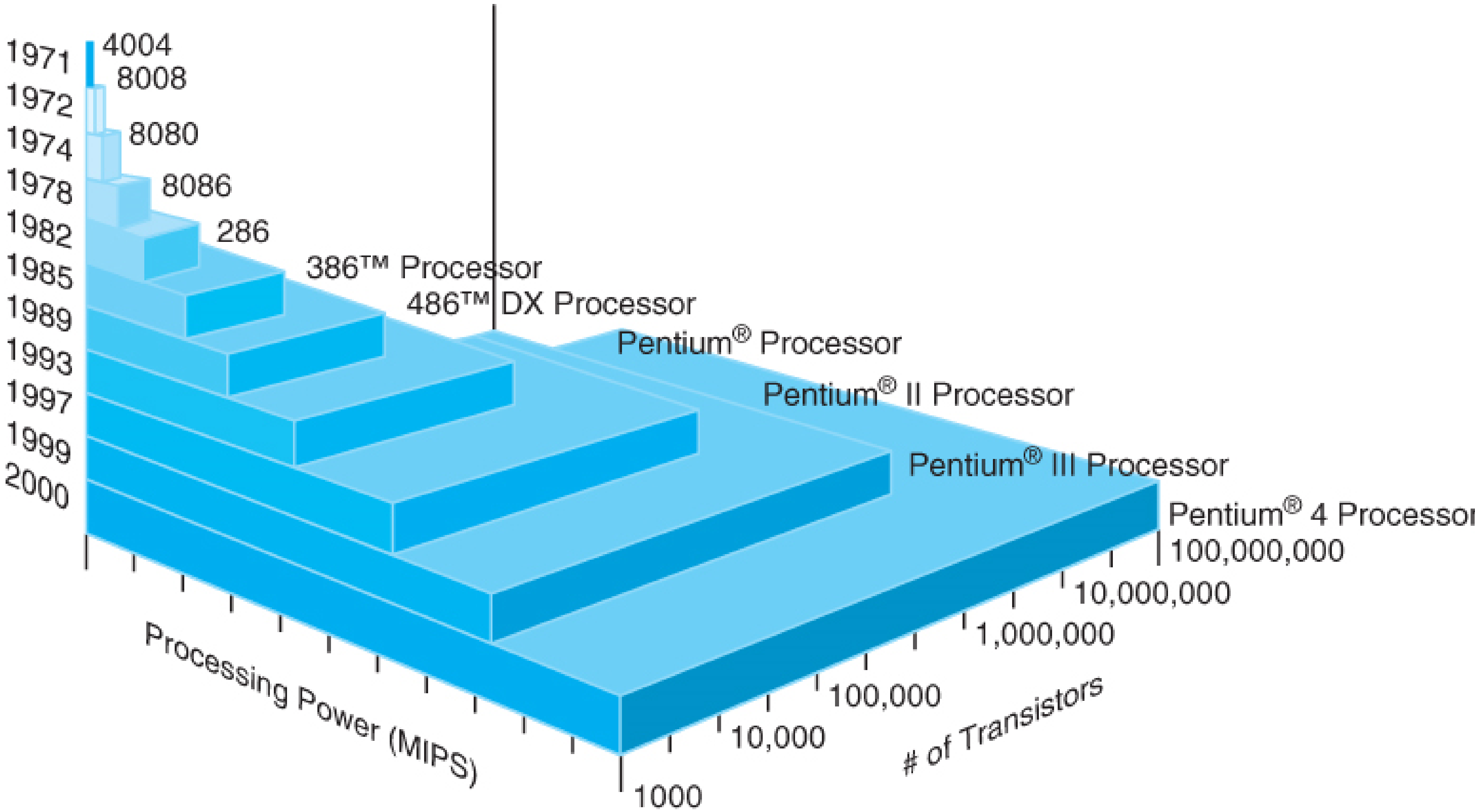
- Figure 6.2 displays the number of transistors per microprocessor chip as a function of time. In barely five decades the chip density has gone from the region of *two thousand* to *two billion* devices.

FIGURE 6.2 The number of devices on a circuit



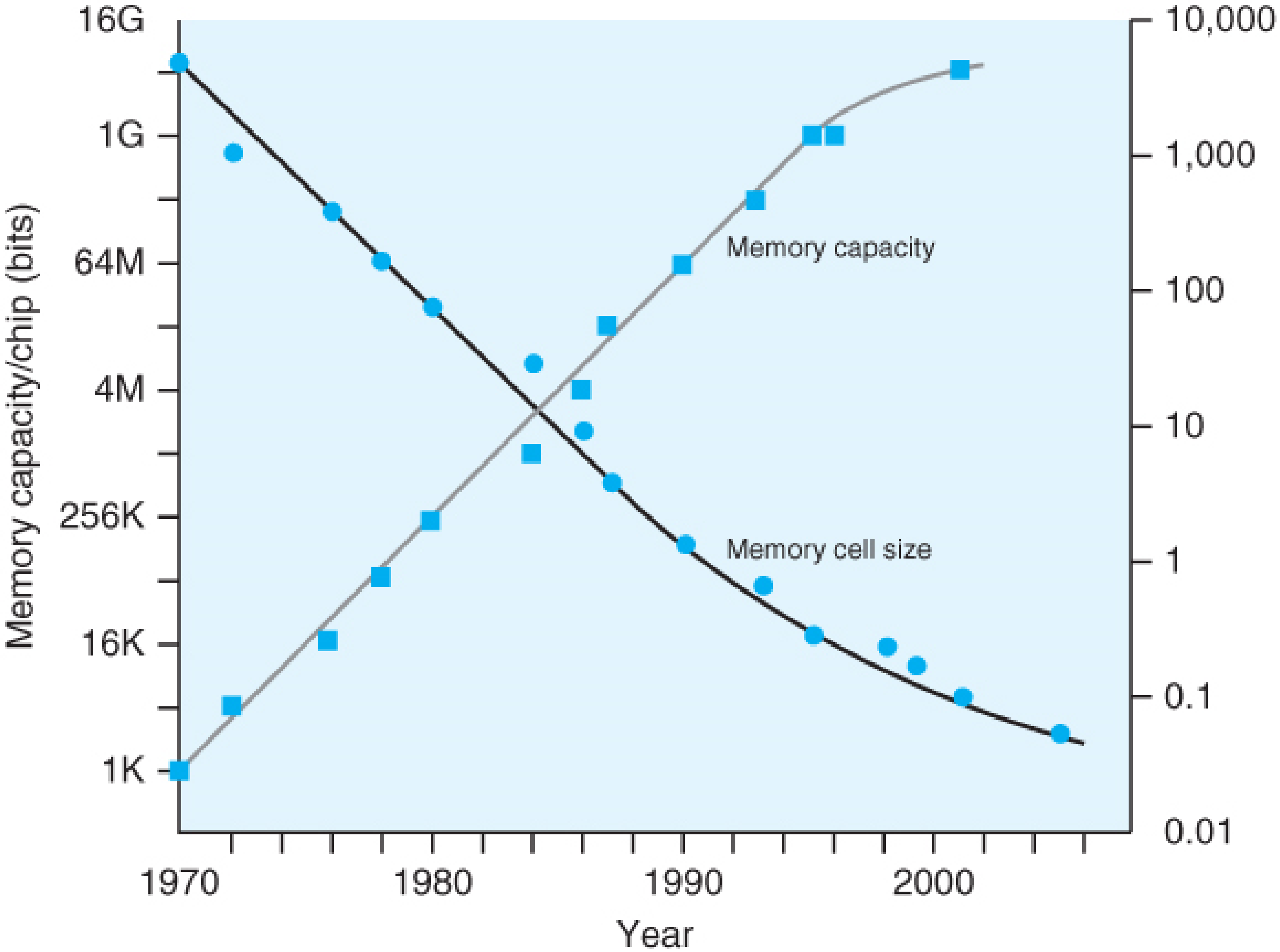
Semiconductor Progress

FIGURE 6.4 **3D illustration of microprocessor progress.** "De-Mystifying Software Performance Optimization," Intel® Software Network, http://software.intel.com/en-us/articles/de-mystifying-software-performance-optimization/. Reprinted with permission of Intel Corporation.



Semiconductor Progress

FIGURE 6.5 The growth of memory capacity



Performance and Design

- The performance of a computer not only depends on its CPU unit, but also on other components of the computer system, including, *but not limited to*
  - ***cache memory*** (fast memory) that holds frequently used data,
  - ***main memory*** that holds programs and data,
  - ***buses*** that allow sub-modules to communicate with each other, and
  - ***secondary storage***, e.g., hard disks and optical drives.

Such topics to be covered in CS3350

It is not enough to see a sticker saying

- *Every one of these systems contributes to the overall performance of the computer* and *you cannot maximize the performance of a computer by improving only one of its components*.

Performance and Design

- A list of *criteria* by which we *can judge the metrics of computer performance* include,
  1. *reliability*,
  2. *linearity*,
  3. *repeatability*,
  4. *ease of measurement*,
  5. *consistency*, and
  6. *independency* .

Performance and Design

- Performance metrics should be *reliable* and correctly indicate whether one computer is better than another.
  You could also call this property *monotonicity*
  - an increase in the value of a metric should indicate a better computer.

Performance and Design

- The *linearity* criterion suggests that a metric should be linear
  - Increasing the performance of a computer by a fraction x should be reflected by an increase of fraction x in the metric.
  - If computer A has a metric of, say 200, and it is twice as fast as computer B, then computer B's metric should be 100.

Performance and Design

- A good metric should be *repeatable* and always yield the same result under the same conditions.

- Suppose you are carrying out a test that requires the reading of data from a disk.

  - In the first run, the data may be about to fall under the read head at the time it is required and, therefore, the data will be immediately ready.

  - In the next run of the same test, the data may have just passed under the read head and the system will have to wait for a complete rotation to access the data.

  - In a third run, the data might be cached in RAM and the system will entirely bypass the disk's hardware.

Consequently, we can have three runs of a test yielding three different metrics using the same data.

Performance and Design

- The *ease of measurement* criterion is self-explanatory.
  - If it is difficult to measure a performance criterion, then *only few* users are likely to be able to correctly and accurately make that measurement.
  - Moreover, if a metric is difficult to measure, an independent tester will have great difficulty in confirming it.

Performance and Design

- A metric is *consistent* if it is precisely defined and can be applied across different systems. Perhaps this should be called *universality* to avoid confusion with *repeatability*.

  - Consistency can be difficult to achieve if the metric measures a feature of a specific processor and that feature is not constant across all platforms.

Performance and Design

- Finally, a good metric should be *independent of commercial influences*.
  - If computer manufacturers to define performance metrics, they might be tempted to select criterion that
    - shows their processor in a better condition than their competitors' processors.
    - emphasizes a specific feature of their processor that is lacking in competitors' devices, even though this feature may have little or no overall effect on system performance in practice.