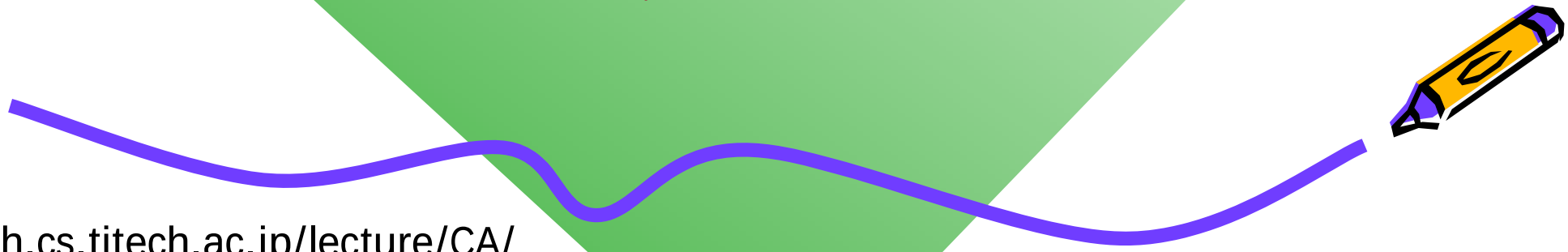
2018年度（平成30年度）版

Ver. 2018-10-01a

Course number: CSC.T363

コンピュータアーキテクチャ
Computer Architecture

2. コンピュータの性能と消費電力の動向
Trends in Performance and Power



www.arch.cs.titech.ac.jp/lecture/CA/
Room No.W321
Tue 13:20-16:20, Fri 13:20-14:50

吉瀬 謙二　情報工学系
Kenji Kise, Department of Computer Science
kise _at_ c.titech.ac.jp

Growth in clock rate of microprocessors

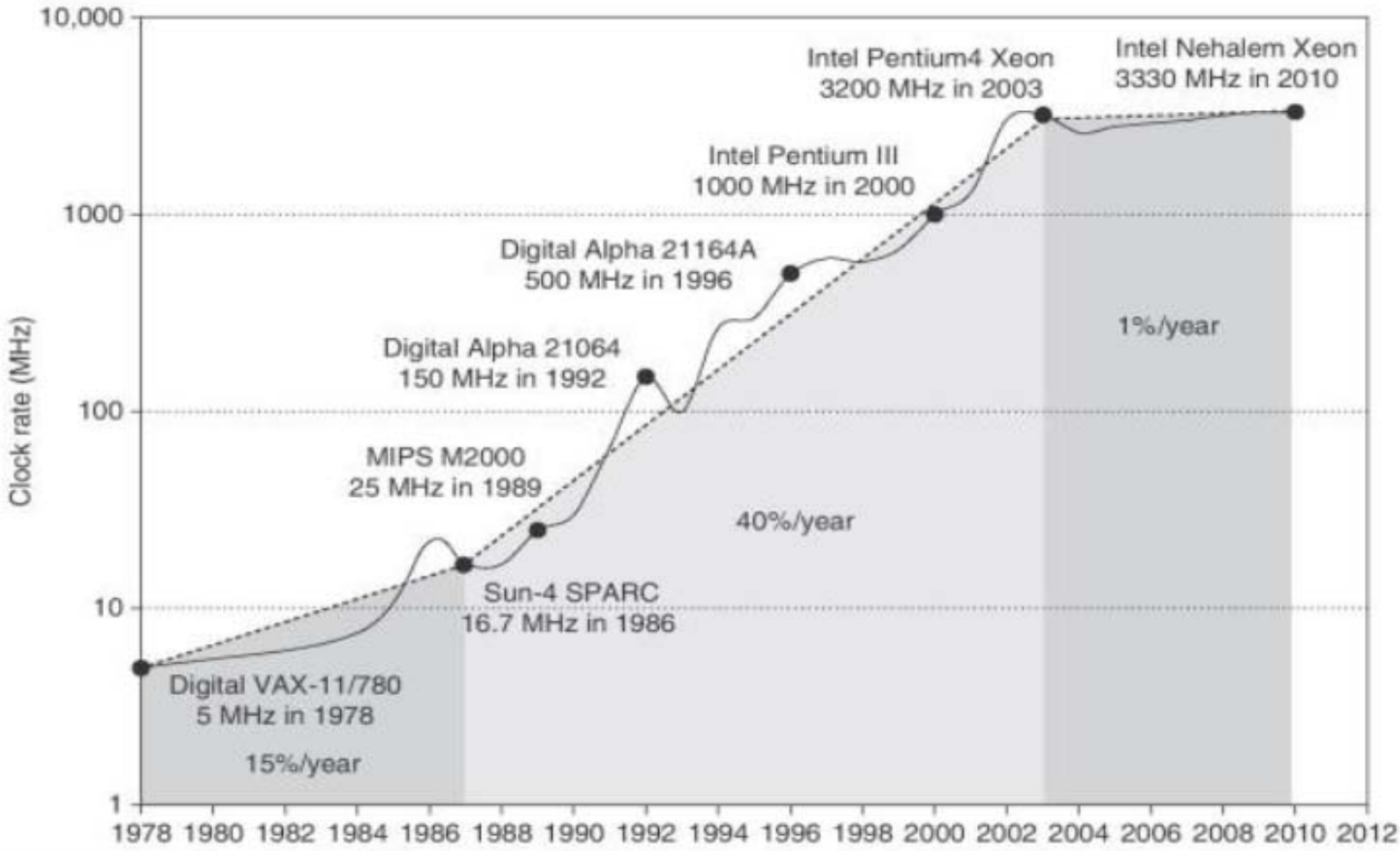
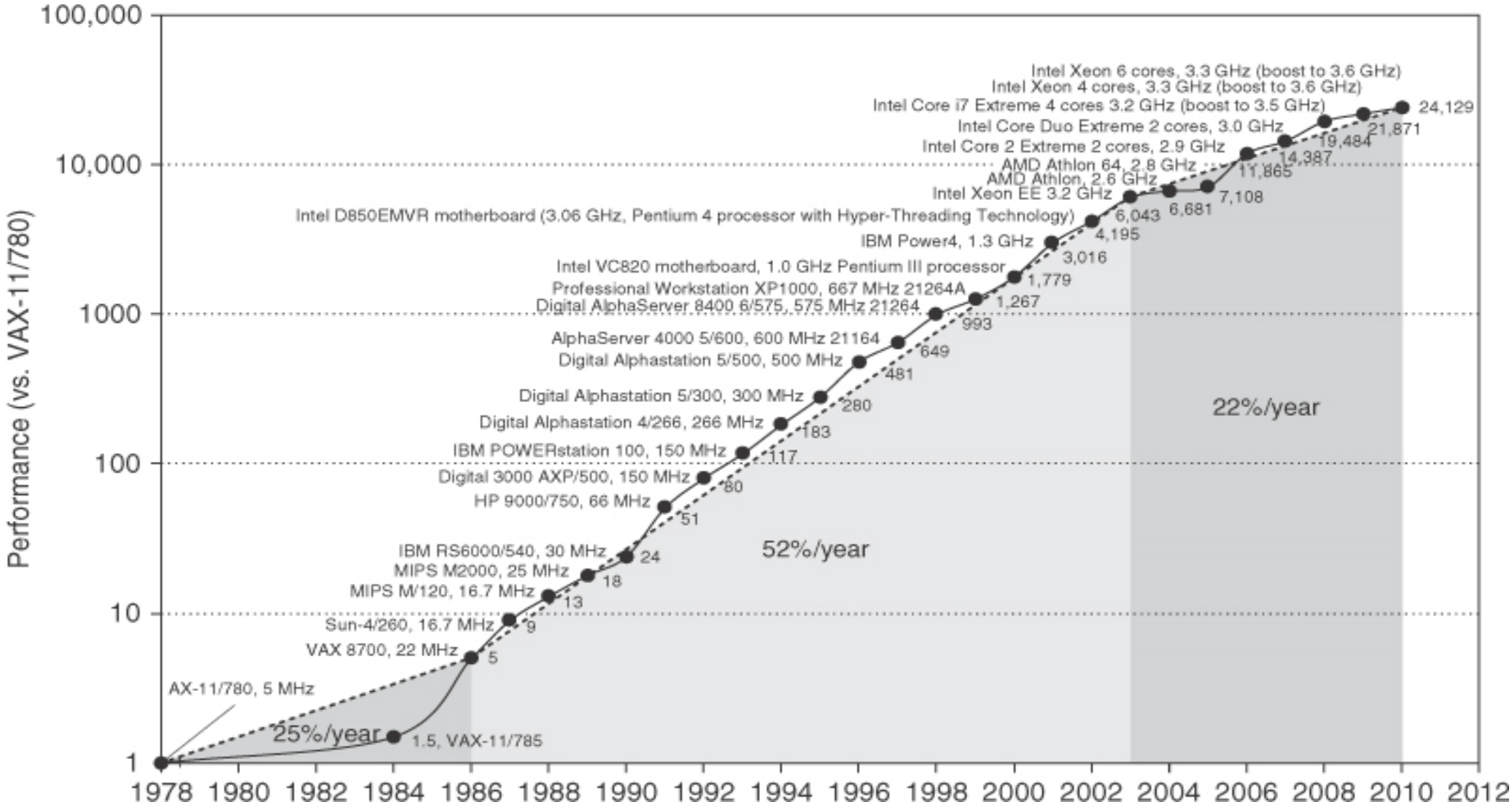


Figure 1.11 Growth in clock rate of microprocessors in Figure 1.1. Between 1978 and 1986, the clock rate improved less than 15% per year while performance improved by 25% per year. During the "renaissance period" of 52% performance improvement per year between 1986 and 2003, clock rates shot up almost 40% per year. Since then, the clock rate has been nearly flat, growing at less than 1% per year, while single processor performance improved at less than 22% per year.

From CAQA 5th edition

Growth in processor performance



From CAQA 5th edition

Which is faster?

Plane	DC to Paris	Speed	Passengers	Throughput (p x mph)
Boeing 747	6.5 hours	610 mph (1130km/h)	470	286,700 (470 x 610)
BAC Concorde	3 hours	1350 mph (2500km/h)	132	178,200 (132 x 1350)

- Time to run the task (ExTime)
  - Execution time, response time, **latency**
- Tasks per day, hour, week, sec, ns … (Performance)
  - **Throughput**, bandwidth

MPH (Mile Per Hour)

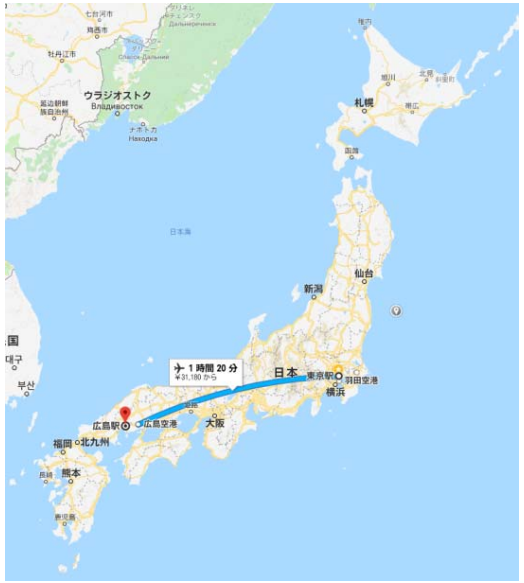
From the lecture slide of David E Culler

Which is faster?

From Tokyo to Hiroshima

	Time Cost	Max Speed	Passengers	Throughput (P x S)
Boeing 737	1:20 32,000yen	800km/h (670km)	170	85,510 (170 x 503)
Nozomi	4:00 18,000yen	270km/h (820km)	1,300	266,500 (1,300 x 205)

- Time to run the task (ExTime)
  - Execution time, response time, **latency**
- Tasks per day, hour, week, sec, ns … (Performance)
  - **Throughput**, bandwidth



From the lecture slide of David E Culler

Defining (Speed) Performance

Normally interested in reducing

Response time (execution time) – the time between the start and the completion of a task or a program

Important to individual users

Thus, to maximize performance, need to minimize execution time

$$performance_X = 1 / execution\_time_X$$

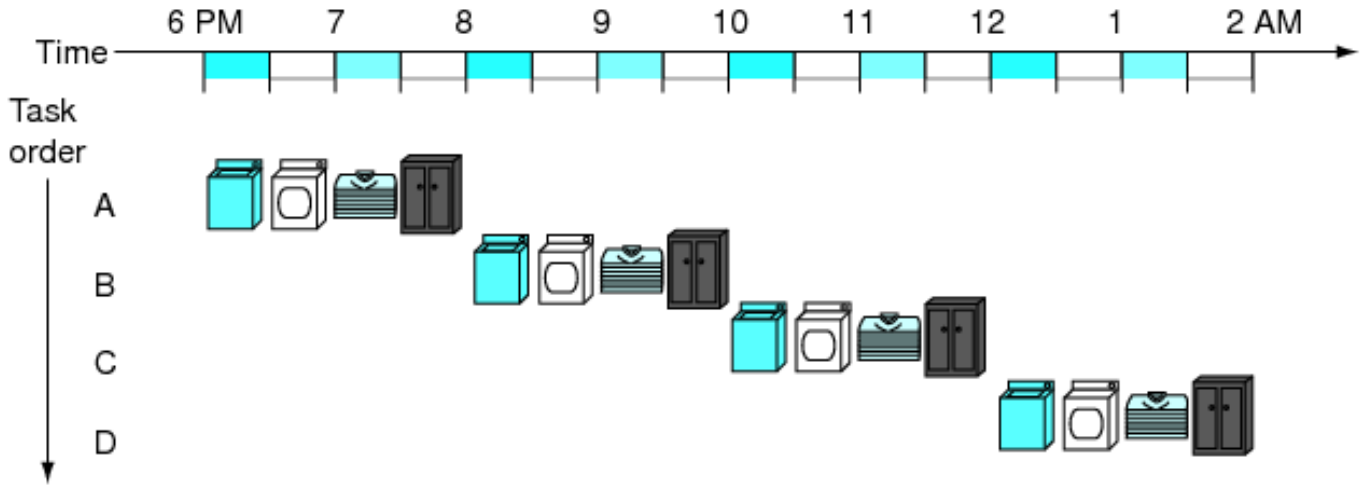
If X is n times faster than Y, then

$$\frac{performance_X}{performance_Y} = \frac{execution\_time_Y}{execution\_time_X} = n$$

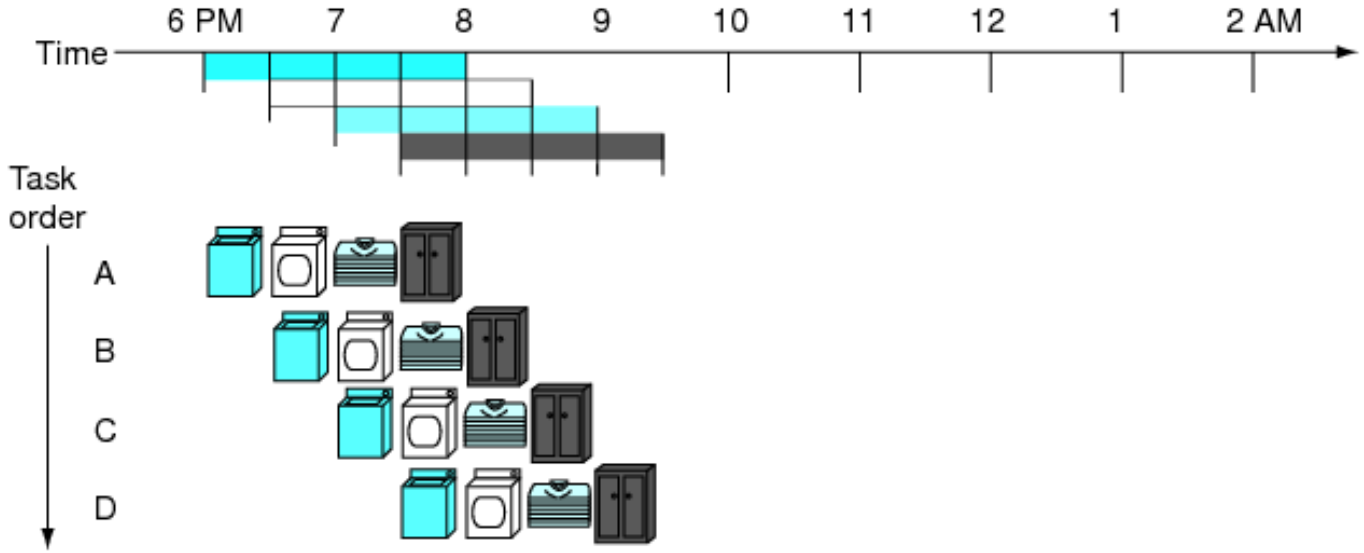
- Throughput – the total amount of work done in a given time
  - Important to data center managers
- Decreasing response time almost always improves throughput

Pipelined Processor

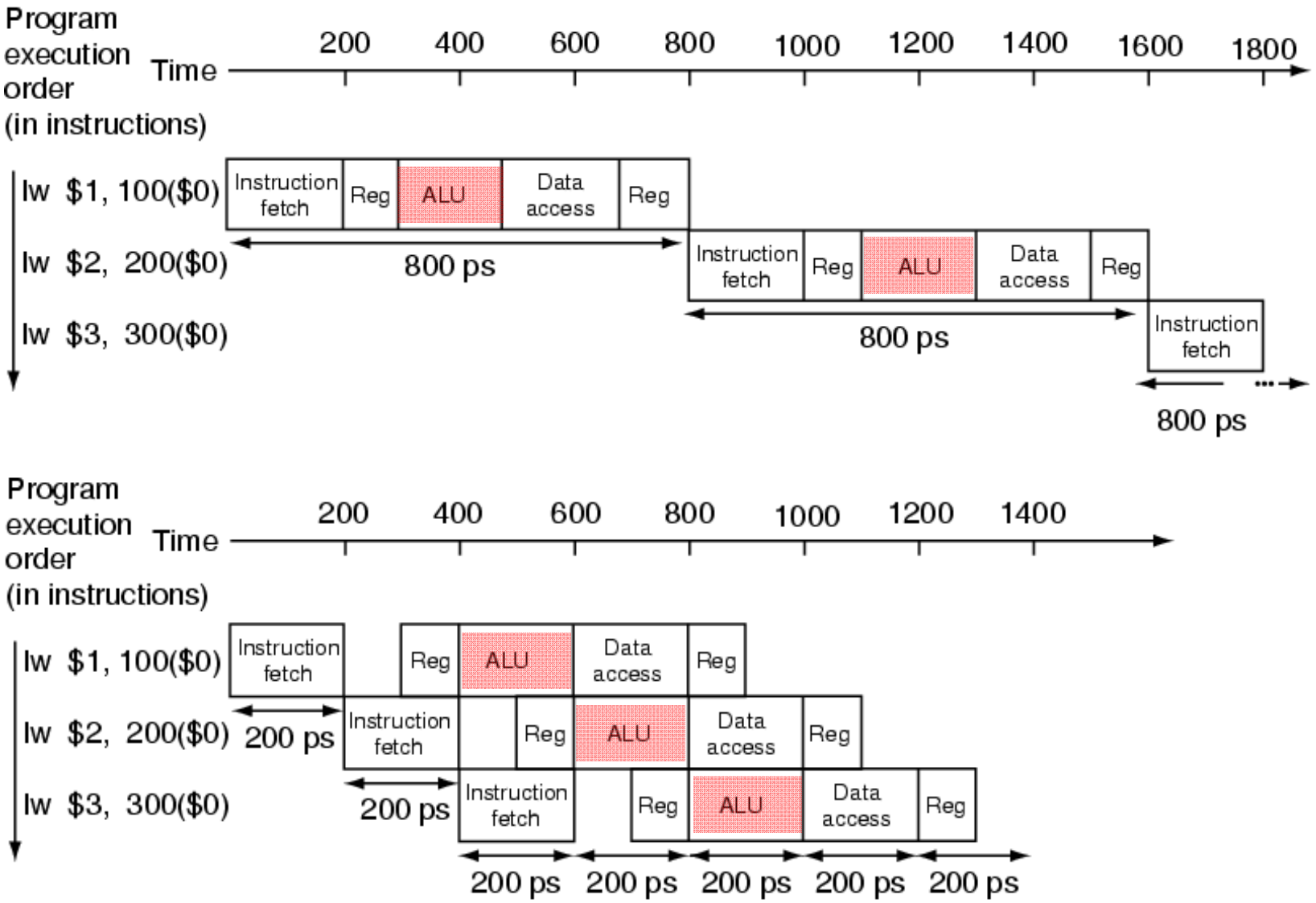
- Non pipelining (Multi-cycle)



- Pipelining

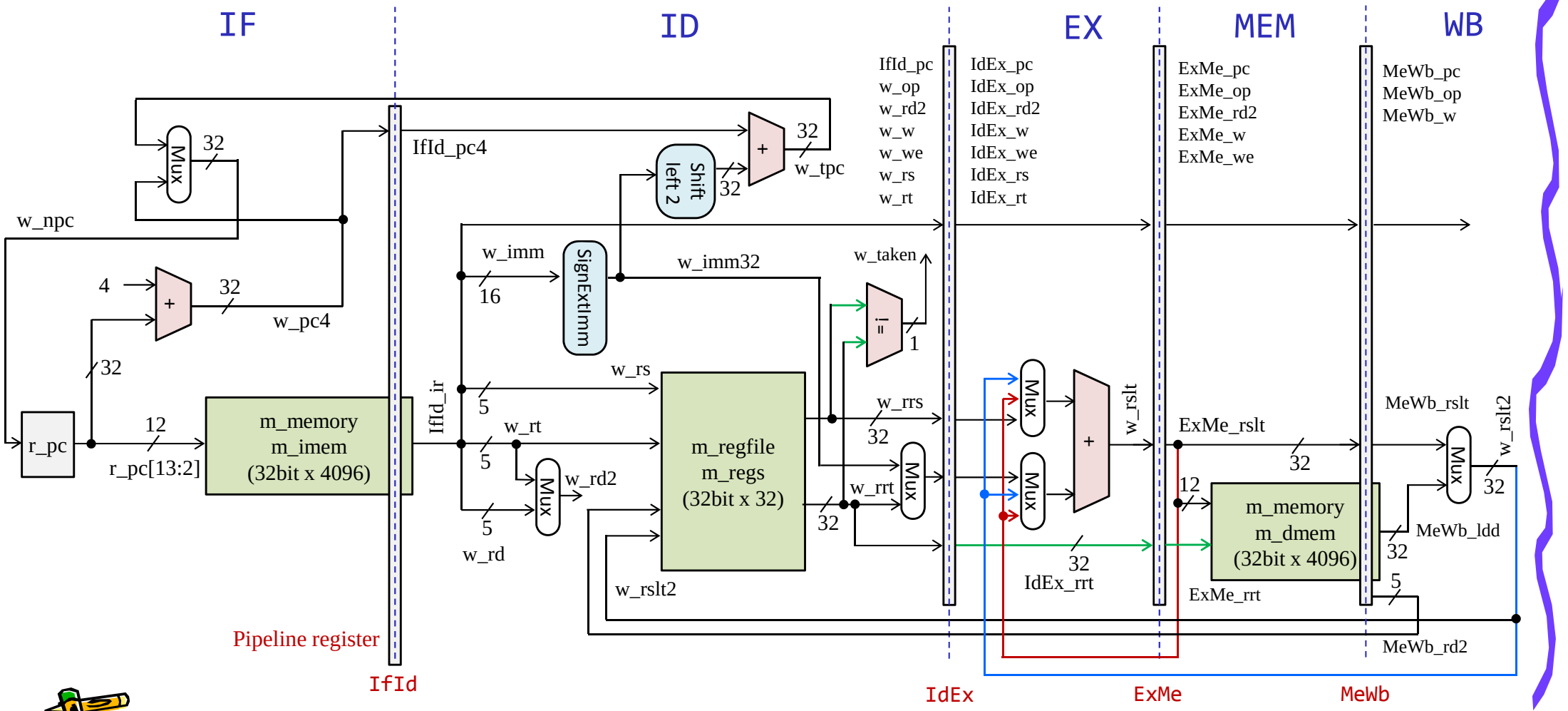


Pipelined Processor



Inside module m_proc12 (pipelined processor)

- add, addi, lw, sw, bne, halt命令に対応したパイプライン版
- 図では省略しているが，分岐のための比較，データメモリの入力データにもフォワーディングが必要



Performance Factors

Want to distinguish elapsed time and the time spent on our task

CPU execution time (CPU time) : time the CPU spends working on a task

Does not include time waiting for I/O or running other programs

$$\text{CPU execution time for a program} = \text{\# CPU clock cycles for a program} \times \text{clock cycle time}$$

or

$$\text{CPU execution time for a program} = \frac{\text{\# CPU clock cycles for a program}}{\text{clock rate}}$$

- Can improve performance by reducing either the length of the clock cycle or the number of clock cycles required for a program

Performance Factors

$$\text{CPU execution time for a program} = \frac{\text{\# CPU clock cycles for a program}}{\text{clock rate}}$$

Performance = clock rate x 1 / # CPU clock cycles for a program

Performance = f x IPC

f: frequency (clock rate)

IPC: retired instructions per cycle

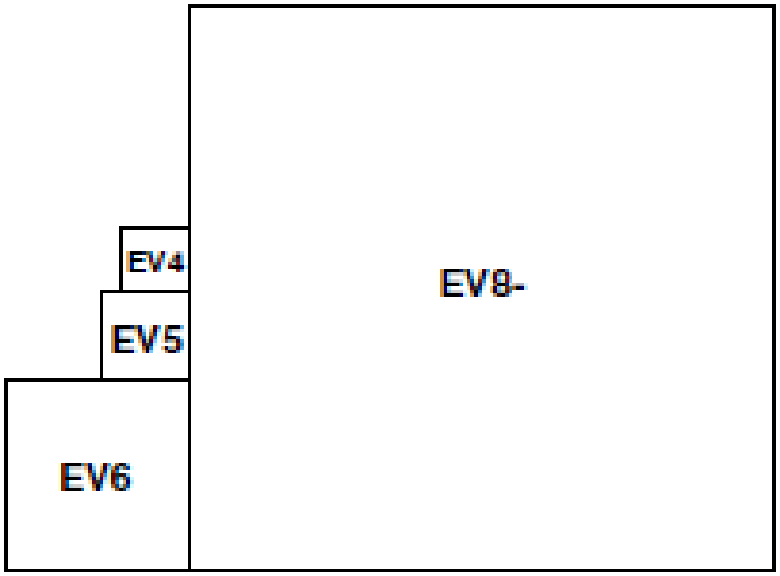
```
int flag = 1;

int foo(){
    while(flag);
}
```

Pollack's Rule

- Pollack's Rule states that microprocessor "performance increase due to microarchitecture advances is roughly proportional to the square root of the increase in complexity".
  Complexity in this context means processor logic, i.e. its area.

- Superscalar, vector
  - Instruction level parallelism, data level parallelism

From multi-core era to many-core era



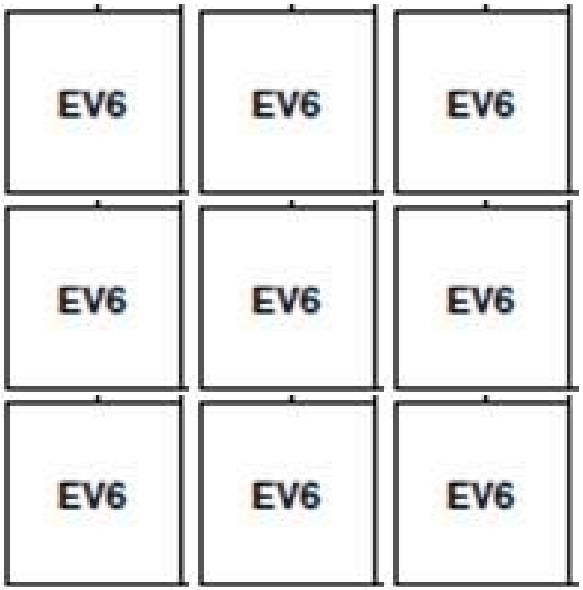


Figure 1. Relative sizes of the cores used in the study

Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction, MICRO-36

From multi-core era to many-core era

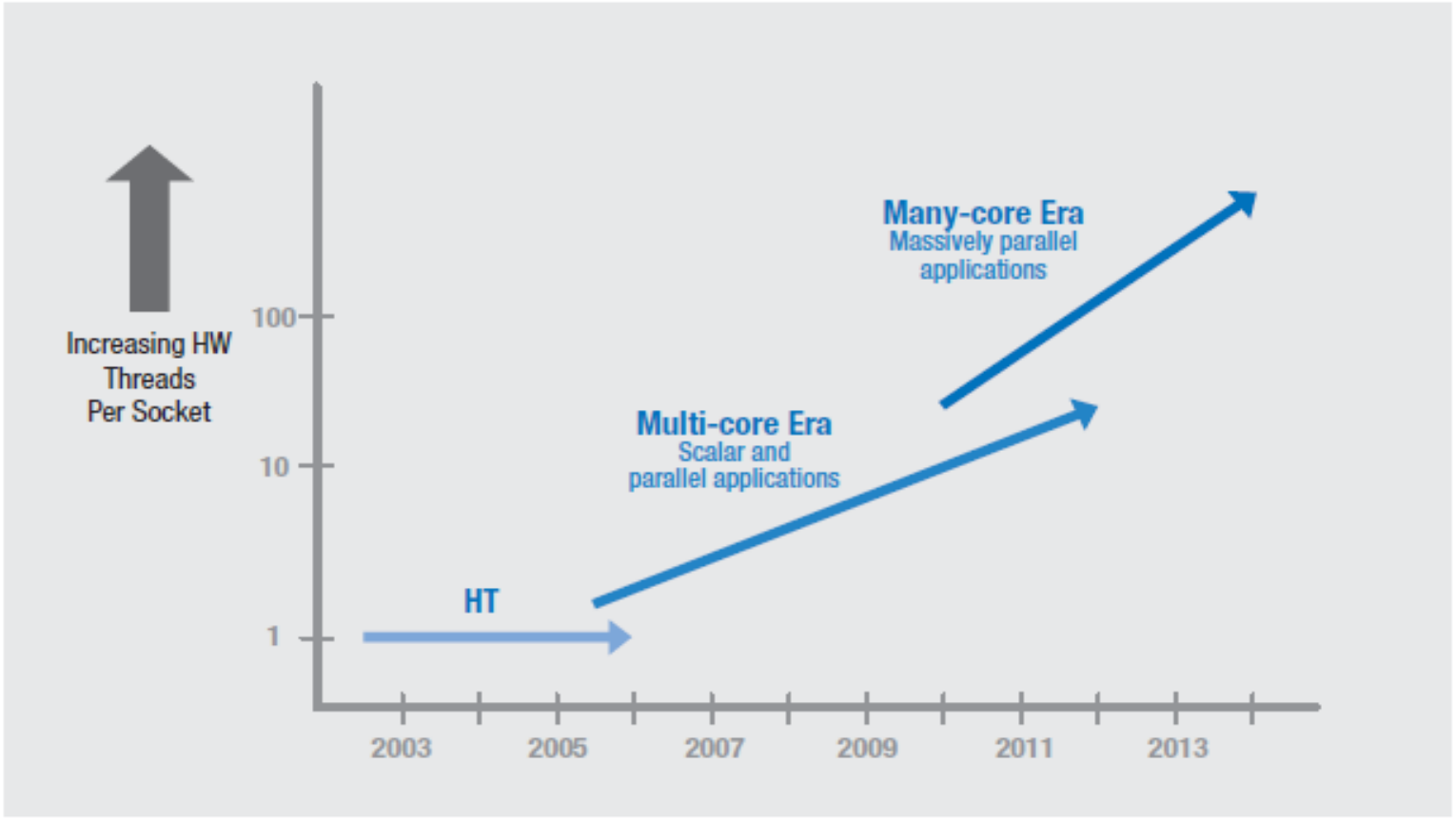
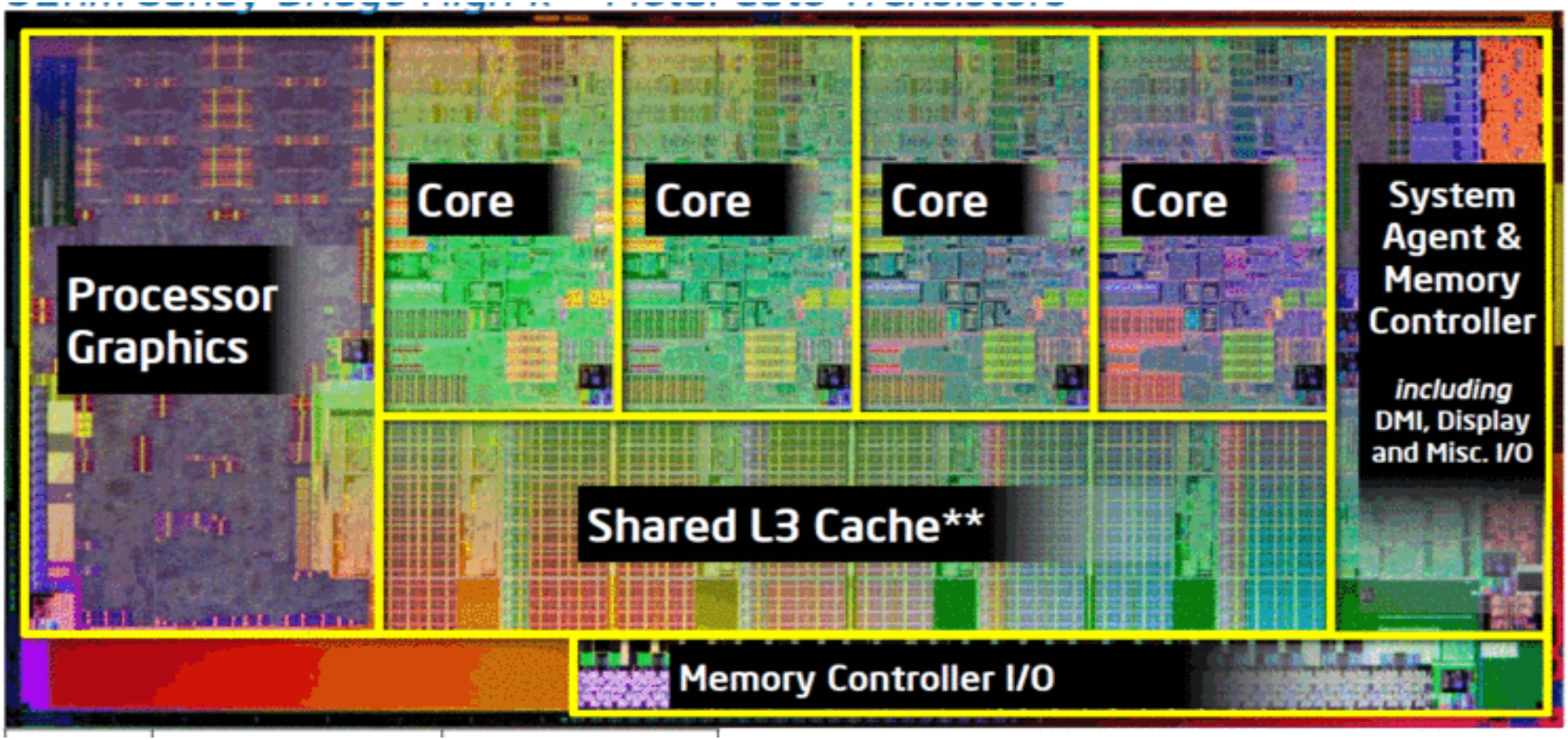


Figure 1: Current and expected eras of Intel® processor architectures

Platform 2015: Intel® Processor and Platform Evolution for the Next Decade, 2005

Intel Sandy Bridge, January 2011

4 to 8 core



アーキテクチャの異なる視点による分類

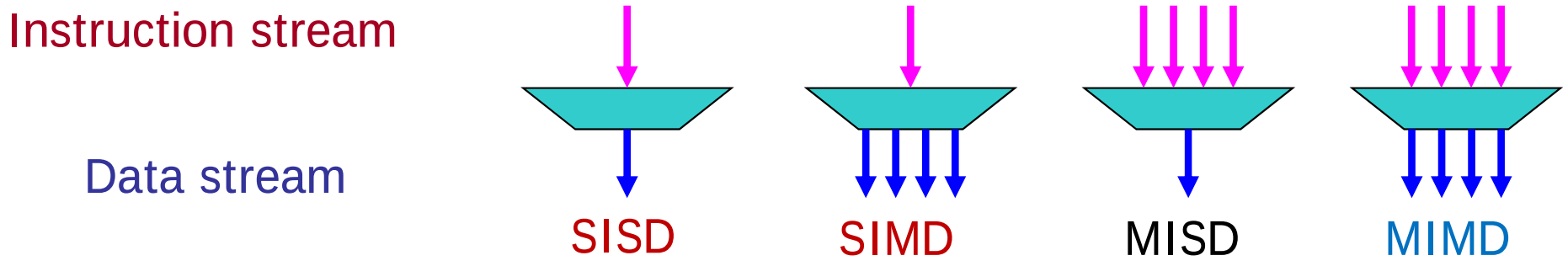
Flynnによる命令とデータの流れに注目した並列計算機の分類(1966年)

SISD (Single Instruction stream, Single Data stream)

SIMD(Single Instruction stream, Multiple Data stream)

MISD (Multiple Instruction stream, Single Data stream)

MIMD (Multiple Instruction stream, Multiple Data stream)



アーキテクチャの異なる視点による分類

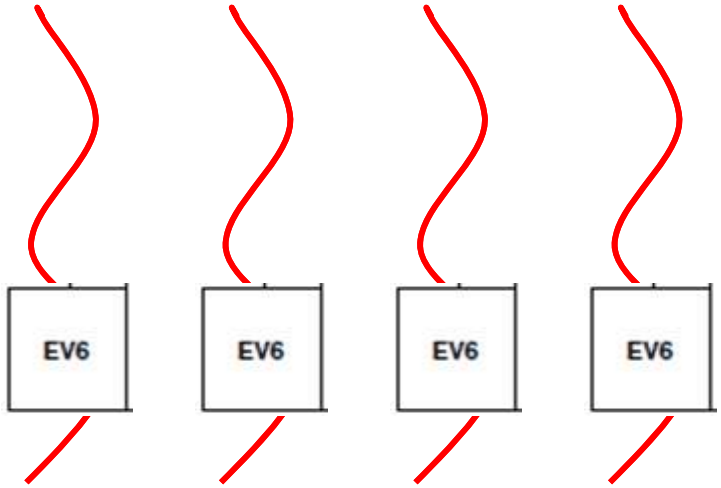
Flynnによる命令とデータの流れに注目した並列計算機の分類(1966年)

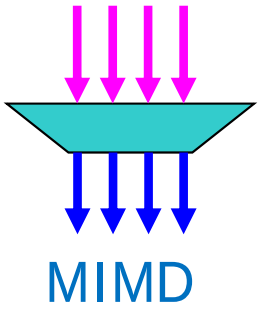
SISD (Single Instruction stream, Single Data stream)

SIMD(Single Instruction stream, Multiple Data stream)

MISD (Multiple Instruction stream, Single Data stream)

MIMD (Multiple Instruction stream, Multiple Data stream)





Amdahl's Law

- If **F** is the fraction of a calculation that is sequential, and (1-**F**) is the fraction that can be parallelized, then
  the maximum speedup that can be achieved by using **P** processors is 1/(**F**+(1-**F**)/**P**).

- Multi/many core
  - Thread level parallelism