

Multicore Computing

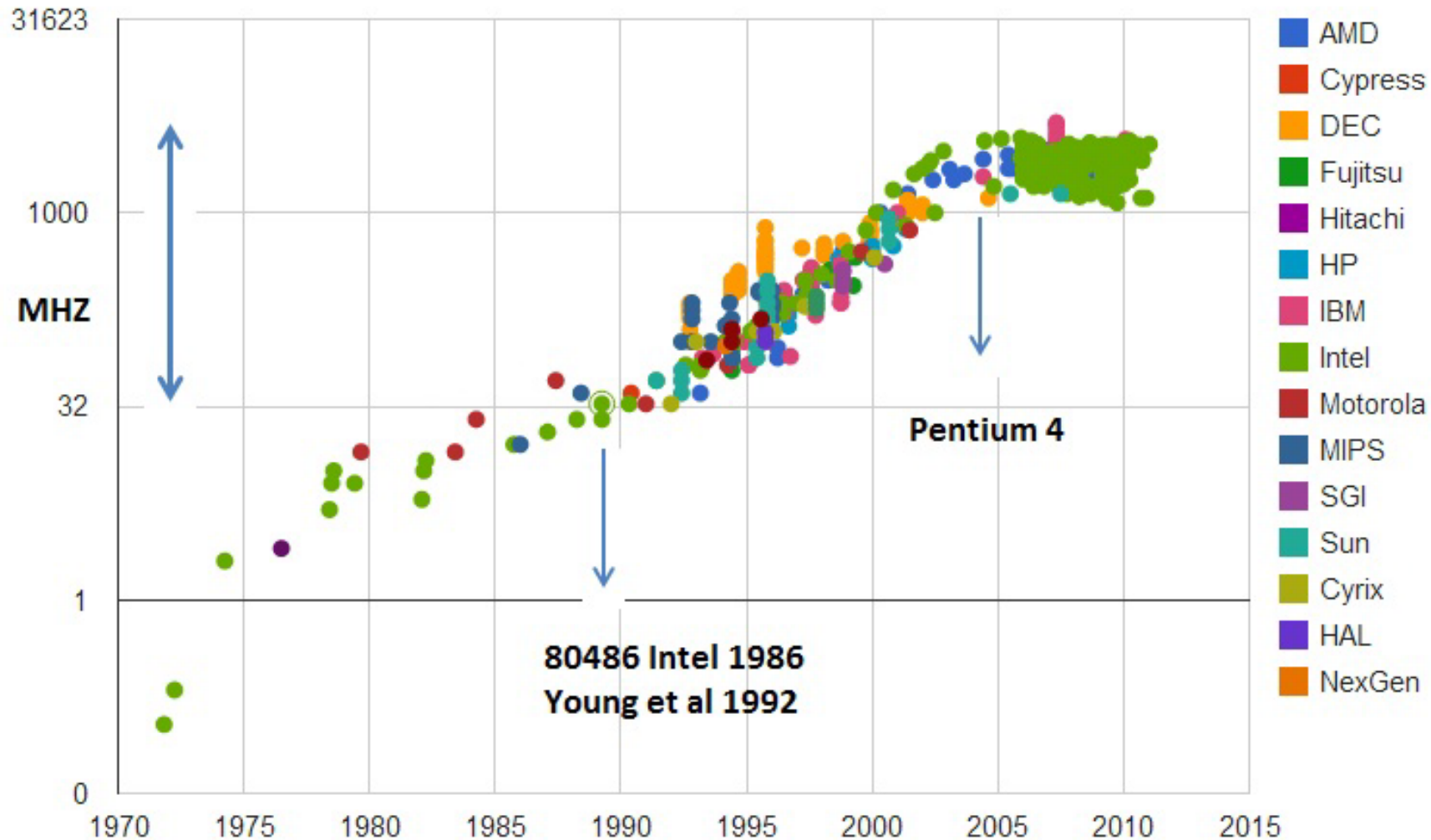
Weijie Zhao

09/01/2026

“Premature optimization is the root of all evil”

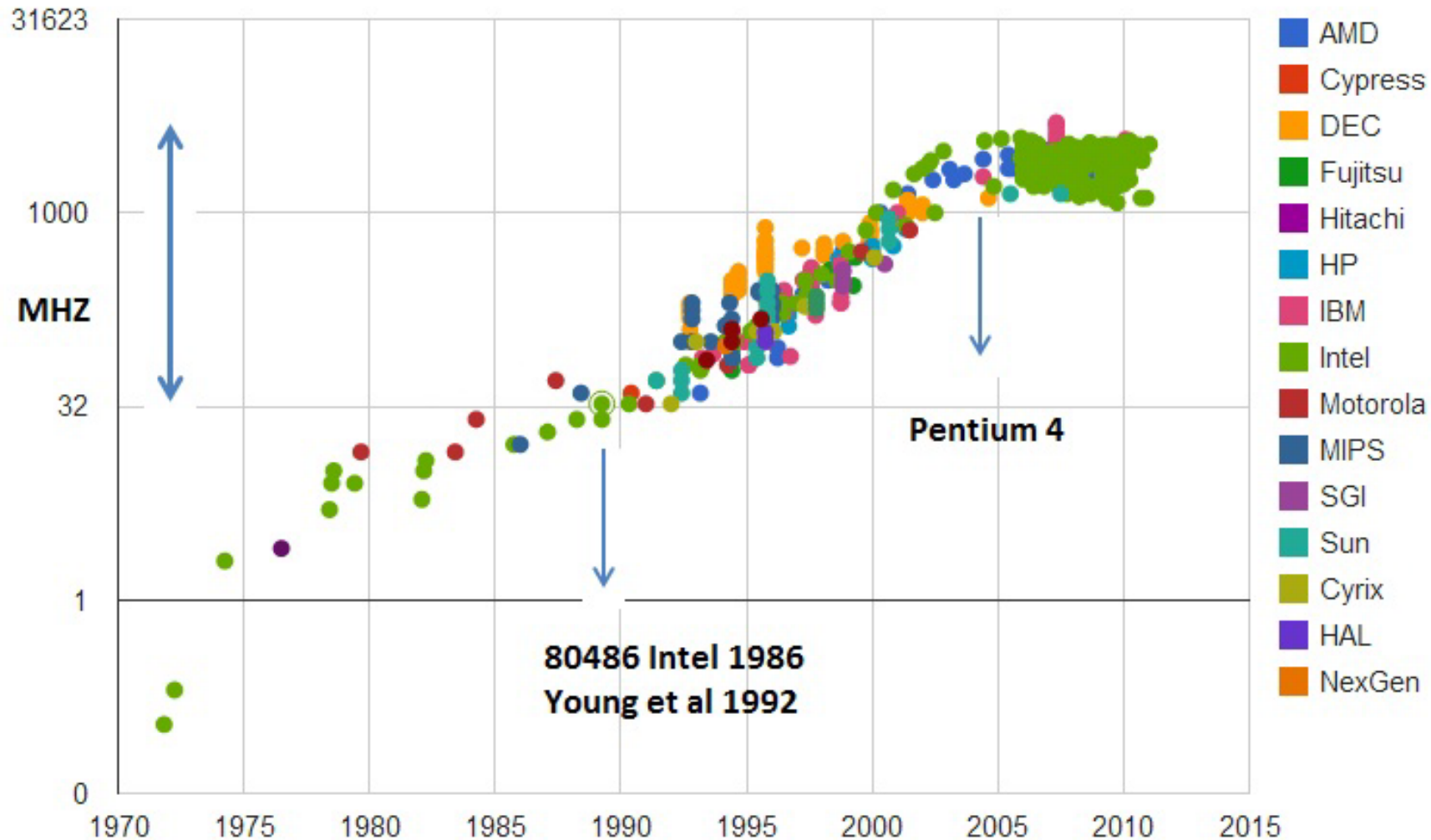
--- Sir Tony Hoare

CPU Clock Rate



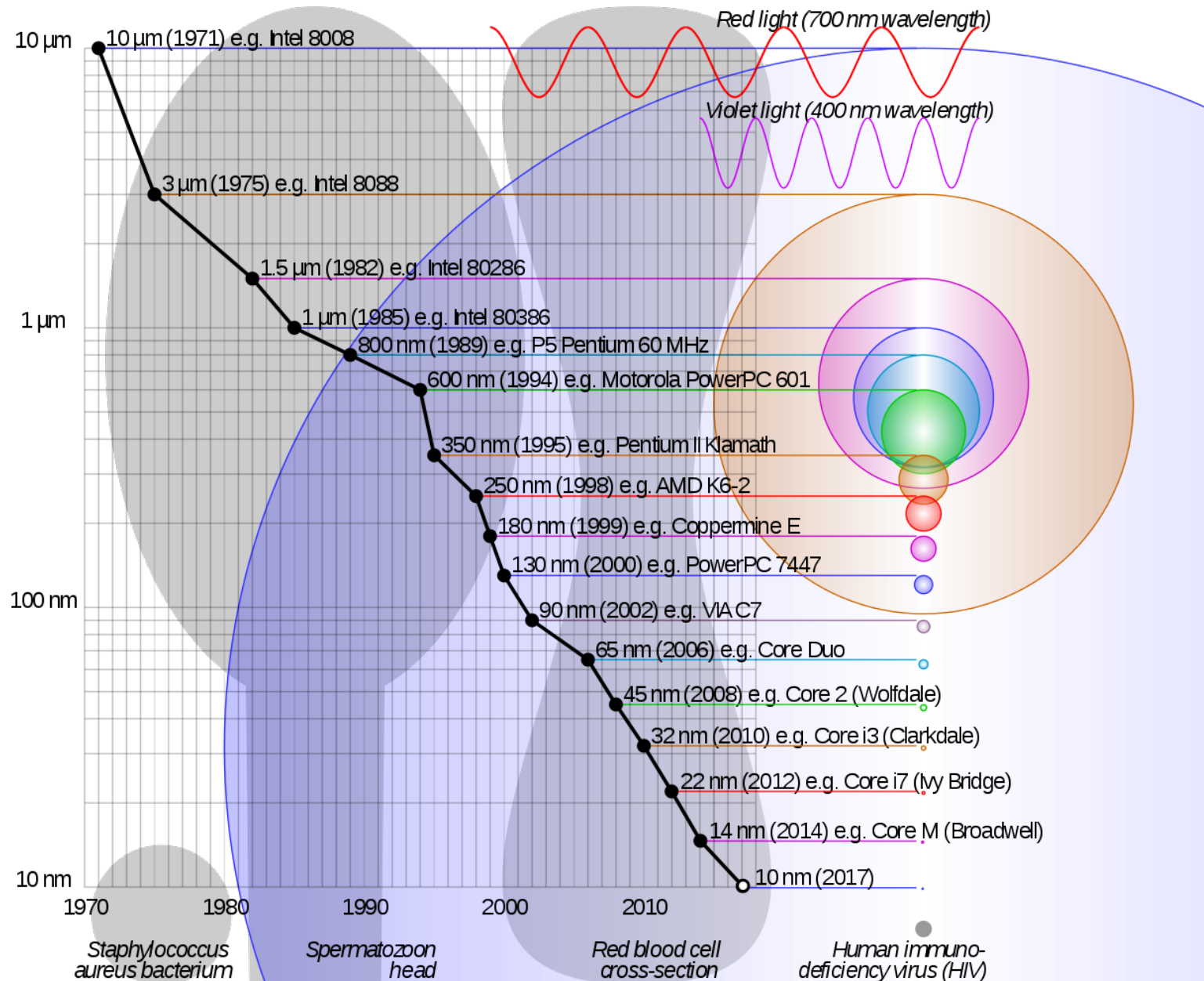
CPU Clock Rate

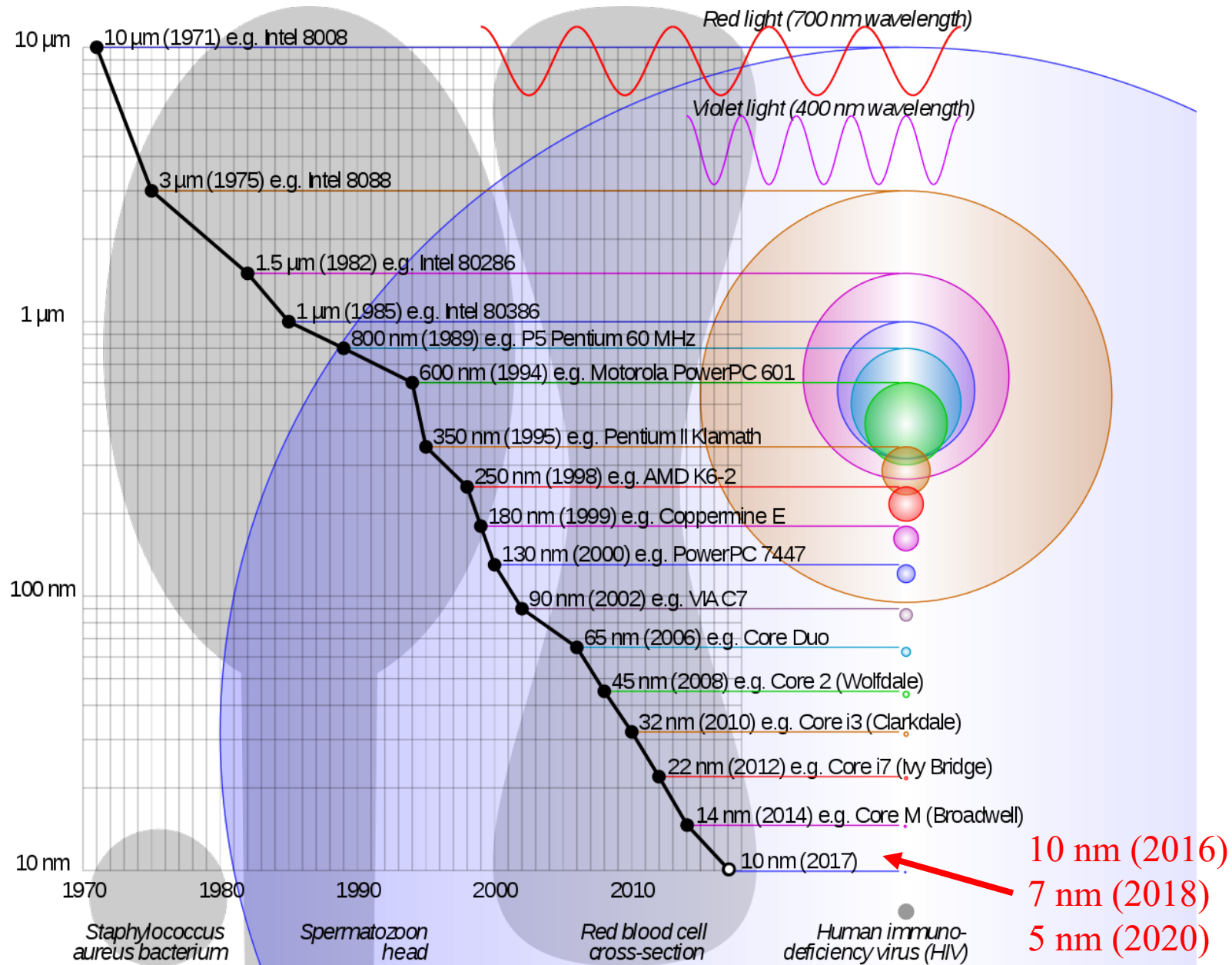
Now our CPU clock is
still around 2-3 GHz

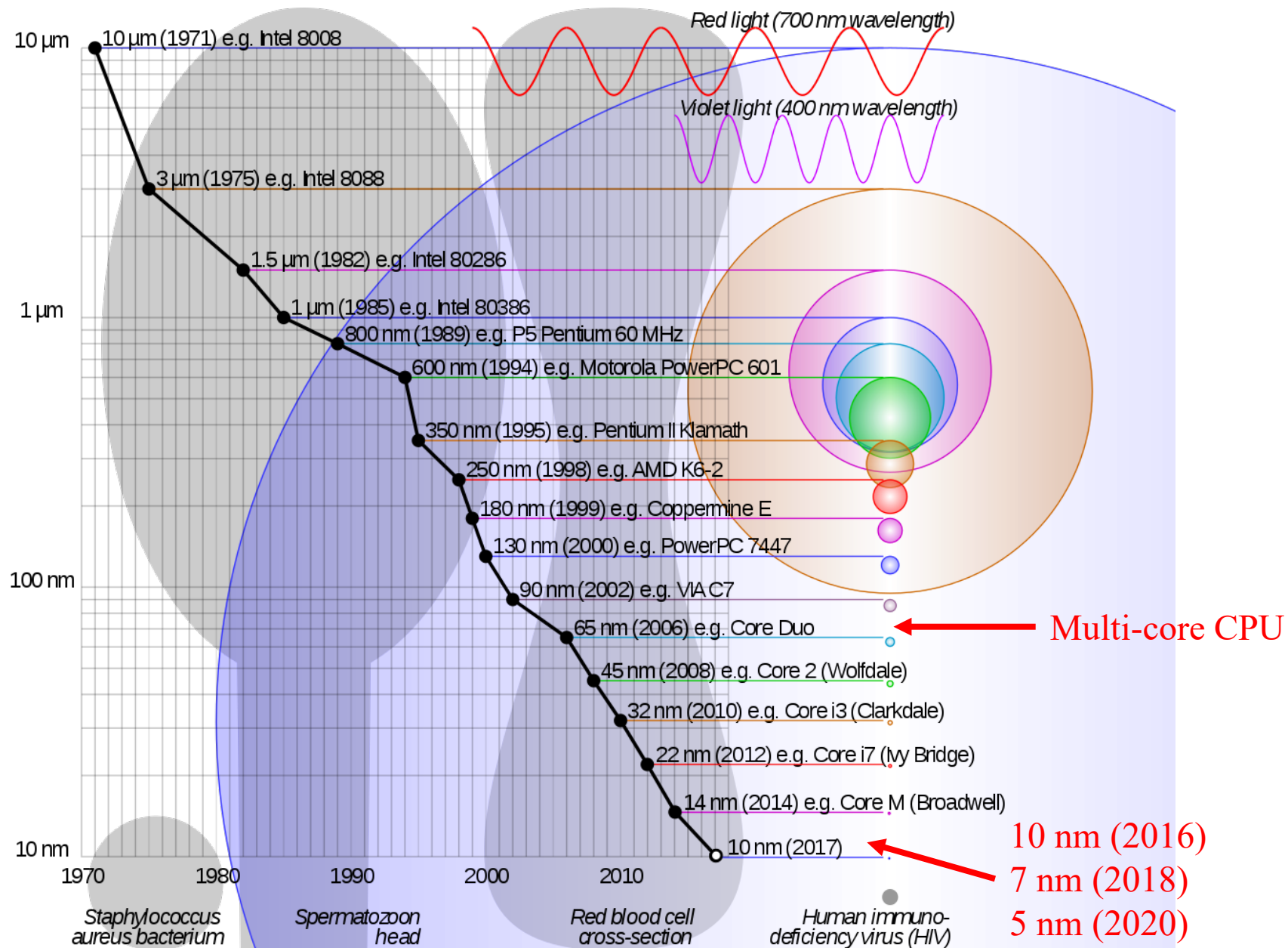


Why the Clock Rate Does Not Increase?

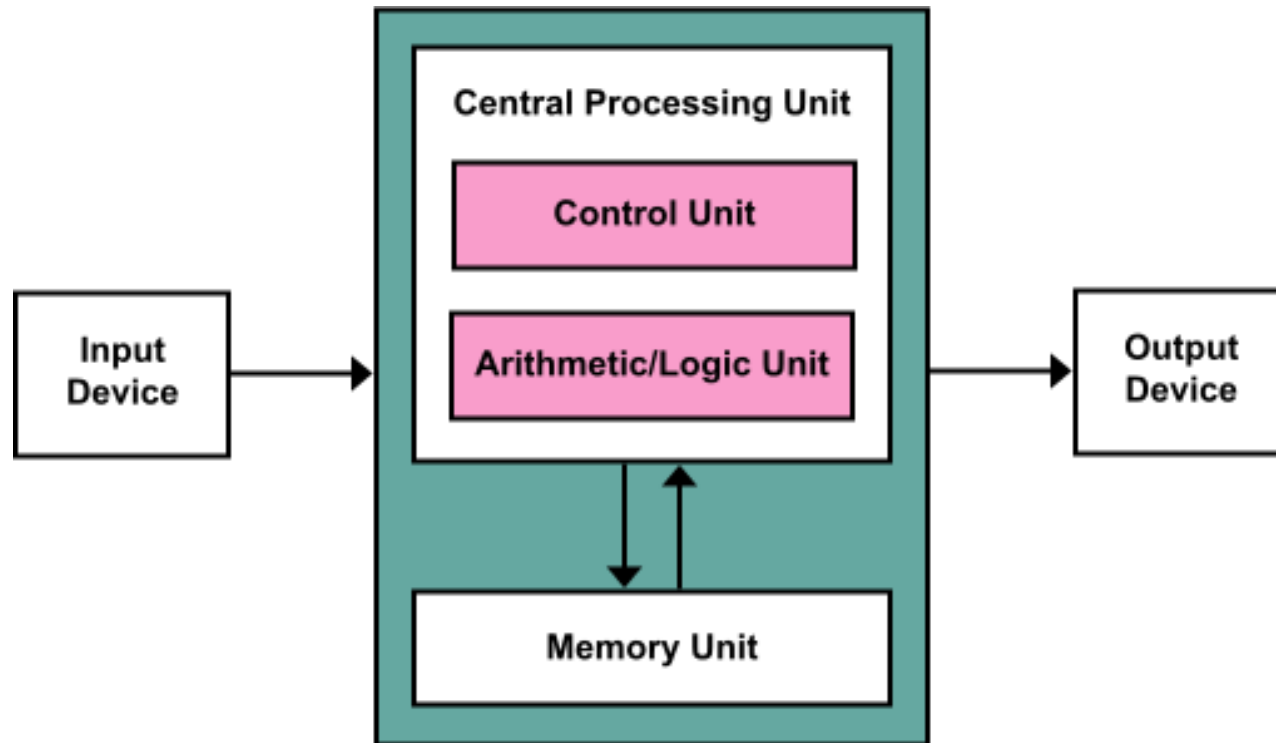
- Power density becomes extremely high
- Heating







Major Components of a CPU



Execution of Instructions

- Instruction Fetch
- Instruction Decode
- Memory Access
- Register Writeback

Execution of Instructions

- Instruction-level parallelism
- SIMD Intrinsics
- Hyper-Threading
- Out-of-order-execution
- Branch Prediction
- Meltdown

Threads

- Process Control Block
 - Process structuring information
 - Process State
 - Process Number (PID)
 - Program Counter (PC)
 - CPU Registers
 - Memory Management Information
 - Accounting Information
 - I/O Status Information
 - ...

Thread Scheduling

- Context Switching
 - Save/Load PCB
- Thread Pool

OpenMP

- Open Multi-Processing
- An API that supports multi-platform shared-memory multiprocessing programming in C, C++, and Fortran

OpenMP: Quick Start

```
for (int i = 0; i < N; ++i) {  
    b[i] = a[i] + 1;  
}
```

OpenMP: Quick Start

```
#pragma omp parallel for schedule(static) num_threads(8)
for (int i = 0; i < N; ++i) {
    b[i] = a[i] + 1;
}
```

- `g++ test.cc -fopenmp -o test -O2`

OpenMP: Quick Start

```
#pragma omp parallel for schedule(static) num_threads(8)  
for (int i = 0; i < N; ++i) {  
    b[i] = a[i] + 1;  
}
```

OpenMP: Quick Start

```
int sum = 0;
```

```
for (int i = 0; i < N; ++i)
```

```
{
```

```
    sum += a[i];
```

```
}
```

OpenMP: Quick Start

```
int sum = 0;
```

```
#pragma omp parallel for schedule(static) default(shared)  
reduction(+:sum) num_threads(8)
```

```
for (int i = 0; i < N; ++i)
```

```
{
```

```
    sum += a[i];
```

```
}
```

HW 1: Sorting

- Given a sequence of integers, output its sorted result.
- 10 independent test cases. Each case weights 1 pt.
- The compilation is considered failed if it does not finish in **1 minute**.
- A test case is considered **incorrect** if it does not finish in **10 minutes**.
- The **summation** of the execution time across 10 cases will be used to rank **correct** solutions.
- Due: 09/11/2026 05:00 pm EST

Grading

- Homework 40%
- Reading 10%
- Project 50%

- $90\% \leq A \leq 100\%$
- $80\% \leq B < 90\%$
- $70\% \leq C < 80\%$
- $60\% \leq D < 70\%$
- $0\% \leq F < 60\%$

- 5 pieces of homework.
- No late submissions.
- No 3rd party code
- Automatically tested: Please **strictly** follow the output format. An incorrect format is considered as a wrong answer.
- The **best 4** scores among the 5 are counted in your final grade.
- The fastest correct solution in each homework gets **10% bonus score in the final grade**.
- Other correct solutions that are no slower than 2X of the fastest one gets **5% bonus score in the final grade**.

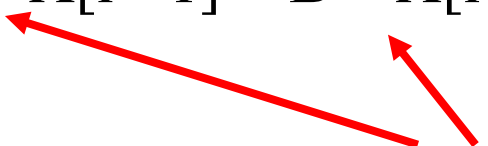
Input Data

- First line contains 6 integers: N K A B C M
- We have K lines in the following. Each line contains 1 integer, representing $X[i]$ for $0 \leq i < K$
- For $i \geq K$
 - $X[i] = (A * X[i - 1] + B * X[i - 2] + C) \% M$
- $N \leq 10^8$
- $M \leq 10^9 + 7$

Input Data

- First line contains 6 integers: N K A B C M
- We have K lines in the following. Each line contains 1 integer, representing $X[i]$ for $0 \leq i < K$
- For $i \geq K$

- $X[i] = (A * X[i - 1] + B * X[i - 2] + C) \% M$



Caution the potential overflow here!

- $N \leq 10^8$
- $M \leq 10^9 + 7$

Input Data

- First line contains 6 integers: N K A B C M
- We have K lines in the following. Each line contains 1 integer, representing $X[i]$ for $0 \leq i < K$

- For $i \geq K$

- $X[i] = (A * X[i - 1] + B * X[i - 2] + C) \% M$

sample.txt:

10 5 257 31 93 100
35
19
86
38
51

sequence:

35
19
86
38
51
78
20
51
20
14

- $N \leq 10^8$
- $M \leq 10^9 + 7$

Output Format

- N lines, each line contains an integer number

sample.txt:

10 5 257 31 93 100
35
19
86
38
51

sequence:

35
19
86
38
51
78
20
51
20
14

output.txt:

14
19
20
20
35
38
51
51
78
86

What Do We Need to Do?

- We are required to complete two scripts
- `compiler.sh`
 - it is executed once before the actual testing starts
- `run.sh`
 - it should takes two arguments, the first argument is the input file name, the second one is the file name that you should write your sorted results into.
- The `test.sh` is used to evaluate your answer for all test cases.
- We only show 2 sample cases here. In the real testing stage, we will have 10 test cases. You can assume the largest test case will not fit in the memory.

Testing Environment

- `ssh yourusername@granger.cs.rit.edu`
- Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
- 48 threads in total (2 sockets, 12 cores per socket, 2 threads per core)
- 251 GB memory
- Testing limit:
 - 24 threads `taskset -c`

NUMA

- Non-uniform memory access
- QuickPath Interconnect (QPI)
- HyperTransport

OpenMP: Slow Start

- `#include<omp.h>`
- `void omp_set_num_threads(int num_threads)`
- `int omp_get_num_threads()`
- `int omp_get_thread_num()`

- `#pragma omp atomic (update/read/write/capture)`
- `#pragma omp critical`

Gauss-Seidel Smoother

- Solving PDE
- Parallel
- Synchronization
- Lock
- Communication