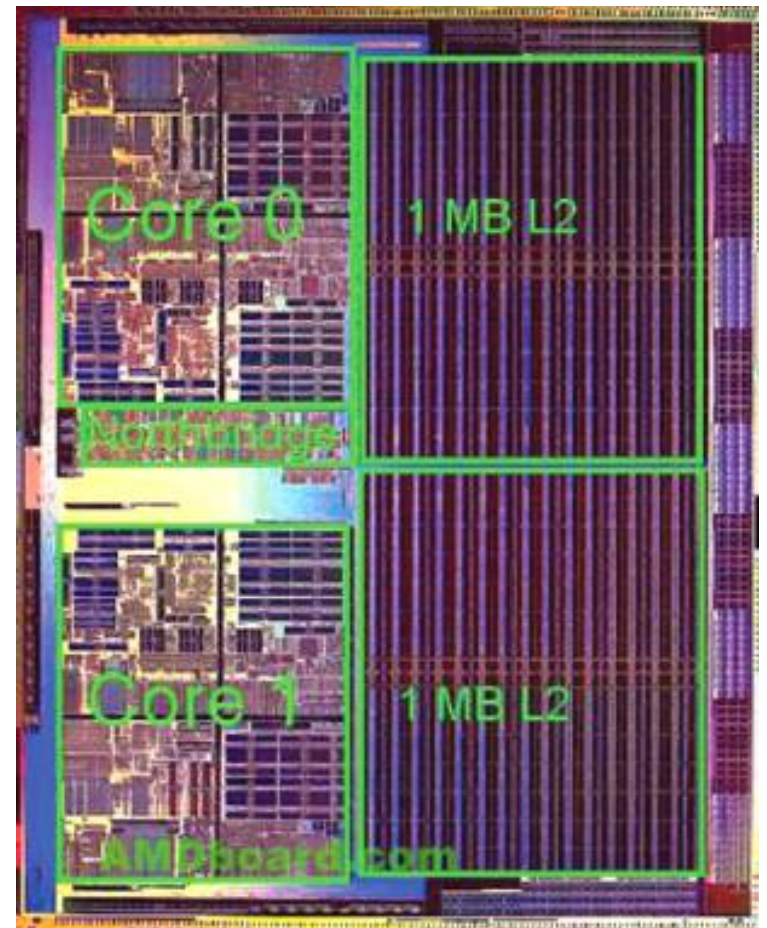


# Exposing Thread-level Parallelism

- Previously, we looked at exposing parallelism at the instruction level
  - SIMD: Single Instruction, Multiple Data
- Now, still parallelism, but at the thread-level.
  - What are Multi-Core Processors?
  - Why are they coming now?
  - How can we use them?

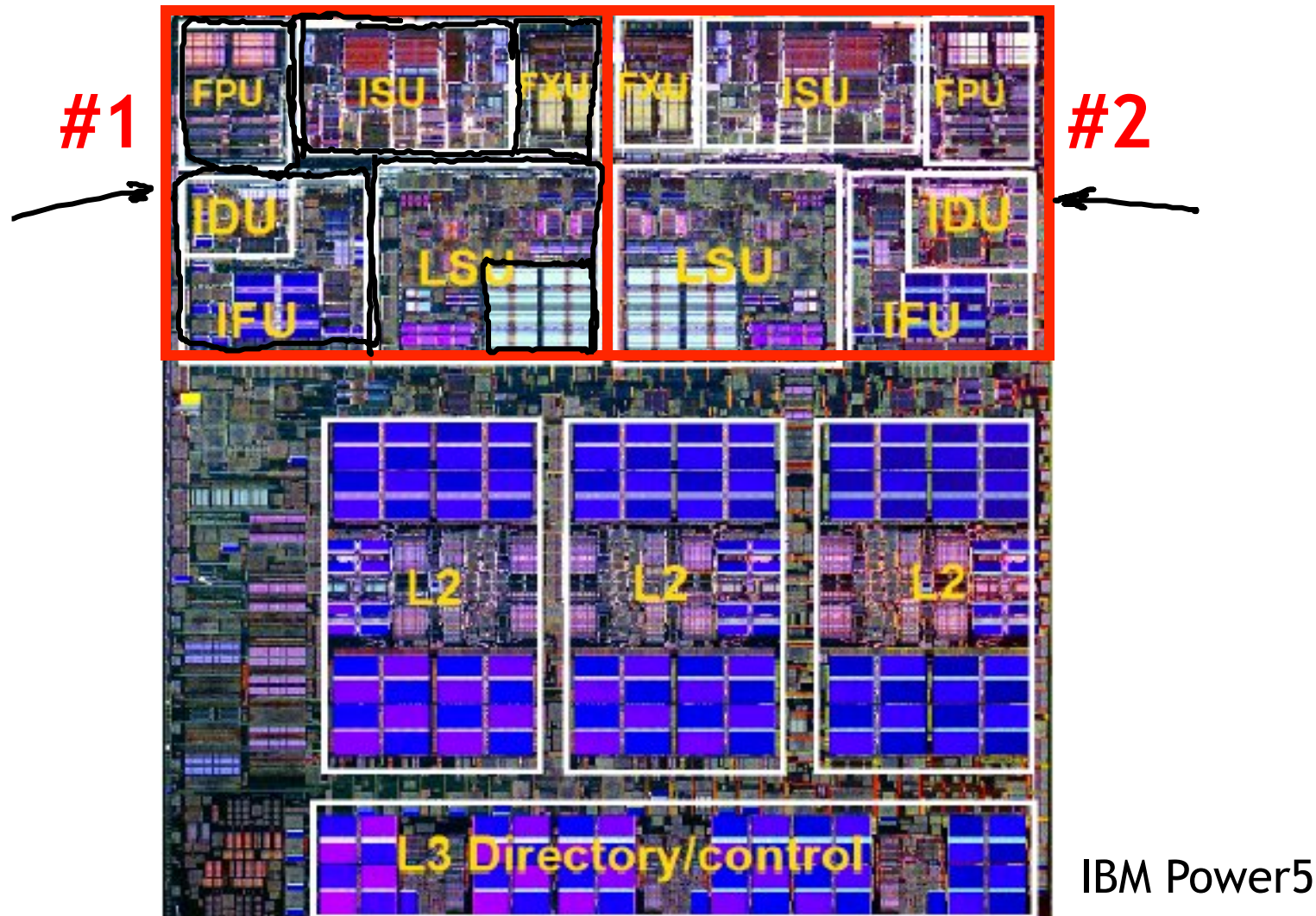
HANDOUT

AMD dual-core Opteron



# What are Multi-Core Processors

- Two (or more) complete processors, fabricated on the same silicon chip
- Execute instructions from two (or more) programs/threads at same time



# Multi-Cores are Everywhere.

---



It is now hard to buy a computer with 1 core

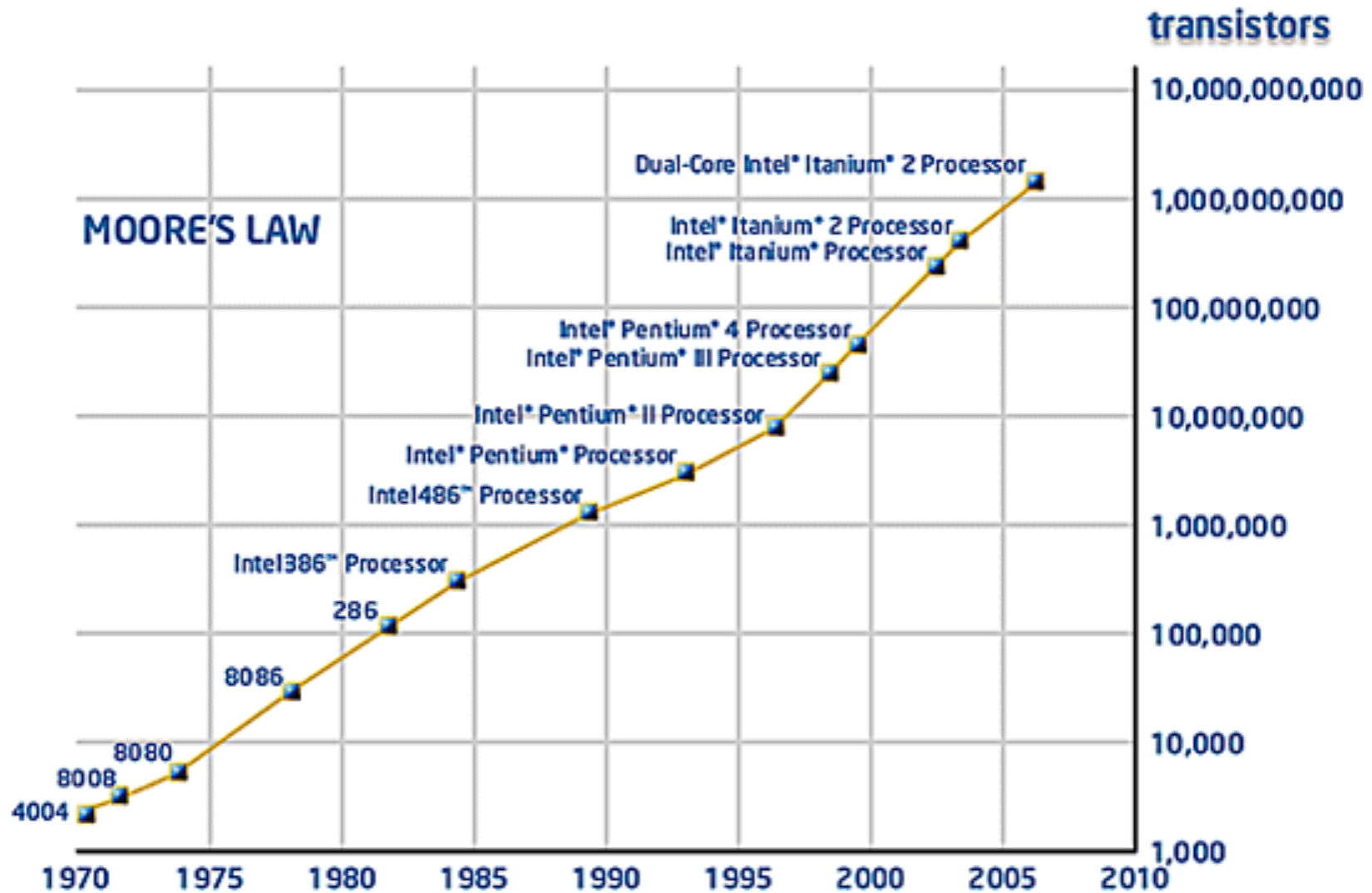
**XBox360: 3 PowerPC cores**



**Sony Playstation 3: Cell processor, an asymmetric multi-core with 9 cores (1 general-purpose, 8 special purpose SIMD processors)**

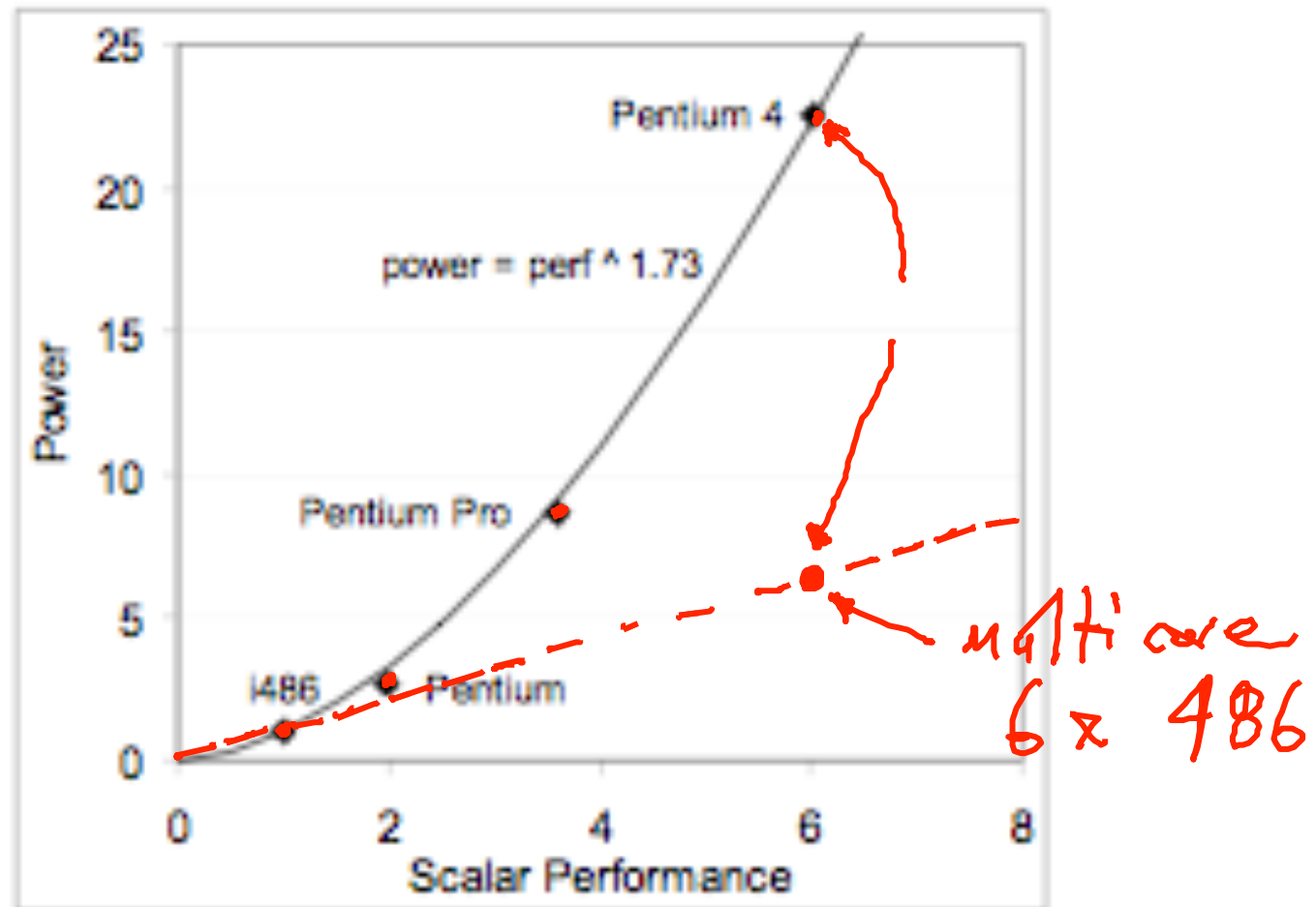
# Why Multi-cores Now?

- number of transistors we can put on a chip growing exponentially, and





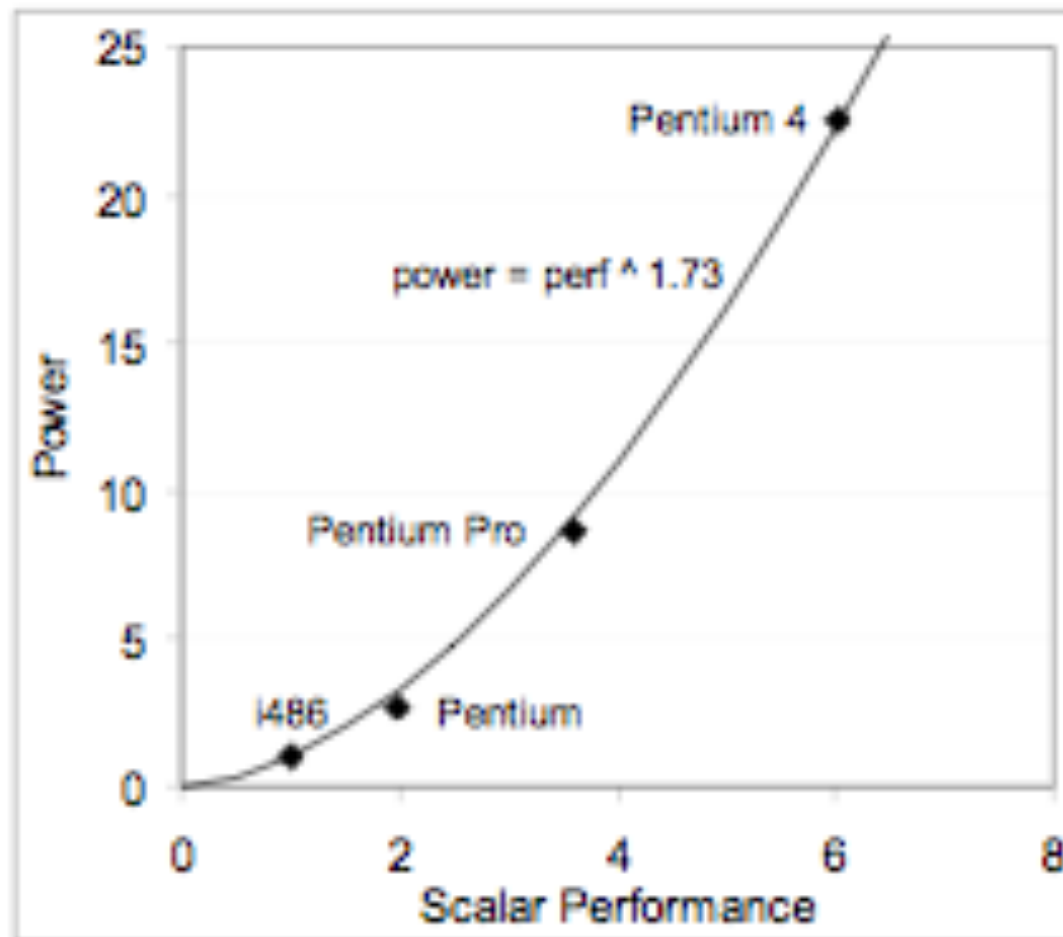
## And, performance is scaling up



- But, power is scaling up, also
  - Power has become limiting factor in current chips

# Hence, Multi-cores

---



# As programmers, do we care?

- What happens if we run a program on a multi-core?

```
void  
array_add(int A[], int B[], int C[], int length) {  
    int i;  
    for (i = 0 ; i < length ; ++ i) {  
        C[i] = A[i] + B[i];  
    }  
}
```

