



1


Parallel Programming: Moore's Law and Multicore



**Oregon State
University**
Mike Bailey
mjb@cs.oregonstate.edu



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/)



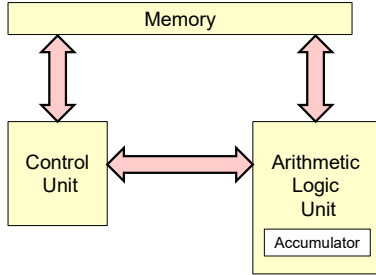
**Oregon State
University**
Computer Graphics

moore's law and multicore.pptx mjb - March 9, 2022

1


2

Von Neumann Architecture: Basically the fundamental pieces of a CPU have not changed since the 1960s



Other elements:

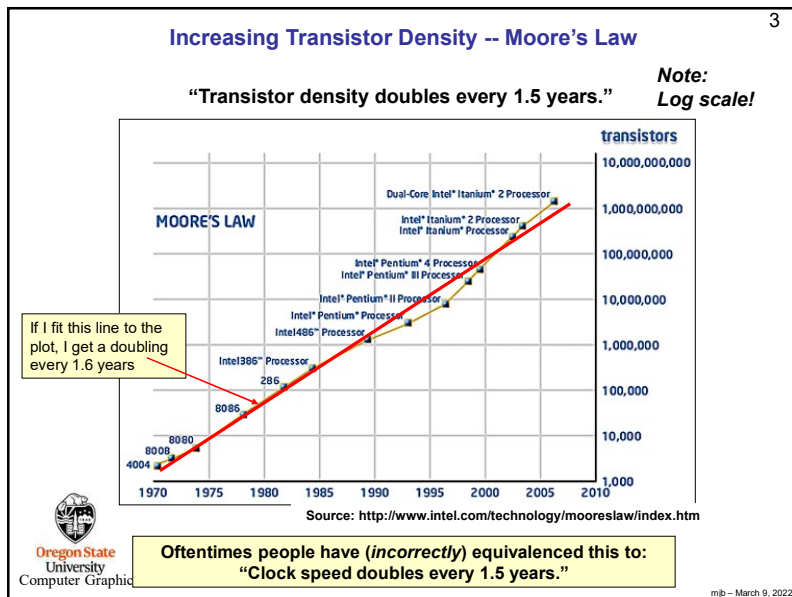
- Clock
- Registers
- Program counter
- Stack pointer



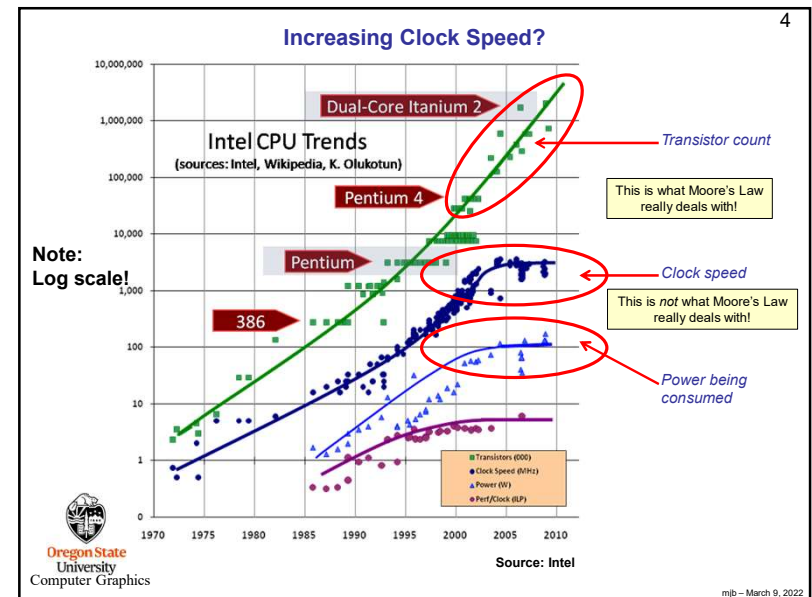
**Oregon State
University**
Computer Graphics

mjb - March 9, 2022

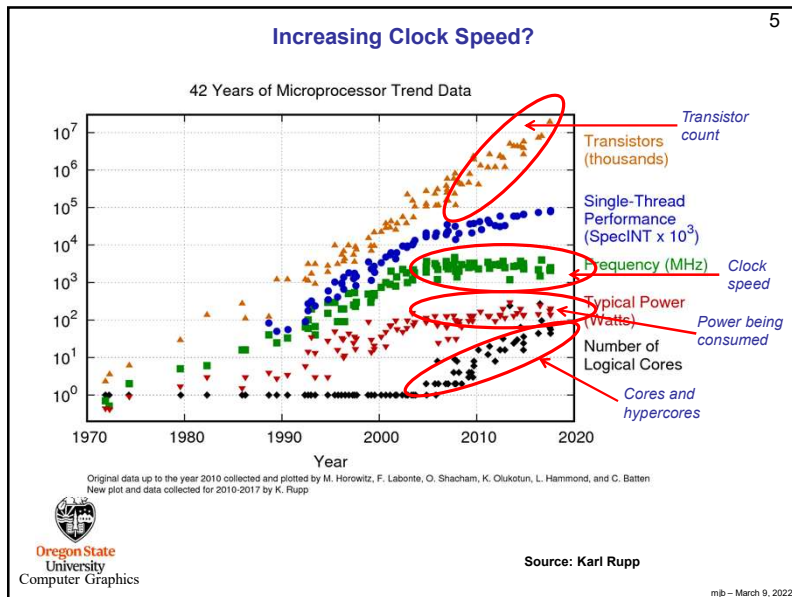
2



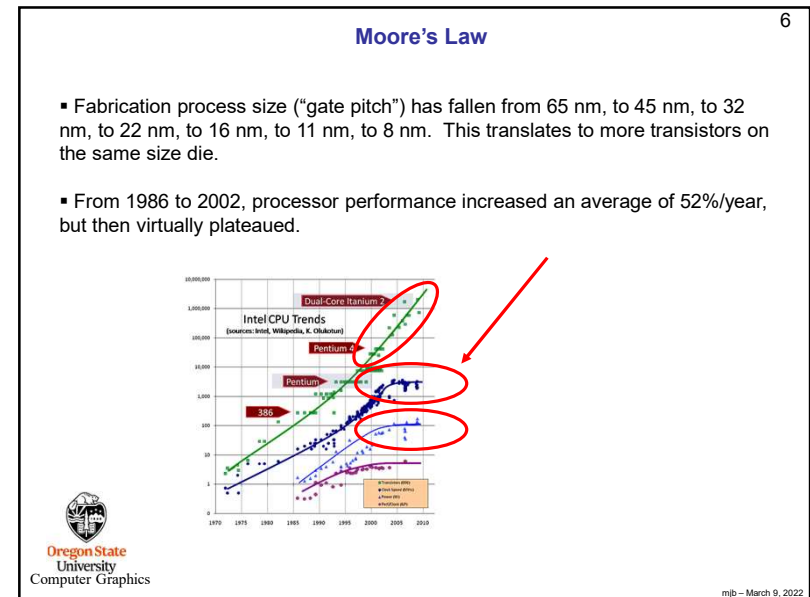
3



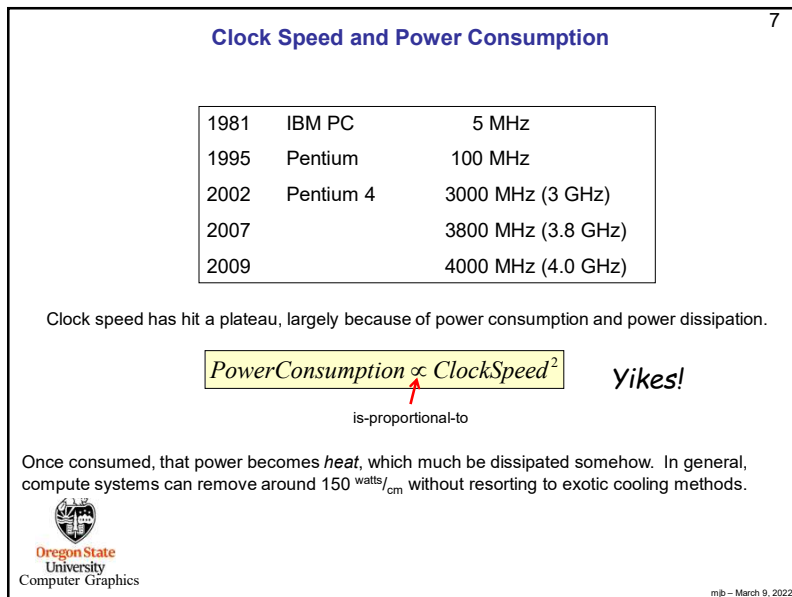
4



5



6



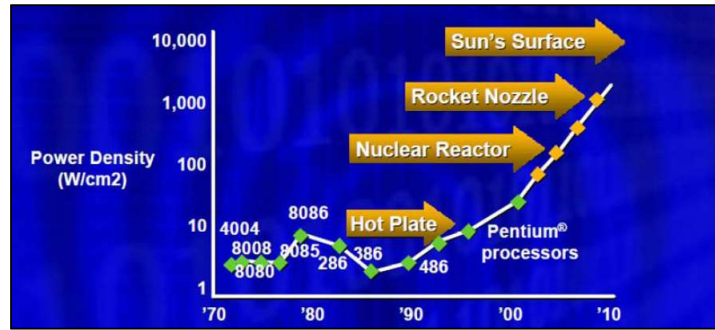
7



8

What Kind of Power Density Dissipation Would it Have Taken to Keep up with Clock Speed Trends?

9



Source: Intel



Oregon State University
Computer Graphics

mjb - March 9, 2022

9

MultiCore -- Multiprocessing on a Single Chip

10

So, to summarize:

Moore's Law of transistor density is still going, but the "Moore's Law" of clock speed has hit a wall. Now what do we do?

We keep packing more and more transistors on a single chip, but don't increase the clock speed. Instead, we increase computational throughput by using those transistors to pack multiple processors onto the same chip.

This is referred to as **multicore**.



Oregon State University
Computer Graphics

Vendors have also reacted by adding SIMD floating-point units on the chip as well. We will get to that later.

mjb - March 9, 2022

10

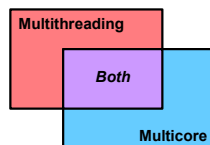
MultiCore and Multithreading

11

Multicore, even without multithreading too, is still a good thing. It can be used, for example, to allow multiple programs on a desktop system to always be executing concurrently.

Multithreading, even without multicore too, is still a good thing. Threads can make it easier to logically have many things going on in your program at a time, and can absorb the dead-time of other threads.

But, the big gain in performance is to use *both* to speed up a *single program*. For this, we need a **combination of both multicore and multithreading**.



Multicore is a very hot topic these days. It would be hard to buy a CPU that doesn't have more than one core. We, as programmers, get to take advantage of that.

We need to be prepared to convert our programs to run on **MultiThreaded Shared Memory Multicore** architectures.



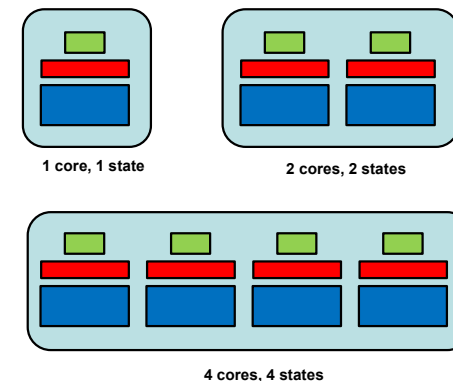
Oregon State University
Computer Graphics

mjb - March 9, 2022

11

Each of the Multiple Cores keeps its own State

12



Oregon State University
Computer Graphics

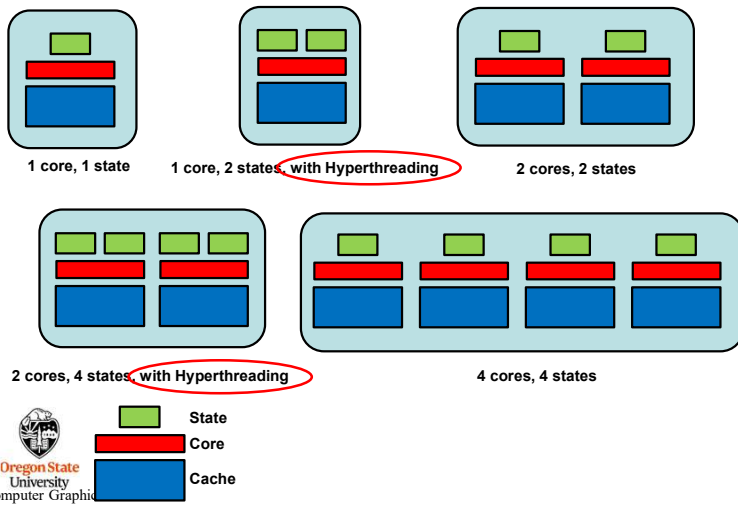
- State
 - Core
 - Cache
- Registers
 - Program Counter
 - Stack Pointer

mjb - March 9, 2022

12

So, if that's what Multicore is about, what is *Hyperthreading*?

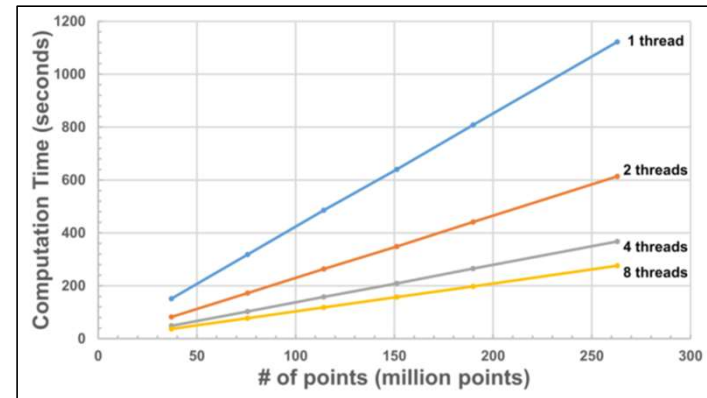
13



13

Four Cores with Two Hyperthreads per Core

14



Source: Erzhuo Che



Note that this is upside-down from our usual convention. Sorry. I got this from someone else.



mjb - March 9, 2022

14