Why Parallelism?

Thursday, March 26, 2020 5:56 PM

A parallel computer is a collection of processing elements that cooperate to solve problems quickly

We care about performance and efficiency

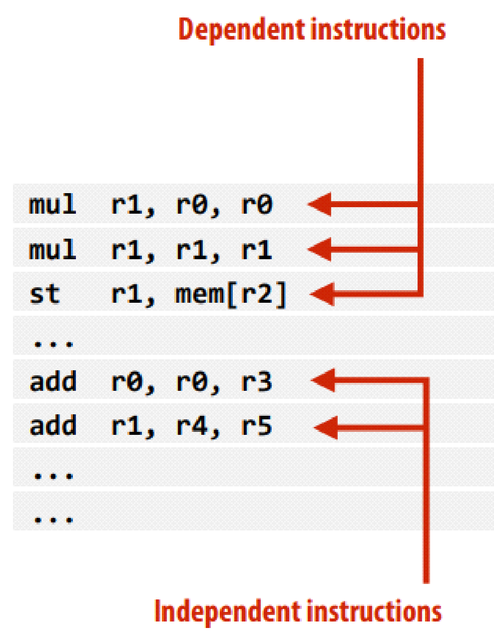
We're going to use multiple processors to get it

Speedup	One major motivation of using parallel processing: achieve a speedup

For a given problem:

$$\text{Speedup(using P processors)} = \frac{\text{execution time(using 1 processor)}}{\text{execution time(using P processors)}}$$

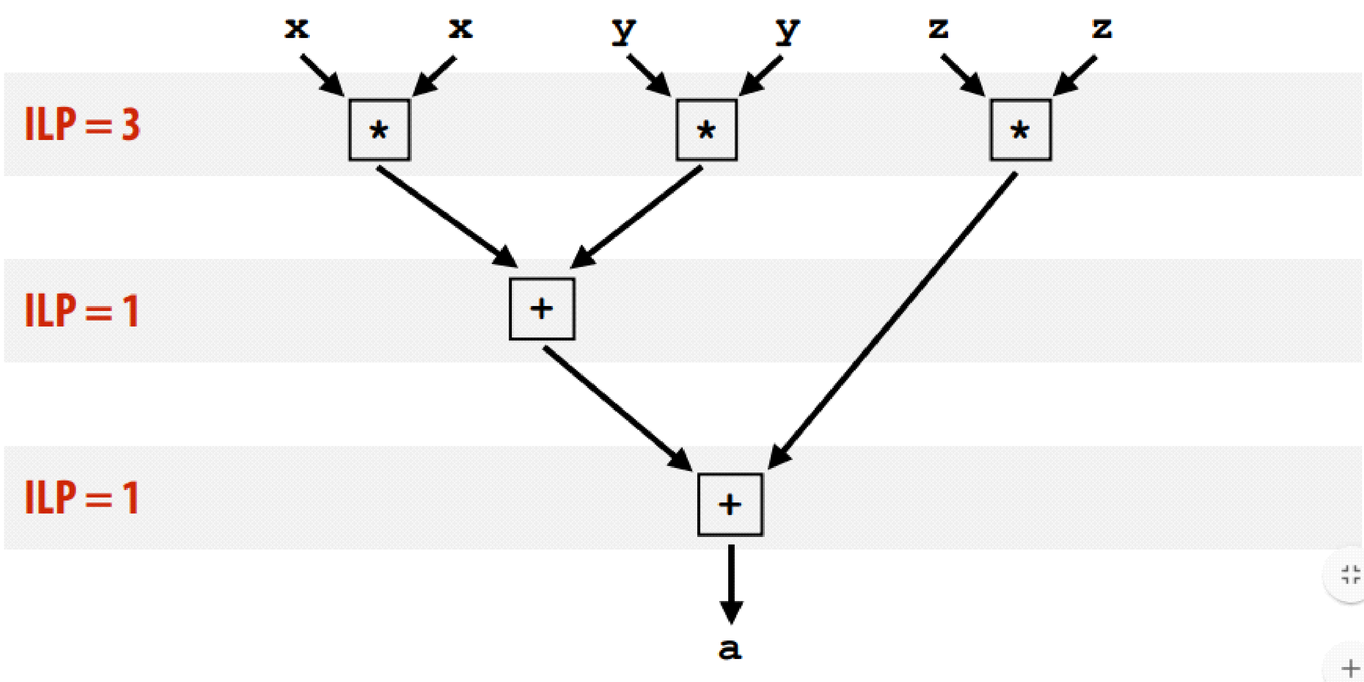
- **Instruction level parallelism (ILP)**
  - **Idea: Instructions must appear to be executed in program order. BUT independent instructions can be executed simultaneously by a processor without impacting program correctness**
  - **Superscalar execution: processor dynamically finds independent instructions in an instruction sequence and executes them in parallel**



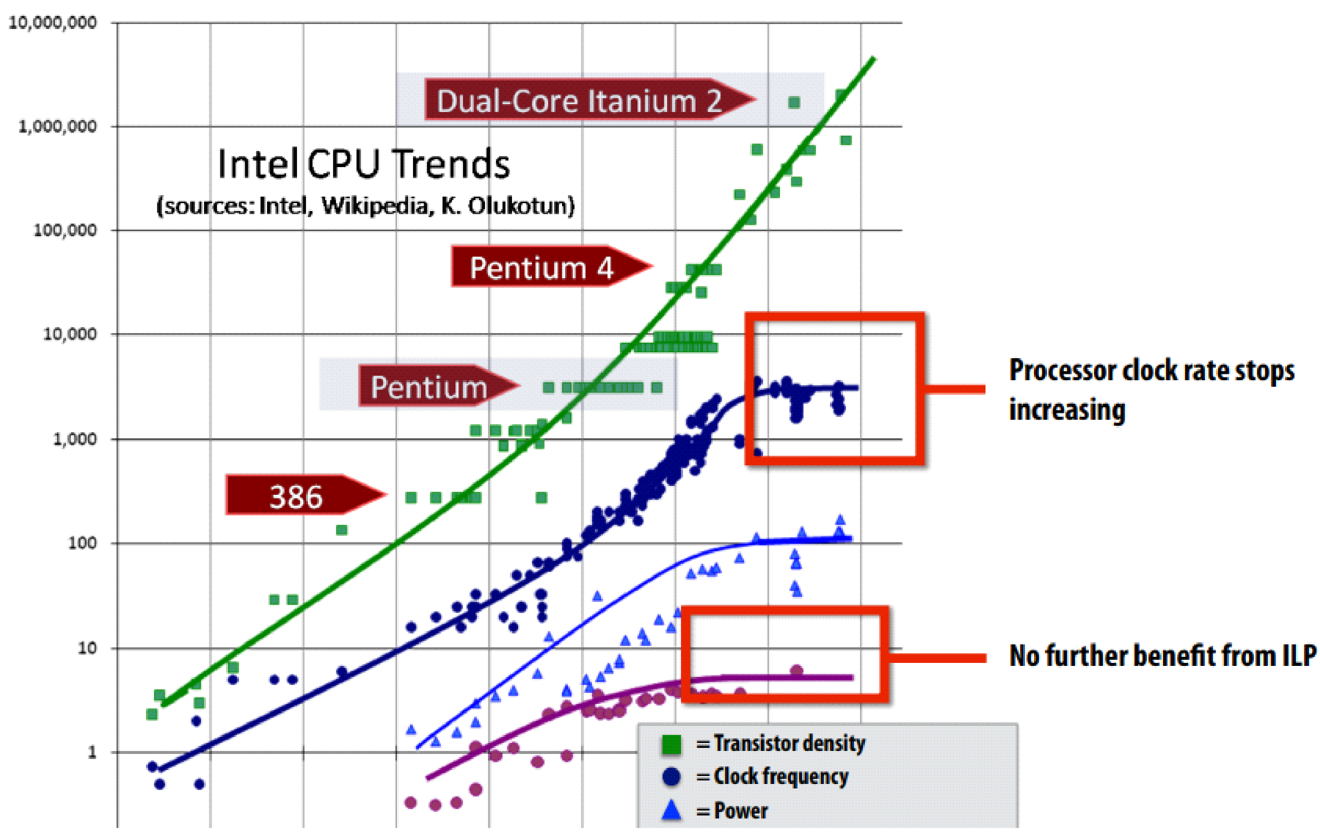
Screen clipping taken: 3/26/2020 7:27 PM

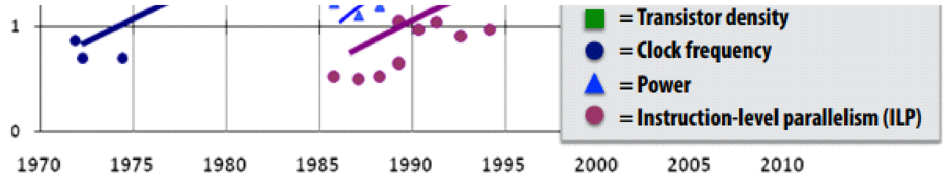
ILP example

```
a = x*x + y*y + z*z
```



Screen clipping taken: 3/26/2020 7:28 PM





Screen clipping taken: 3/26/2020 7:30 PM

The "power wall"

Power consumed by a transistor:

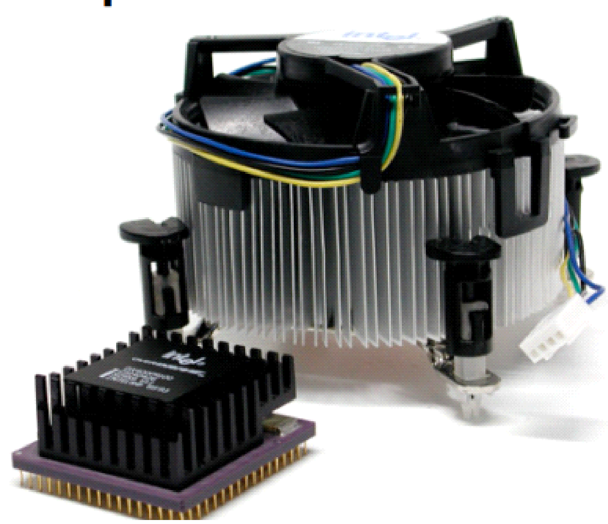
Dynamic power $\propto$ capacitive load $\times$ voltage^2 $\times$ frequency

Static power: transistors burn power even when inactive due to leakage

High power = high heat

Power is a critical design constraint in modern processors

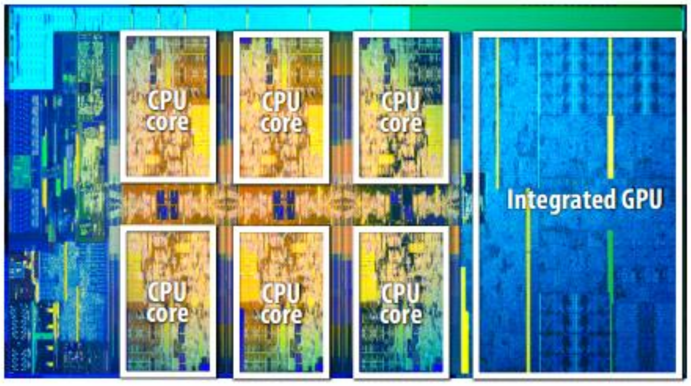
	TDP
Intel Core i7 (in this laptop):	45W
Intel Core i7 2700K (fast desktop CPU):	95W
NVIDIA Titan V GPU	250W
Mobile phone processor	1/2 - 2W
World's fastest supercomputer	megawatts
Standard microwave oven	700W



Screen clipping taken: 3/26/2020 7:34 PM

Intel Coffee Lake (2017) (aka "8th generation Core i7")

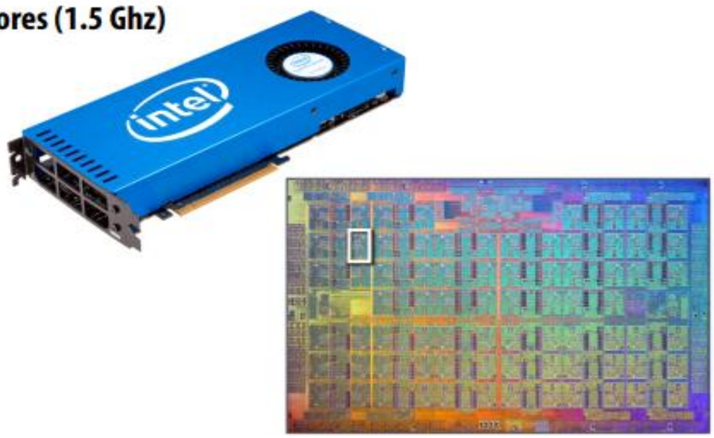
Six-core CPU + multi-core GPU integrated on one chip



Screen clipping taken: 3/26/2020 7:41 PM

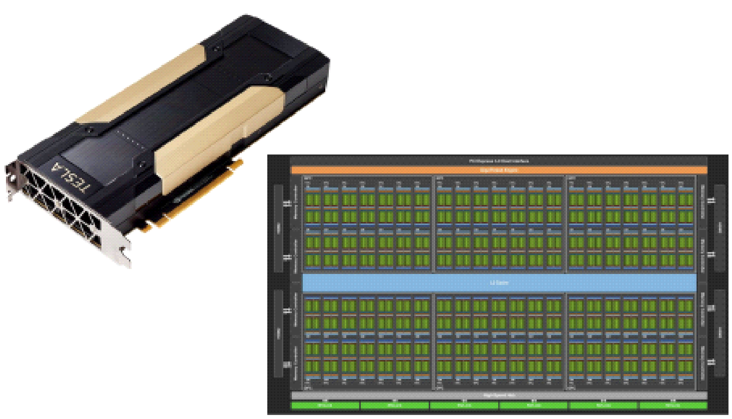
Intel Xeon Phi 7290 (2016)

72 cores (1.5 Ghz)



NVIDIA Tesla V100 GPU (2017)

5,376 fp32 units grouped into 84 major processing blocks

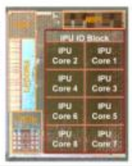


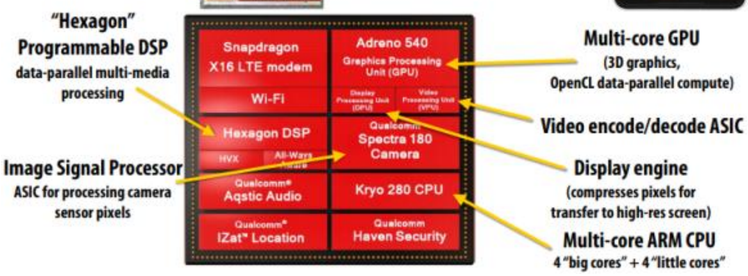
Supercomputing

- Today: combinations of multi-core CPUs + GPUs
- Oak Ridge National Laboratory: Summit (#1 supercomputer in world)
  - 9,216 x 22-core IBM Power9 CPUs + 27,648 NVIDIA Volta GPUs

Let's crack open a modern smartphone

Google Pixel 2 Phone:
Qualcomm Snapdragon 835 SoC + Google Visual Pixel Core





Specialized processors for evaluating deep networks

Countless recent papers at top computer architecture research conferences on the topic of ASICs or accelerators for deep learning or evaluating deep networks…

- Cambricon: an instruction set architecture for neural networks, Liu et al. ISCA 2016
- EIE: Efficient Inference Engine on Compressed Deep Neural Network, Han et al. ISCA 2016
- Cnvlutin: Ineffectual-Neuron-Free Deep Neural Network Computing, Albericio et al. ISCA 2016
- Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators, Reagen et al. ISCA 2016
- vDNN: Virtualized Deep Neural Networks for Scalable, Memory-Efficient Neural Network Design, Rhu et al. MICRO 2016
- Fused-Layer CNN Architectures, Alwani et al. MICRO 2016
- Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Network, Chen et al. ISCA 2016
- PRIME: A Novel Processing-in-memory Architecture for Neural Network Computation in ReRAM-based Main Memory, Chi et al. ISCA 2016
- DNNWEAVER: From High-Level Deep Network Models to FPGA Acceleration, Sharma et al. MICRO 2016

Intel Lake Crest ML accelerator
(formerly Nervana)

Summary

- Today, single-thread-of-control performance is improving very slowly
  - To run programs signifcantly faster, programs must utilize multiple processing elements
  - Which means you need to know how to write parallel code

- Writing parallel programs can be challenging
    - Requires problem partitioning, communication, synchronization
    - Knowledge of machine characteristics is important