Parallelism in Hardware

Minsoo Ryu

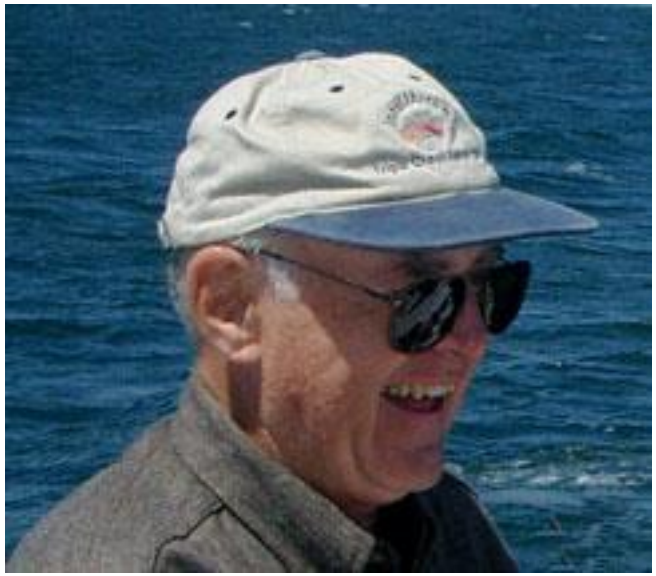
Department of Computer Science and Engineering
Hanyang University

Outline

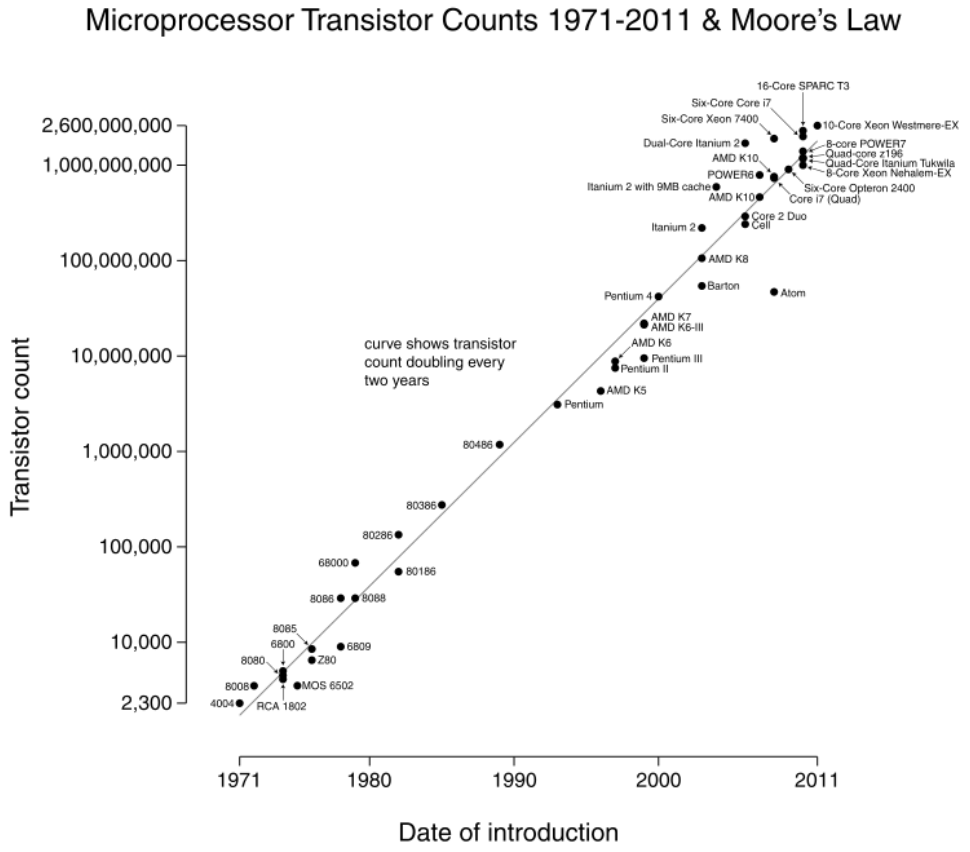
1. Advent of Multicore Hardware
2. Multicore Processors
3. Amdahl's Law
4. Parallelism in Hardware
5. Q & A

Moore's Law

- **The number of transistors on integrated circuits doubles approximately every two years**

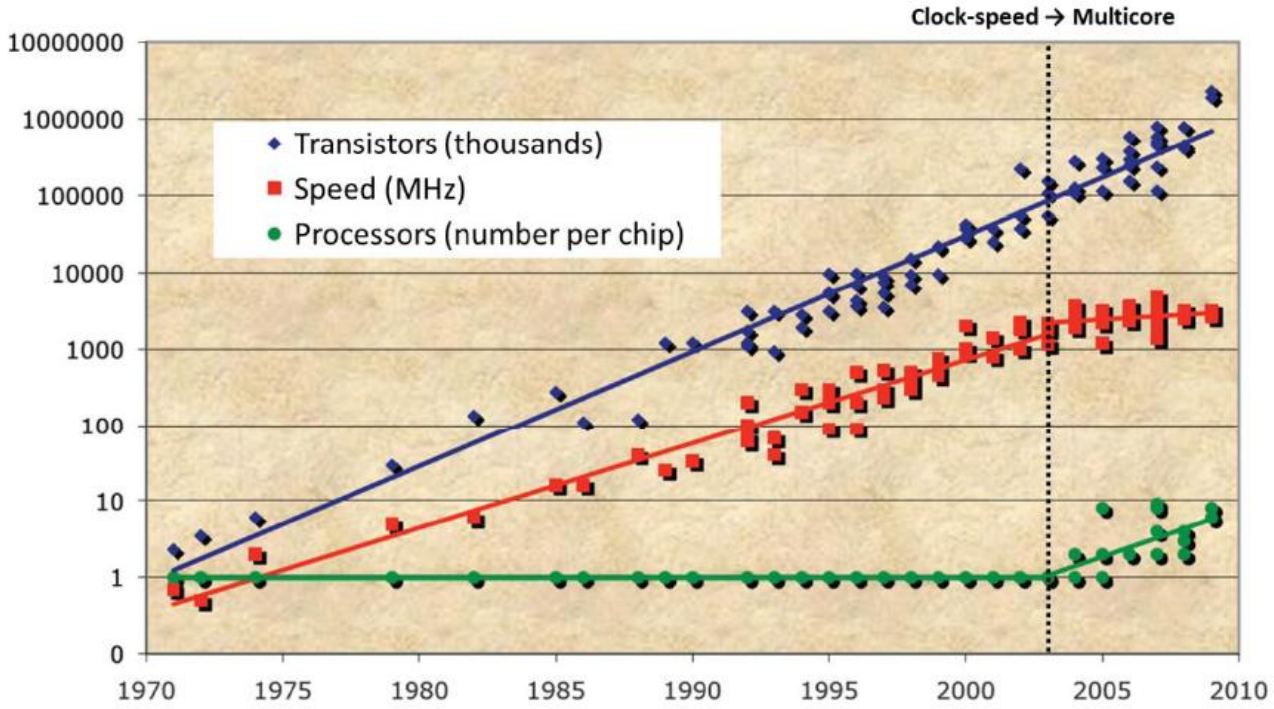


Gordon Earle Moore (1929 ~)



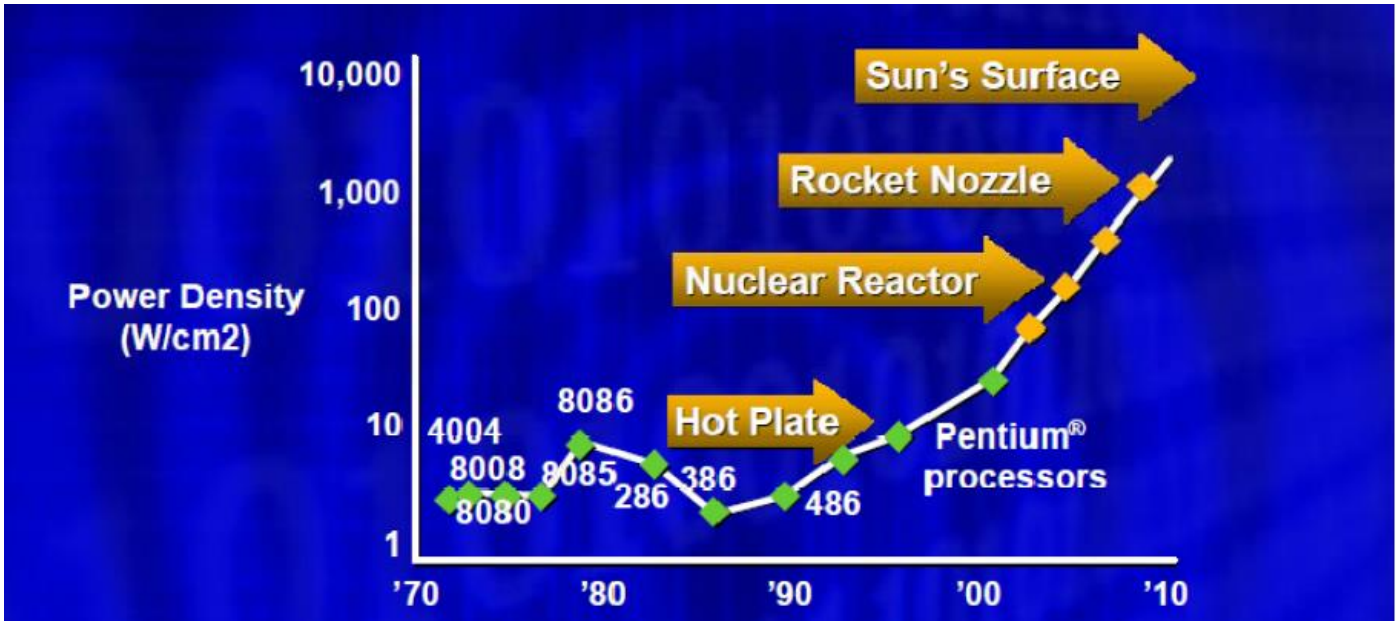
End of Frequency Scaling

- **Frequency scaling ended in 2004**
  - **Intel cancelled the Tejas and Jayhawk projects that aimed at 7 GHz or higher**

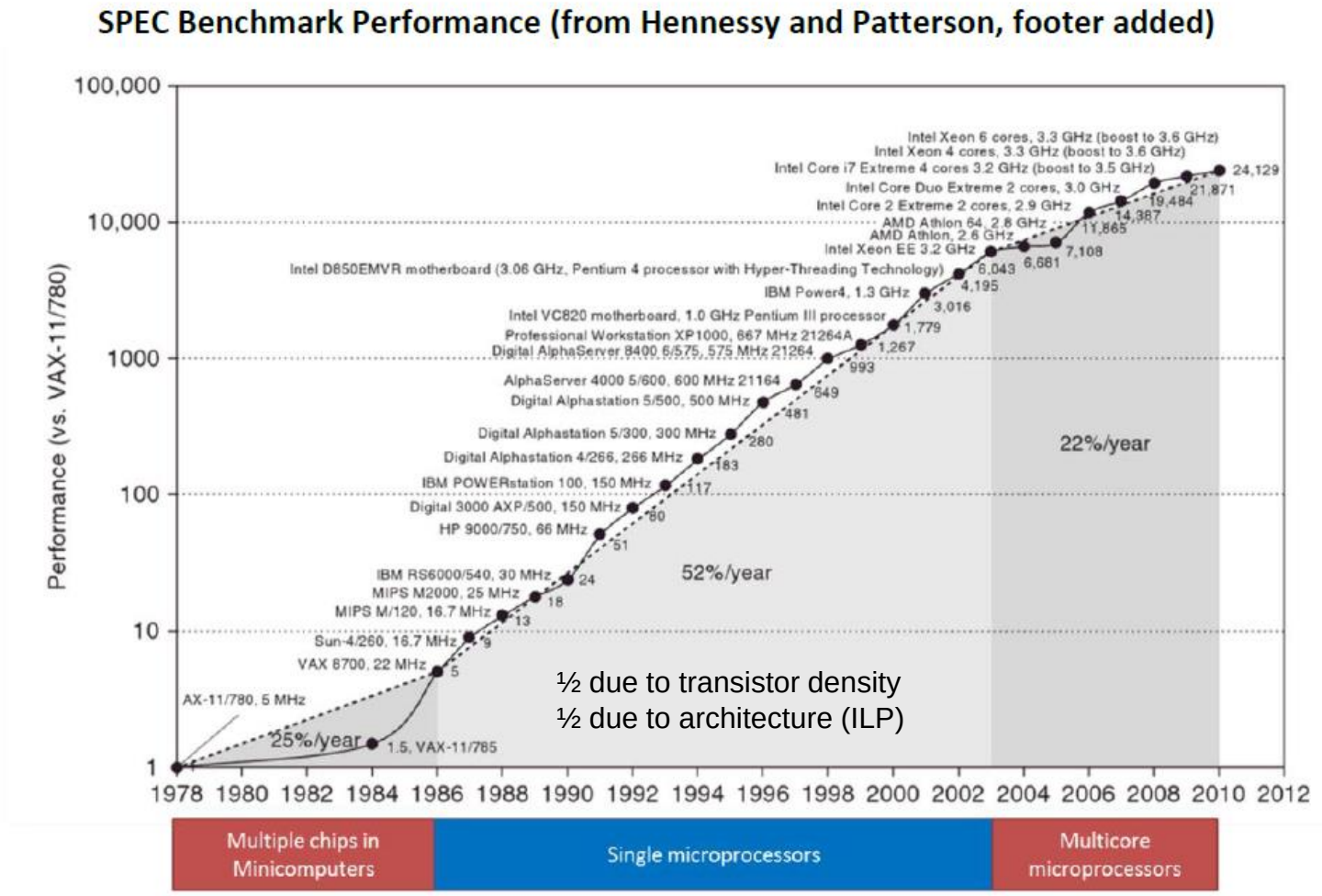


Power Consumption and Heat

- Heat problems due to the extreme power consumption of the core (power wall)
  - P (power) = C × V^2 × F

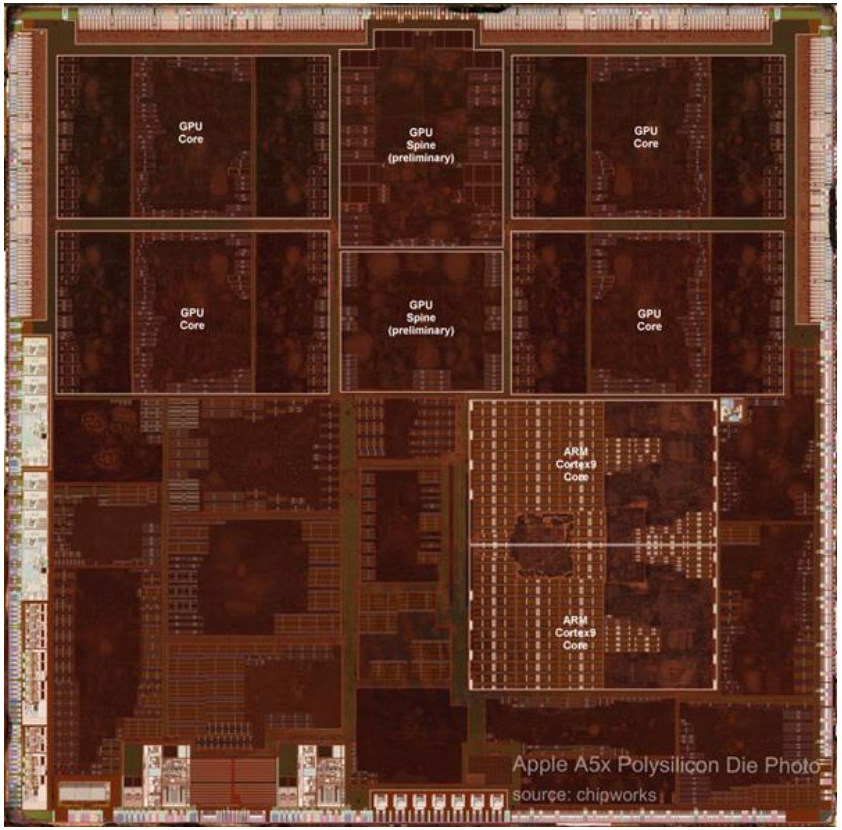


Slow-Down of Performance Gain

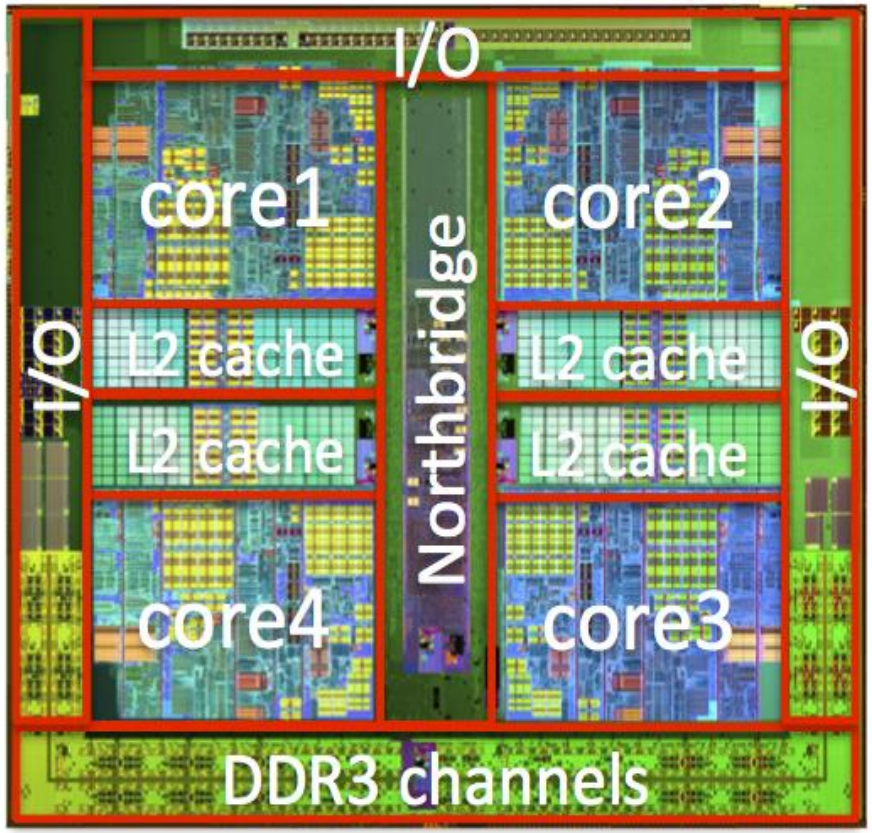


Source: Computer Architecture, A Quantitative Approach by Hennessy and Patterson

Multicore Processors #1

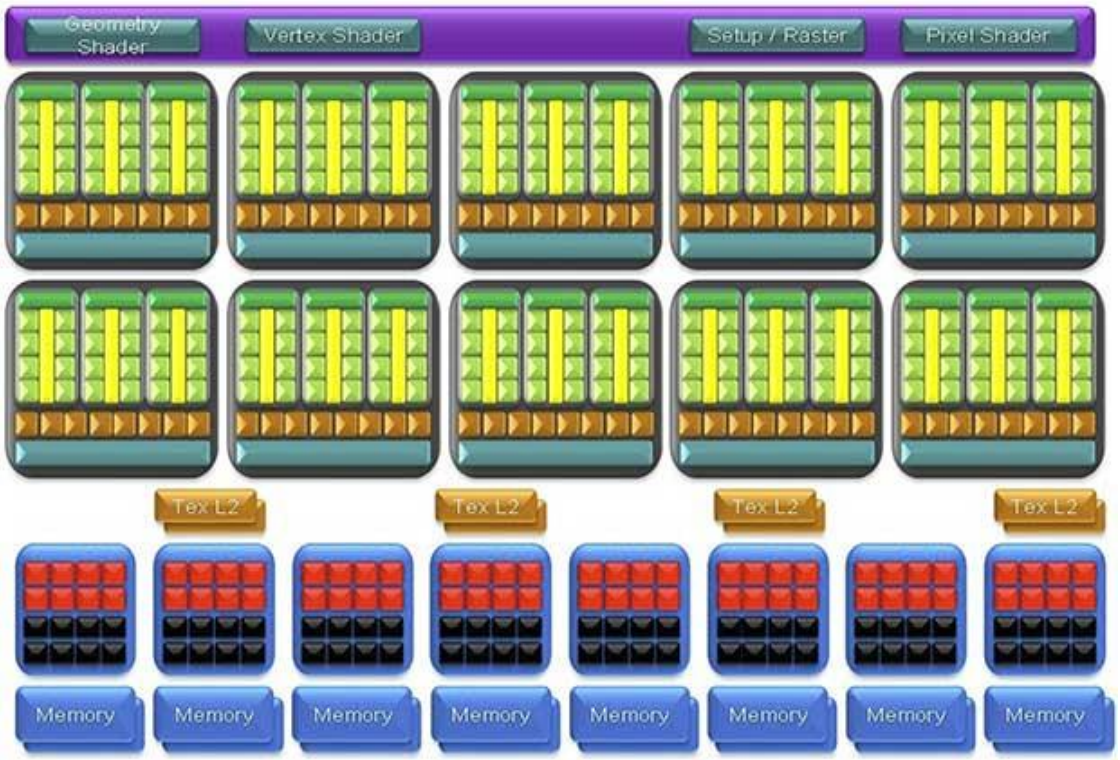


Apple A5x with 4 CPU cores

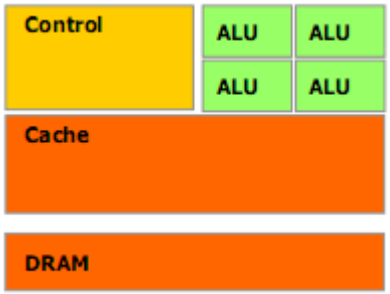


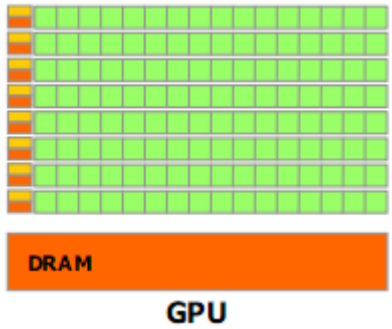
AMD Athlon II x4 Quad-core Processor

Multicore Processors #2



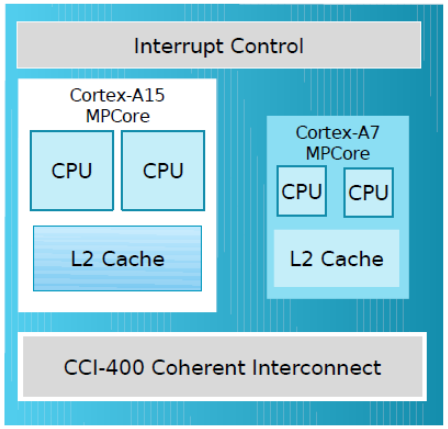
NVIDIA GeForce GTX 280



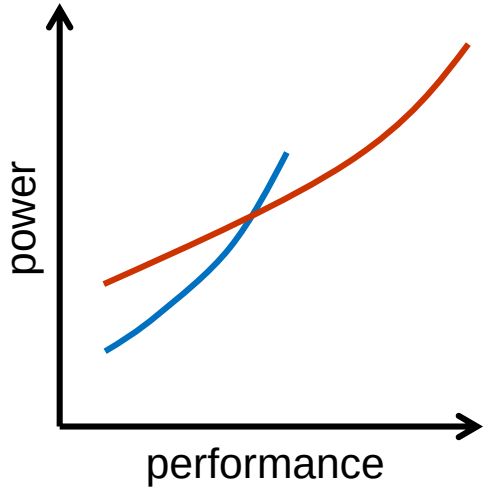


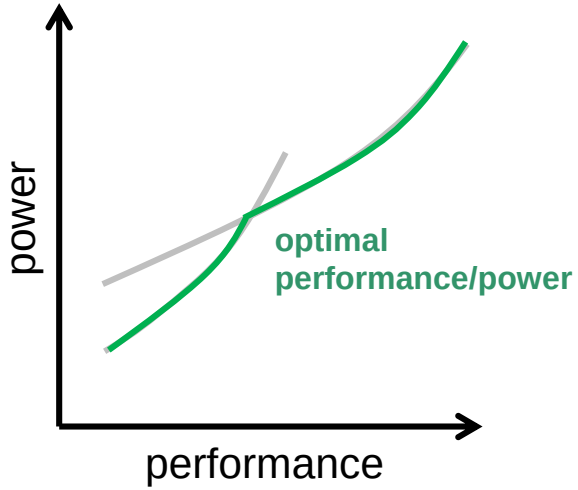
Multicore Processors #3

- ARM big.LITTLE architecture
  - Two clusters on one chip
  - Cortex-A7 for power efficiency
  - Cortex-A15 for high performance



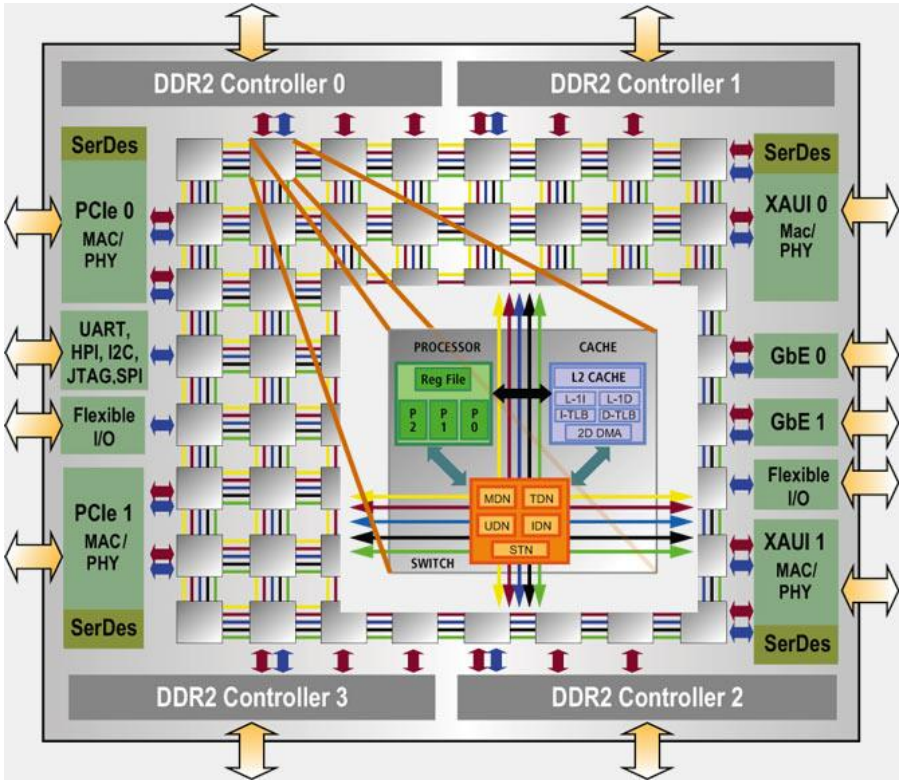
- Why heterogeneity?





Multicore Processors #4

- **Tilera Tile-Gx 72-core processor**
  - **Tilera founded by prof. Anant Agarwal at MIT in 2004**
  - **Tiled architecture**
  - **Mesh network**
  - **Each core has**
    - a processor
    - cache
    - switch



Reasons for Multicore in Cars

- **More computing power and less power dissipation**
  - Similar die size, same or lower CPU clock frequency, parallel processing
  - Lower CPU clock frequency, computing power on demand, i.e. sleep modes for unused cores
- **Separation of Applications**
  - Specialized cores (FPP, DSP)
  - Various operating systems on the same ECU, e.g. OSEK and Linux
  - Legal reasons, e.g. software originating from multiple suppliers
- **Functional Safety**
  - Avoidance of mutual interference → running application software components on separated cores
  - Redundancy → identical application running on 2 cores (lockstep mode)

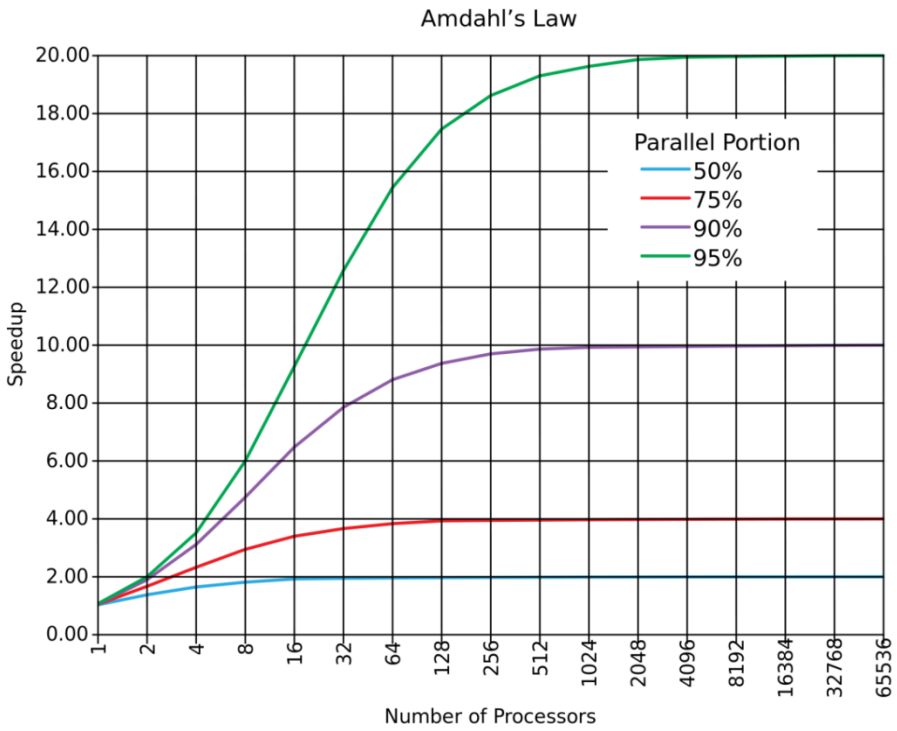
Amdahl's Law

- Speedup $= \frac{1}{(1-p)+p/n}$
  - p: parallel portion, n: # of CPUs

sequential part

parallel part

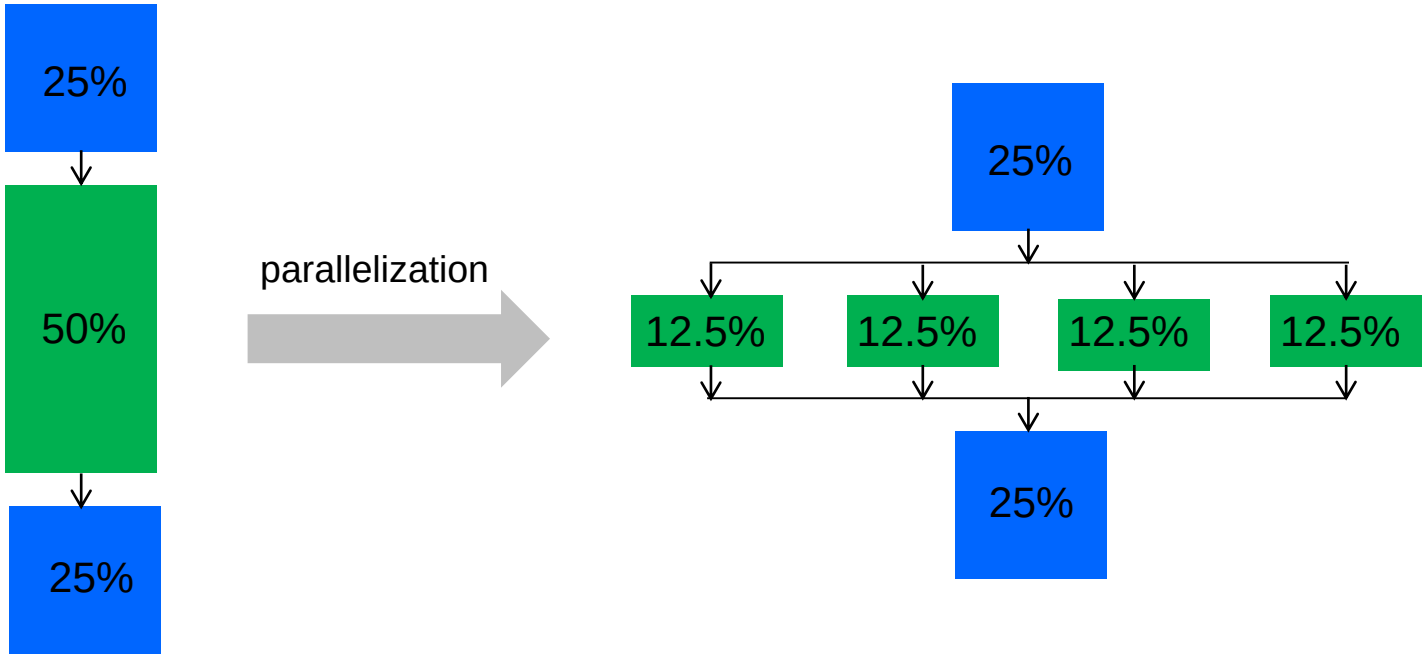
```
a = b + c;
d = a + 1;
e = d + a;
for (i=0; i<e; i++)
    M[i] = 1;
```



Amdahl's Law

Example

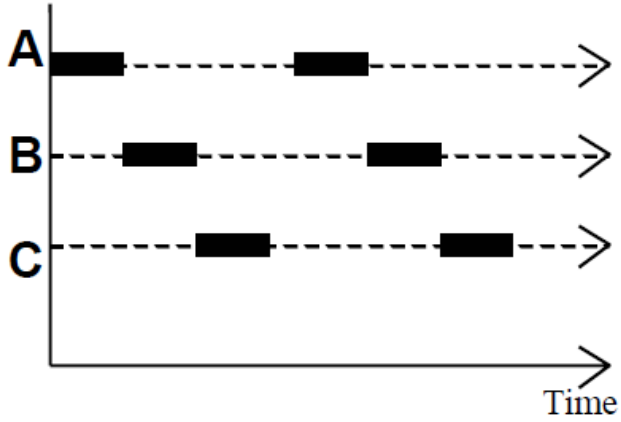
- $p = 0.5, n = 4 \rightarrow \frac{1}{(1-0.5)+0.5/4} = \frac{1}{0.625} = 1.6$



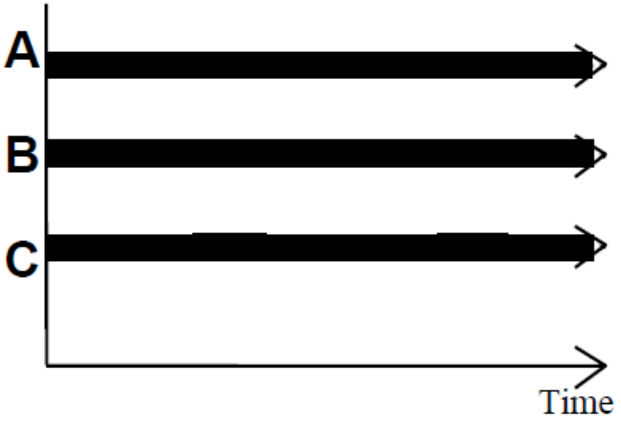
Concurrency and Parallelism

- Concurrency is not (only) parallelism

- Interleaved Concurrency
  - Logically simultaneous processing
  - Interleaved execution on a single processor



- Parallelism
  - Physically simultaneous processing
  - Requires a multiprocessor system or multicore system



Types of Parallelism

- **Parallelism in hardware**
  - **Pipelining**
  - **Superscalar, VLIW**
  - **SIMD processing**
  - **HW multithreading**

} instruction level parallelism (ILP)
} data level parallelism (DLP)
} thread level parallelism (TLP)

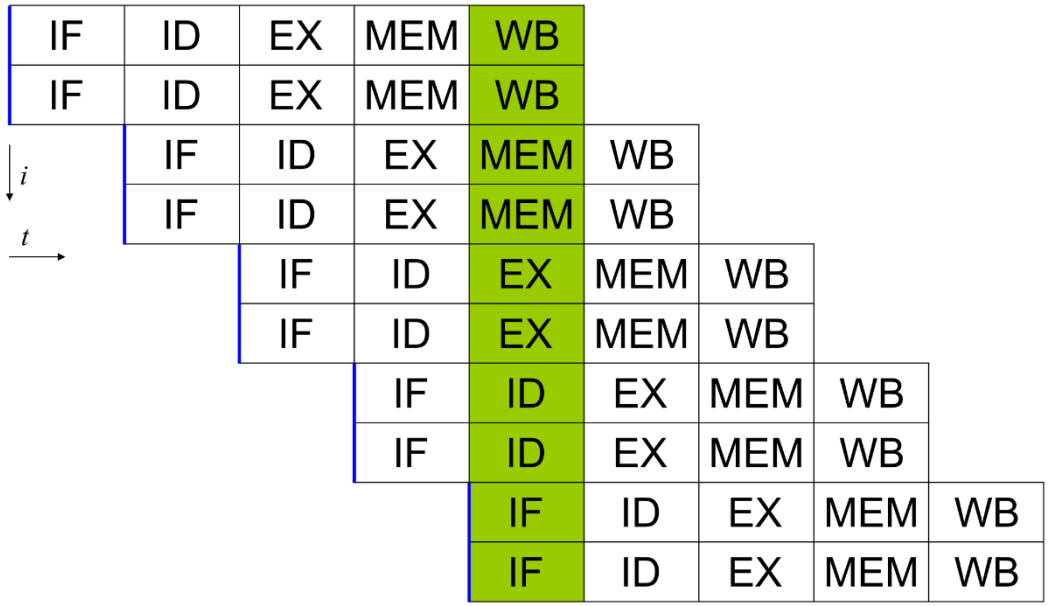
Parallelism in HW: Pipelining

- The basic instruction cycle is broken up into a series called a pipeline
- Each instruction is split up into a sequence of steps and executed in parallel (at the same time)

Inst r No.	Pipeline Stage						
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

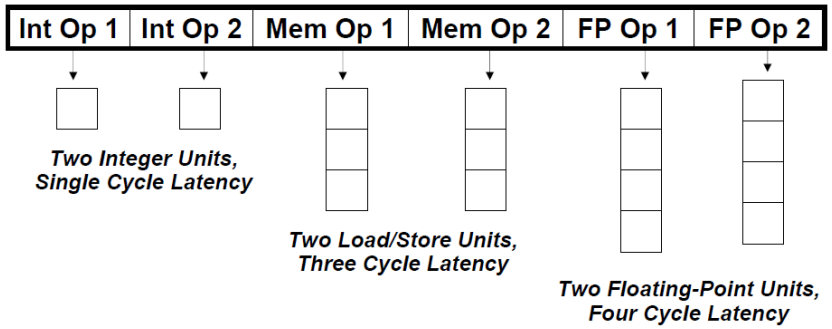
Parallelism in HW: Superscalar

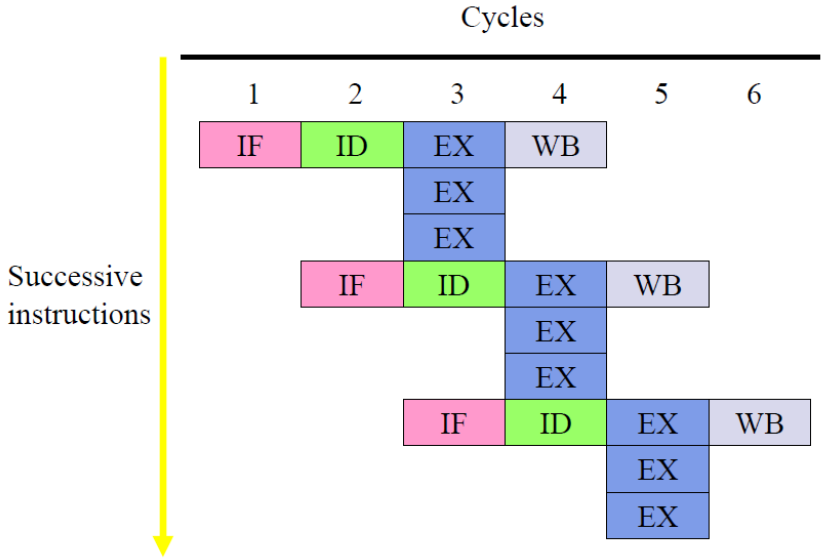
- A superscalar processor executes more than one instruction during a clock cycle
  - Using an execution resource within a single CPU such as an arithmetic logic unit, a bit shifter, or a multiplier



Parallelism in HW: VLIW

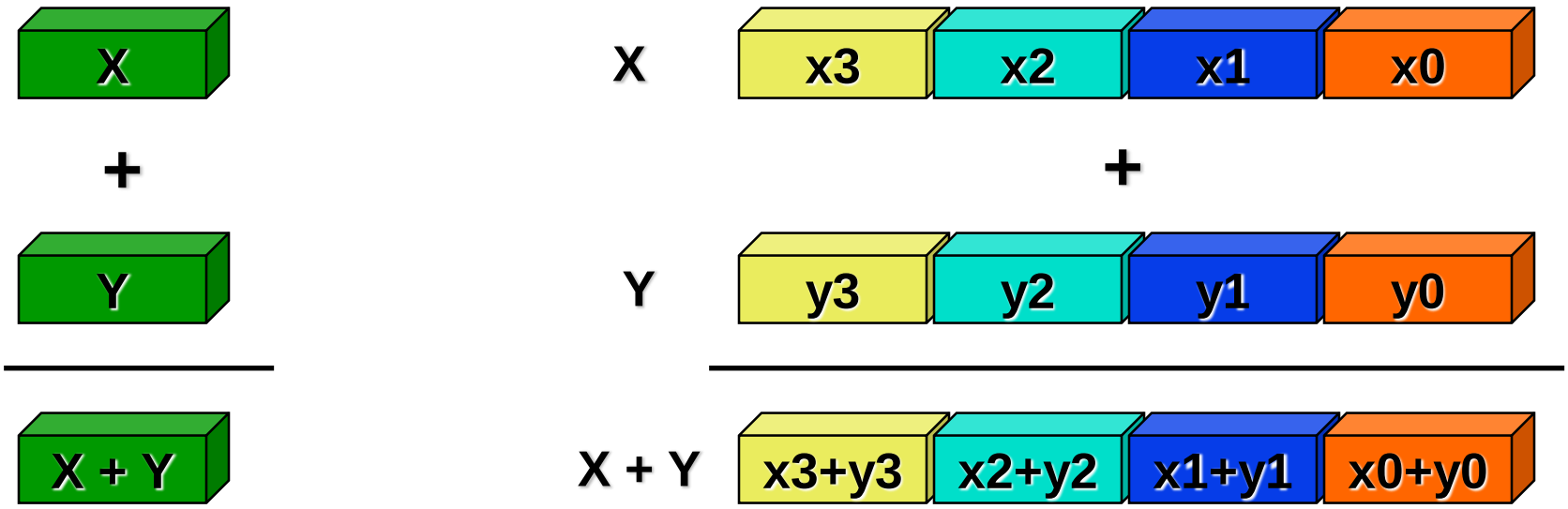
➢ VLIW (Very Long Instruction Word)





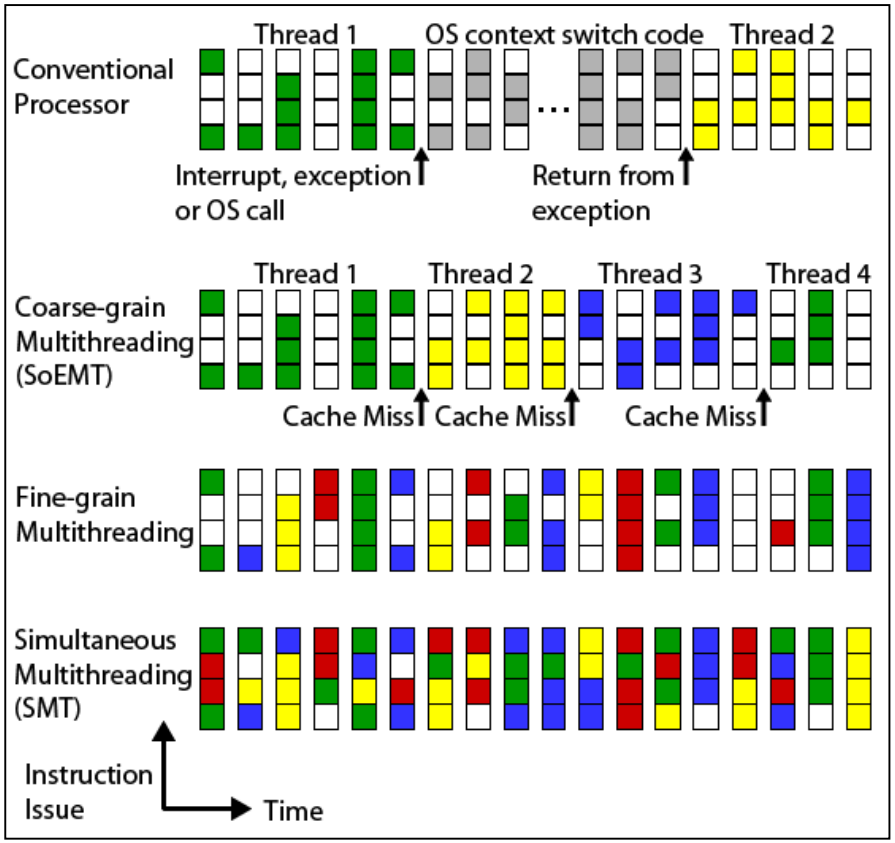
Parallelism in HW: SIMD Processing

- **Scalar processing**
  - One operation produces one result
- **SIMD processing**
  - One operation produces multiple results



Parallelism in HW: HW Multithreading

- The goal of hardware multithreading is to allow quick switching between threads



- Hardware registers need to be replicated such as the program counter

- Switching from one thread to another thread means the hardware switches from using one register set to another

- SMT exploits parallelism available across multiple threads to decrease the waste associated with unused issue slots


Q&A