中山大學
SUN YAT-SEN UNIVERSITY

国家超级计算广州中心
NATIONAL SUPERCOMPUTER CENTER IN GUANGZHOU

Computer Architecture

计 算 机 体 系 结 构

第1讲：量化设计分析（1）

张献伟

xianweiz.github.io

DCS3013, 9/7/2022

News: GPU Ban[禁令]

- US bans Nvidia and AMD from shipping high-performance GPU chips to China

Item 8.01 Other Events

On August 26, 2022, the U.S. government, or USG, informed NVIDIA Corporation, or the Company, that the USG has imposed a new license requirement, effective immediately, for any future export to China (including Hong Kong) and Russia of the Company's A100 and forthcoming H100 integrated circuits. DGX or any other systems which incorporate A100 or H100 integrated circuits and the A100X are also covered by the new license requirement. The license requirement also includes any future NVIDIA integrated circuit achieving both peak performance and chip-to-chip I/O performance equal to or greater than thresholds that are roughly equivalent to the A100, as well as any system that includes those circuits. A license is required to export technology to support or develop covered products. The USG indicated that the new license requirement will address the risk that the covered products may be used in, or diverted to, a 'military end use' or 'military end user' in China and Russia. The Company does not sell products to customers in Russia.

The new license requirement may impact the Company's ability to complete its development of H100 in a timely manner or support existing customers of A100 and may require the Company to transition certain operations out of China. The Company is engaged with the USG and is seeking exemptions for the Company's internal development and support activities.

In addition, the Company is engaging with customers in China and is seeking to satisfy their planned or future purchases of the Company's Data Center products with products not subject to the new license requirement. To the extent that a customer requires products covered by the new license requirement, the Company may seek a license for the customer but has no assurance that the USG will grant any exemptions or licenses for any customer, or that the USG will act on them in a timely manner.

The Company's outlook for its third fiscal quarter provided on August 24, 2022 included approximately $400 million in potential sales to China which may be subject to the new license requirement if customers do not want to purchase the Company's alternative product offerings or if the USG does not grant licenses in a timely manner or denies licenses to significant customers.

https://www.sec.gov/ix?doc=/Archives/edgar/data/1045810/000104581022000146/nvda-20220826.htm

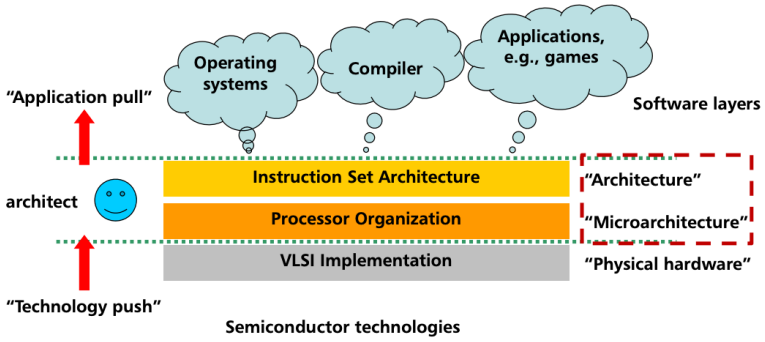
About

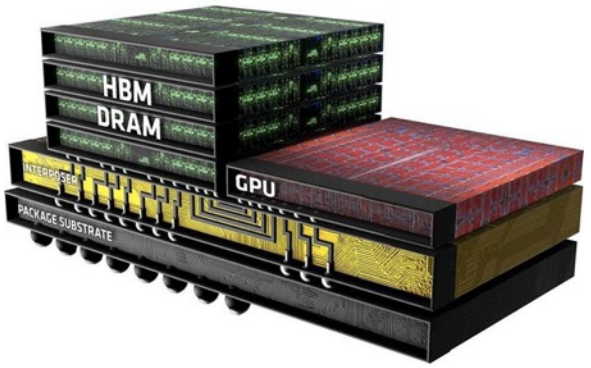
- 选课（94人）
  - 计科-48，网安-23，信计-10，保密-13
- 课件
  - xianweiz.github.io > Teaching > DCS3013-Computer Architecture
    - https://xianweiz.github.io/teach/dcs3013/f2022.html
- **课程QQ群：419 655 749**
- 超算习堂：https://easyhpc.net/course/157

- 助教
  - 黄亮鸿/理论 (huanglh59@mail2.sysu.edu.cn)
  - 郭天宇/实践 (guoty9@mail2.sysu.edu.cn)
  - 顾宇浩/实践 (guyh9@mail2.sysu.edu.cn)

Review

- What is Computer Architecture?
  - To make design trade-offs across the hw/sw interface to meet functional, performance and cost requirements
- Why study Computer Architecture?
  - Understand how computer works, design concepts and perf …
  - In a golden age
- How? Use software to improve hardware
  - You are not directly working on hardware
  - Instead, you use software to design/optimize hardware
  - C/C++, Python, Bash



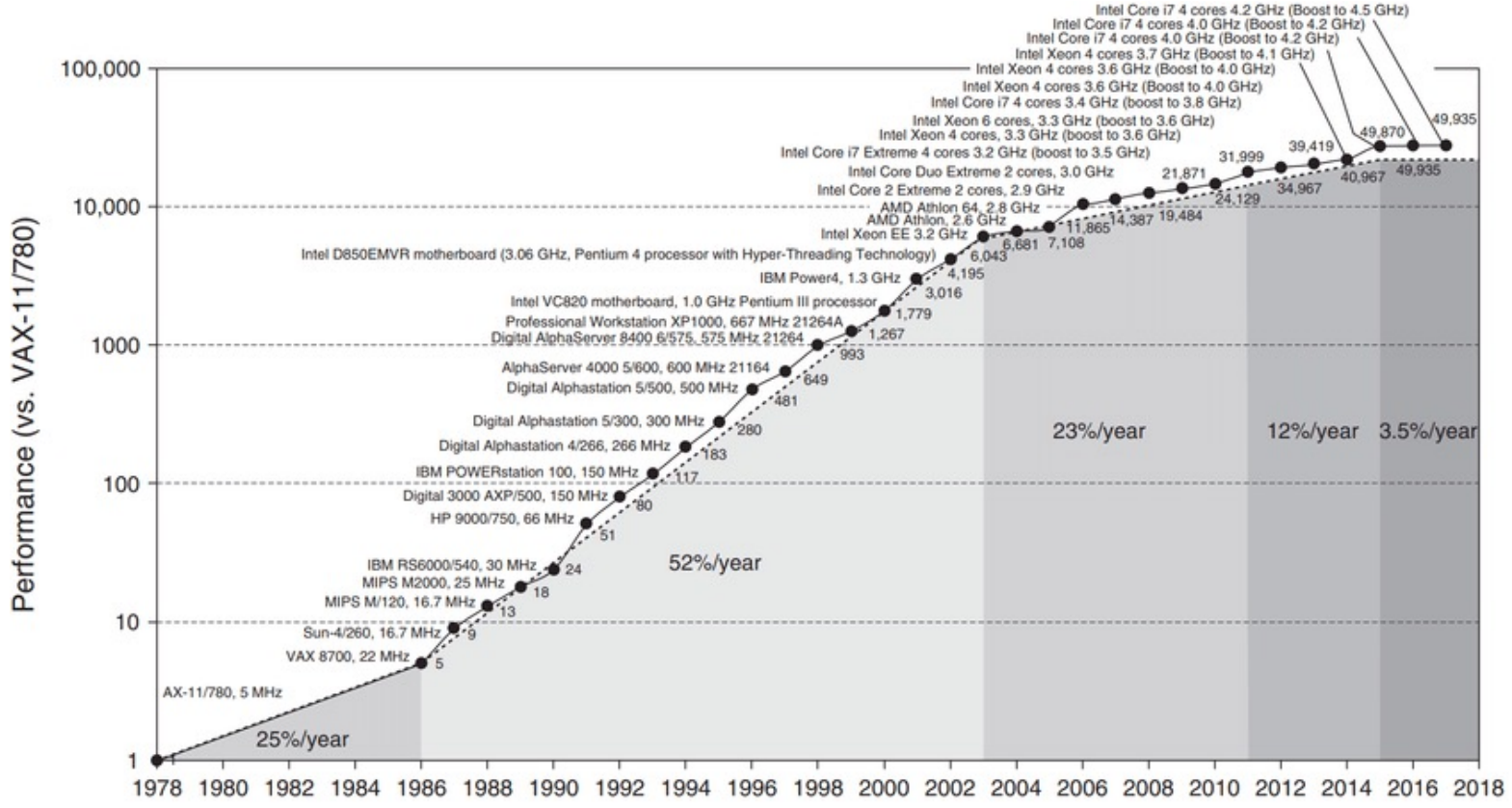


Role of [Computer] Architect[职责]

- Look backward (to the past)
  - Understand tradeoffs and designs, upsides/downsides, past workloads. Analyze and evaluate the past.
- Look forward (to the future)
  - Be the dreamer and create new designs. Listen to dreamers.
  - Push the state of the art. Evaluate new design choices.
- Look up (towards problems in the computing stack)
  - Understand important problems and their nature.
  - Develop architectures and ideas to solve important problems.
- Look down (towards device/circuit technology)
  - Understand the capabilities of the underlying technology.
  - Predict and adapt to the future of technology (you are designing for *N* years ahead). Enable the future technology.

Technology Improvement[技术提升]

- Computer technology has greatly improved
  - A $500 cellphone today outperforms the fastest supercomputer in 1993 ($50 million)
  - Improvement from both advances in the tech used to build computers and from innovations in computer design

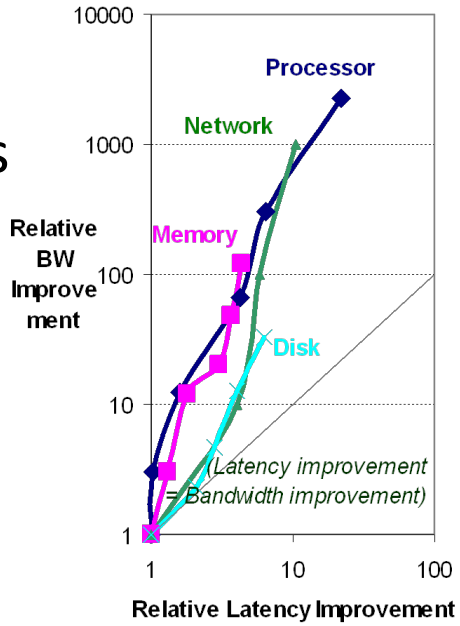


Trends in Technology (§1.4)[技术趋势]

- Integrated circuit (IC) logic[集成电路]
  - Transistor density: +35%/year (feature size decreases)
  - Die size: +10-20%/year
  - Integration overall: +40-55%/year (Moore's Law)
- DRAM capacity: +25-40%/year (growth is slowing)[内存]
  - Memory usage quadruples every three years
- Flash capacity: +50-60%/year[闪存]
  - 8-10X cheaper/bit than DRAM
- Magnetic disk: +40%/year[磁盘]
  - 8-10X cheaper/bit then Flash and 200-300X cheaper/bit than DRAM
- Network[网络]

Performance Trends[性能趋势]

- Bandwidth or throughput[带宽/吞吐]
  - Total work done in a given time
  - 32,000 - 40,000X improvement for processors
  - 400 - 2400X improvement for memory and disks
- Latency or response time[时延/响应时间]
  - Time between start and completion of an event
  - 50 - 90X improvement for processors
  - 8 - 9X improvement for memory and disks
    - Memory wall[内存墙]



- Latency lags bandwidth (in the last 30 years)
  - CPU: 20x vs. 2000x
  - Memory: 4x vs. 120x

Transistors and Wire[晶体管/线路]

- IC processes are characterized by feature size[特征尺寸]
- **Feature size**: minimum size of transistor or wire
  - 10um in 1971 to 22nm in 2012 to 7nm in 2017 to 5nm in 2020 (3nm is being developed)
- Moore's Law: aka "technology scaling"[缩放]
  - Literally: density (transistors/area) doubles every 18 months
  - Public interpretation: performance doubles every 18 months
  - Continued miniaturization (esp. reduction in channel length)
    - + Improves switching speed, power/transistor, area(cost)/transistor
    - – Reduces transistor reliability

Intel 7: 10nm

Intel 4: 7nm

Intel: 7+

Intel 20A: 5nm

Power and Energy (§1.5)[功耗/能耗]

- **Energy** is a biggest challenge facing computer design
  - Bring power in with 100s of pins
  - Power is dissipated as heat and must be removed
- **Power/energy** are increasingly important
  - Battery life for mobile devices[电池续航]
    - Laptops, phones, cameras
  - Tolerable temperature for devices without active cooling[温度]
    - Power means temperature, active cooling means cost
    - No room for a fan in a cell phone, no market for a hot cell phone
  - Electric bill for compute/data centers[电费]
    - Pay for power twice: once in, once out (to cool)
  - Environmental concerns[环保]
    - “Computers” account for growing fraction of energy consumption

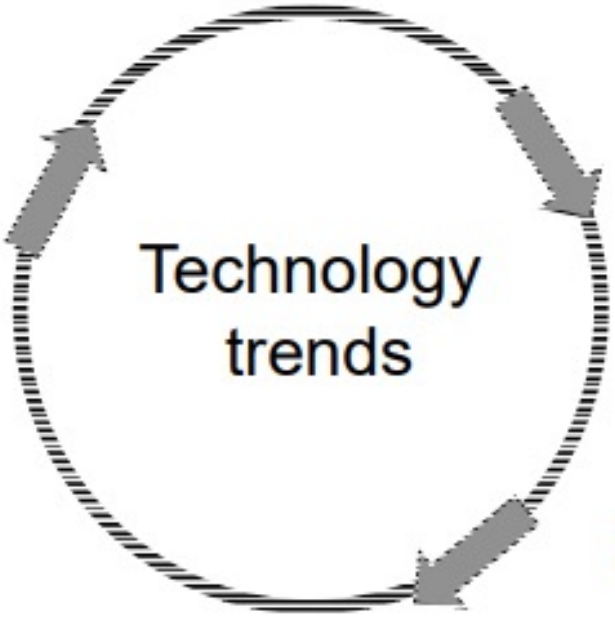
Power and Energy (cont.)

- **Energy**: measured in Joules or Watt-seconds[焦耳]
  - Total amount of energy stored/used
  - Battery life, electric bill, environmental impact
- **Power**: energy per unit time (measured in Watts)[瓦特]
  - Joules per second
  - Power impacts power supply and cooling requirements (cost)
  - Peak power vs average power
- Two sources[来源]
  - Dynamic power: active switching of transistors
  - Static power: leakage of transistors even while inactive
- Calculation
  - Energy is proportional to Voltage^2
  - Power is proportional to (Voltage^2 x Frequency)

Example：天河2号

- 配置
  - 16000个节点，每个2*12-核CPU + 2 Matrix-2000 + 64GB内存
    - 系统峰值运算速度为每秒10.07亿亿次，持续速度每秒6.14亿亿次
- 峰值功耗：17.6MW （加散热系统20+MW）
  - 17.6MW x 24h x 365 = 1.5亿度电/年
    - 40万度/天 → 30万元电费/天
- 水冷散热
  - 机柜内循环水冷的模式：8℃进水， 21℃出水

Methodology: Design/Evaluation[方法]



Evaluate Existing Systems for Bottlenecks

分析当前

Simulate New Designs and Organizations

设计及评估

Implement Next Generation System

实现下一代

Design Goals[设计目标]

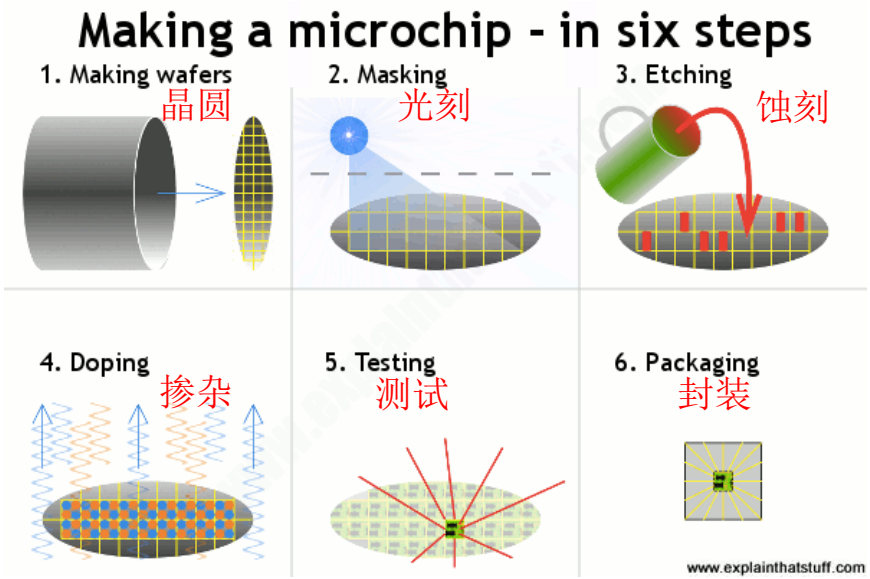
- Functional[功能性]
  - What functions should it support？
  - Needs to be correct
    - Unlike software, difficult to update once deployed
- Performant[性能]
  - “Fast” is only meaningful in the context of a set of important tasks
  - Not just “Gigahertz”
  - Impossible goal: fastest possible design for all programs
- Reliable[可靠性]
  - Does it continue to perform correctly?
  - Hard fault vs. transient fault
    - Example： memory errors and sun spots
  - Space satellites vs. desktop vs. server reliability

Design Goals (cont.)

- Low cost[低成本]
  - Design cost (huge design teams, why?)[设计]
  - Cost of making first chip after design (mask cost)[流片]
  - Per unit manufacturing cost (wafer cost)[量产]
- Low power/energy[低能耗]
  - Energy in (battery life, cost of electricity)
  - Energy out (cooling and related costs)
  - Cyclic problem, very much a problem today
- **Challenge**: balancing the relative importance of these goals
  - And the balance is constantly changing
  - No goal is absolutely important at expense of all others
  - Our focus: performance, only touch on cost, power, reliability

Manufacturing Process[制造流程]

- Silicon wafers[晶圆] undergo many processing steps so that different parts of the wafer behave as insulators, conductors, and transistors (switches)
- Multiple metal layers on the silicon enable connections between transistors
- The wafer is chopped into many dies[裸晶或裸片] – the size of the die determines yield and cost



晶圆：制造晶圆作为衬底

光刻：把电路版图信息转移到晶圆上

蚀刻：去除多余的空白部分

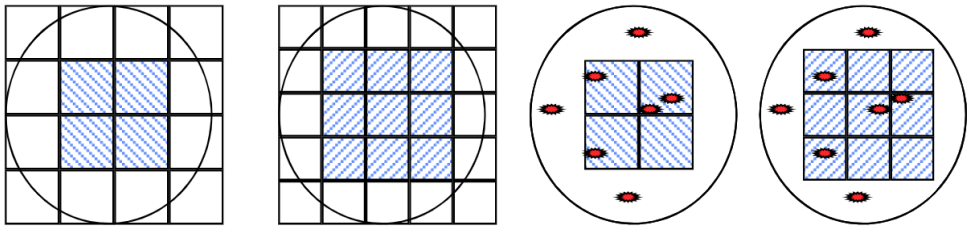
掺杂：离子注入，控制导电性

测试：检查是否符合要求

封装：切割晶圆，获得单个芯片

Integrated Circuits Costs (§1.6)[成本]

- $Dies\ per\ wafer = \frac{\pi * (Wafer\ diameter/2)^2}{Die\ area} - \frac{\pi * Wafer\ diameter}{\sqrt{2 * Die\ area}}$



- $Die\ yield = Wafer\ yield * \frac{1}{(1 + Defects\ per\ unit\ area * Die\ area)^N}$

Where *N*= process-complexity factor=7.5-9.5(28nm,2017)

- $Cost\ of\ die = \frac{Cost\ of\ wafer}{Dies\ per\ wafer * Die\ yield}$
- $Cost\ of\ IC = \frac{Cost\ of\ die + Cost\ of\ testing\ die + Cost\ of\ packaging\ and\ final\ test}{Final\ test\ yeild}$

Integrated Circuits Costs (cont.)

- Real-world examples

							Compound Annual Percentage Change:		
Year Intel 1st Shipped New Product at Tech Node	Tech Node (nm)	Wafer Processing Cost ($ / mm²)	X	Transistor size (mm² / transistor)	=	$ Cost / Transistor	Wafer Processing Cost ($ / mm²)	Transistor size (mm² / transistor)	$ Cost / Transistor
2002	130	1		1		1			
2004	90	1.09		0.62		0.68	5%	-21%	-18%
2006	65	1.24		0.38		0.47	7%	-21%	-16%
2008	45	1.43		0.24		0.34	7%	-21%	-15%
2010	32	1.64		0.15		0.24	7%	-21%	-16%
2012	22	1.93		0.09		0.18	8%	-21%	-14%
2014	14	2.49		0.04		0.11	14%	-31%	-22%

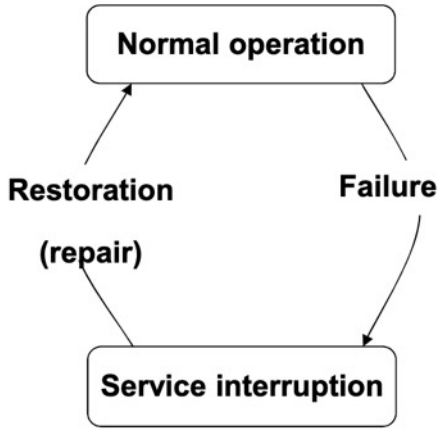
Source: Bill Holt, "Advancing Moore's Law," presentation to Intel Investor Meeting, 2015, Santa Clara, slide 6, graph digitized using WebPlotDigitizer. Year node introduced from ark.intel.com .

Wafer size conversions offset Intel's increased wafer-processing cost

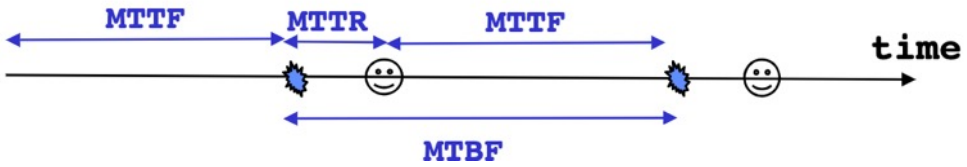
https://www.imf.org/~/media/Files/Conferences/2017-stats-forum/session-6-kenneth-flamm.ashx

Dependability (§1.7)[可靠性]

- Fault vs. error
  - Fault: failure of a component
  - Error: manifestation of a fault
  - Faults may or may not lead to system failure



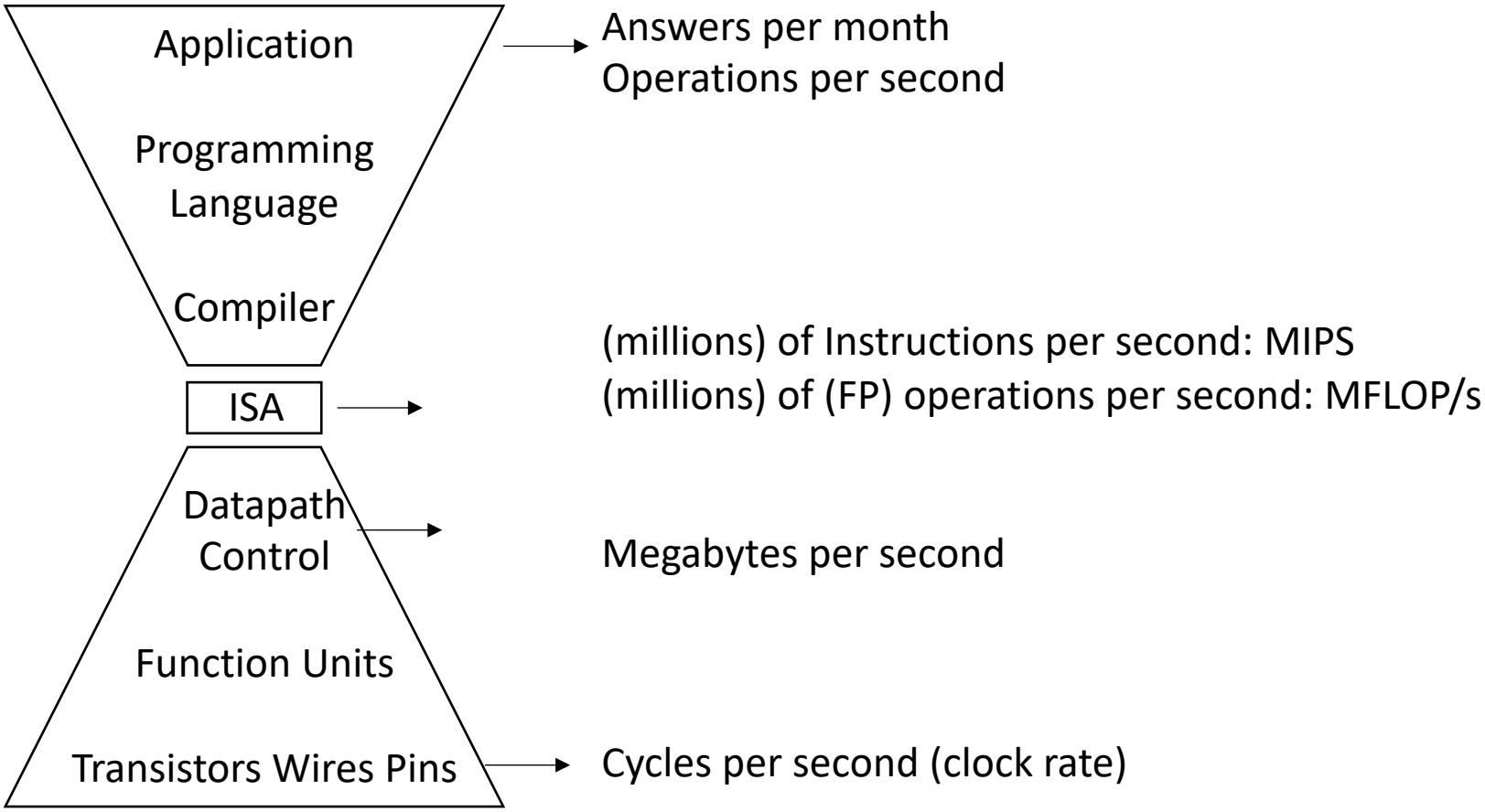
- Metrics
  - Reliability measure: mean time to failure (MTTF)
  - Repair efficiency: mean time to repair (MTTR)
  - Mean time between failures
    - MTBF = MTTF + MTTR
  - Availability = MTTF / MTBF



- Improving availability
  - Increase MTTF: fault avoidance, fault tolerance, fault forecasting
  - Reduce MTTR: improved tools and processes for diagnosis/repair

Performance (§1.8)[性能]

- The performance metric may mean different things



Measuring Performance[评估性能]

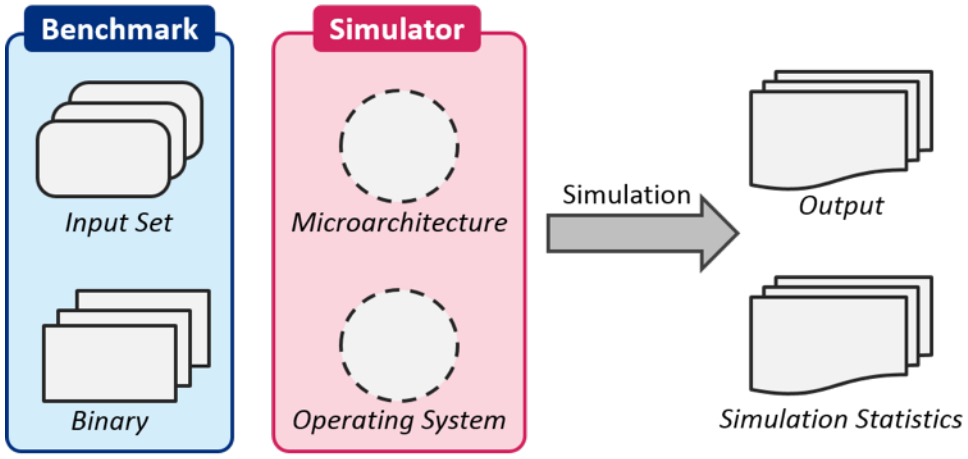
- Time to run the task (latency)
  - Execution time, response time, CPU time, …
- Tasks per day, hour, week, sec, ns, …
  - Throughput, bandwidth
- Performance measurement[测试]
  - Hardware prototypes : cost, delay, area, power estimation
  - Simulation (many levels, ISA, RT, Gate, Circuit, …)
  - Benchmarks (kernels, toy programs, synthetic), Traces, Mixes
  - Analytical modeling and Queuing Theory

Measuring Performance (cont.)

- Direct measurement[直接测量]
  - Can provide the best result: no simplifying assumptions
  - Not flexible (difficult to change parameters)
  - Prone to perturbation (if instrumented)
  - Made much easier these days by using performance counters
- Simulation[模拟]
  - Very flexible
  - Time consuming
  - Difficult to model details and validate
- Analytical modeling[分析模型]
  - Quick insight for overall behaviors
  - Limited applicability
  - Used to confine simulation scope, validate simulations, etc.

Simulator[模拟器]

- What is an architecture (or architectural) simulator?
  - A tool that reproduces the behavior of a computing device
- Why use a simulator?
  - Leverage faster, more flexible software development cycle
  - Permits more design space exploration
  - Facilitates validation before hardware becomes available
  - Possible to increase/improve system instrumentation



Simulation Goals Vary[不同目标]

- Explore the design space quickly and see what you want to
  - potentially implement in a next-generation platform
  - propose as the next big idea to advance the state of the art
  - the goal is mainly to see relative effects of design decisions
- Match the behavior of an existing system so that you can
  - debug and verify it at cycle-level accuracy
  - propose small tweaks to the design that can make a difference in performance or energy
  - the goal is very high accuracy
- Other goals in-between:
  - Refine the explored design space without going into a full detailed, cycle-accurate design
  - Gain confidence in your design decisions made by higher-level design space exploration

Tradeoffs in Simulation[平衡]

- Three metrics to evaluate a simulator
  - Speed, Flexibility, Accuracy
- **Speed**[速度]: How fast the simulator runs (xIPS, xCPS, slowdown)
- **Flexibility**[灵活性]: How quickly one can modify the simulator to evaluate different algorithms and design choices?
- **Accuracy**[准确度]: How accurate the performance (energy) numbers the simulator generates are vs. a real design (Simulation error)
- The relative importance of these metrics varies depending on where you are in the design process (what your goal is)

Tradeoffs in Simulation (cont.)

- Speed & flexibility affect:
  - How quickly you can make design tradeoffs
- Accuracy affects:
  - How good your design tradeoffs may end up being
  - How fast you can build your simulator (simulator design time)
- Flexibility also affects:
  - How much human effort you need to spend modifying the simulator

- You can trade off between the three to achieve design exploration and decision goals

High-level Simulation[高层级模拟]

- Key Idea: Raise the abstraction level of modeling to give up some accuracy to enable speed & flexibility (and quick simulator design)
  - Get first-hand insights
- Advantages
  - Can still make the right tradeoffs, and can do it quickly
  - All you need is modeling the key high-level factors, you can omit corner case conditions
  - All you need is to get the "relative trends" accurately, not exact performance numbers
- Disadvantages
  - Opens up the possibility of potentially wrong decisions
  - How do you ensure you get the "relative trends" accurately?