PERFORMANCE METRICS

Mahdi Nazm Bojnordi

Assistant Professor

School of Computing

University of Utah

Overview

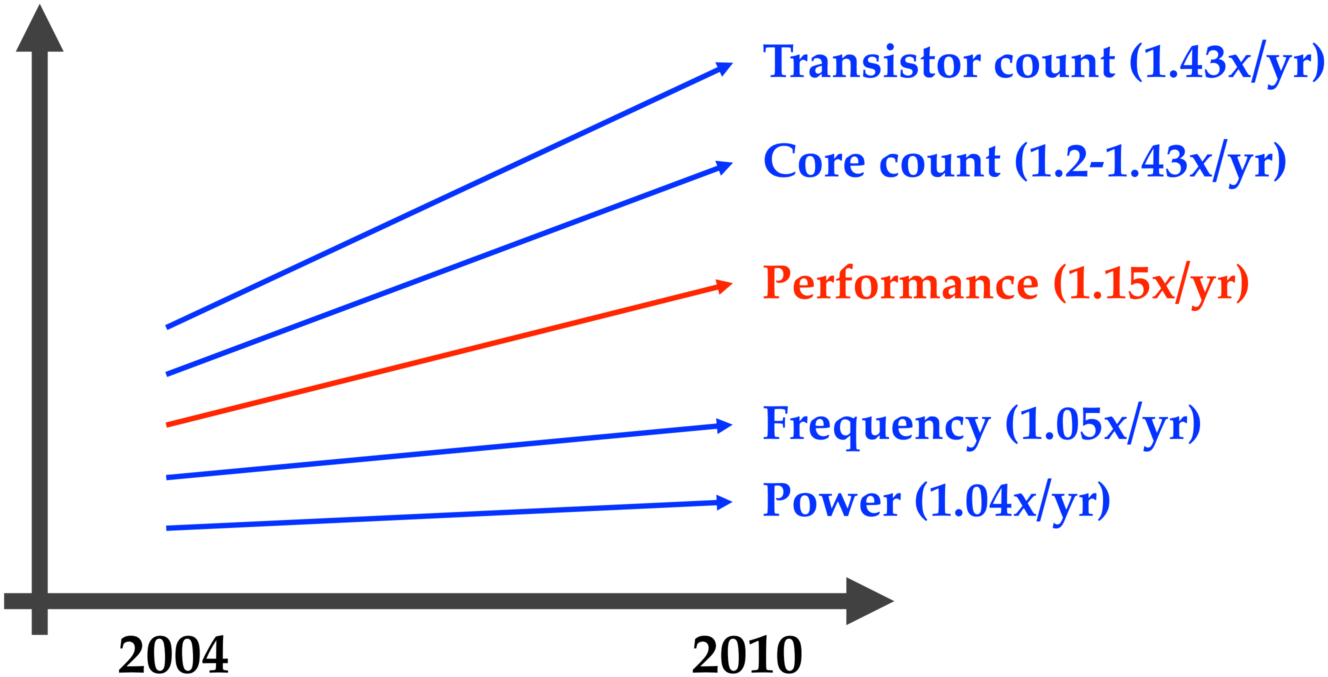
- Announcement
  - Jan. 17th: Homework 1 release (due on Jan. 30th)
- This lecture
  - Technology trends
  - Measuring performance
  - Principles of computer design
  - Power and energy
  - Cost and reliability

Technology Trends (Historical Data)

- IC logic Technology: on-chip transistor count doubles every 18-24 months (Moore's Law)
  - Transistor density increases by 35% per year
  - Die size increases 10-20% per year
- DRAM Technology
  - Chip capacity increases 25-40% per year
- Flash Storage
  - Chip capacity increases 50-60% per year

Technology Trends (Historical Data)

- Recent Microprocessor Trends



Source: Micron University Symposium

Performance Trends

- How to measure performance?
  - Latency or response time: the time between start and completion of an event (e.g., milliseconds for disk access)
  - Bandwidth or throughput: the total amount of work done in a given time (e.g., megabytes per second for disk transfer)
- Which one grows faster?
  - Bandwidth, by at least the square of latency improvement rate.
- Which one is better? latency or throughput?

Measuring Performance

- Which one is better (faster)?

Car	Bus
Delay=10m	Delay=30m
Capacity=4p	Capacity=30p
Throughput=0.4PPM	Throughput=1PPM

It really depends on your needs (goals).

Measuring Performance

- What program to use for measuring performance?
- Benchmarks Suites
  - A set of representative programs that are likely relevant to the user
  - Examples:
    - SPEC CPU 2006: CPU-oriented programs (for desktops)
    - SPECweb: throughput-oriented (for servers)
    - EEMBC: embedded processors/workloads

Summarizing Performance Numbers

- How to capture the behavior of multiple programs with a single number

	Comp-A	Comp-B	Comp-C
Prog-1	10	5	25
Prog-2	5	10	20
Prog-3	25	10	25

- AM: Arithmetic Mean (good for times and latencies)

$$\frac{1}{n}\sum_{i=1}^{n} x_i$$

Summarizing Performance Numbers

- How to capture the behavior of multiple programs with a single number

	Comp-A	Comp-B	Comp-C
Prog-1	1/10	1/5	1/25
Prog-2	1/5	1/10	1/20
Prog-3	1/25	1/10	1/25

- HM: Harmonic Mean (good for rates and throughput)

$$\frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

Summarizing Performance Numbers

- How to capture the behavior of multiple programs with a single number

	Comp-A	Comp-B	Comp-C
Prog-1	10/10	10/5	10/25
Prog-2	5/5	5/10	5/20
Prog-3	25/25	25/10	25/25

- GM: Geometric Mean (good for speedups)

$$\left(\prod_{i=1}^{n} x_i\right)^{1/n}$$

The Processor Performance

- Clock cycle time (CT = 1/clock frequency)
  - Influenced by technology and pipeline
- Cycles per instruction (CPI)
  - Influenced by architecture
  - IPC may be used instead (IPC = 1/CPI)
- Instruction count (IC)
  - Influenced by ISA and compiler
- CPU time = IC x CPI x CT

Example Problem

- Find the average CPI of a load/store machine when running an application that results in the following statistics

Instruction Type	Frequency	Cycles
Load	20%	2
Store	20%	2
Branch	20%	2
ALU	40%	1

$$CPI = 0.2x2 + 0.2x2 + 0.2x2 + 0.4x1 = 1.6$$

Example Problem

- Find the average CPI of a load/store machine when running an application that results in the following statistics

Instruction Type	Frequency	Cycles
Load	20%	2
Store	20%	2
Branch	20%	2
ALU	40%	1

- 50% of the branches can be combined with ALU instructions and executed as Branch-ALU fused in 2 cycles. What is the new average CPI?

Example Problem

- Find the average CPI of a load/store machine when running an application that results in the following statistics

Instruction Type	Frequency	Cycles
Load	22%	2
Store	22%	2
Branch	11%	2
ALU	33%	1
Branch-ALU	12%	2

- 80% of the branches can be combined with ALU instructions and executed as Branch-ALU fused in 2 cycles. What is the new average CPI? CPI = 1.67

The Processor Performance

- Points to note
  - Performance = 1 / execution time
  - AM(IPCs) = 1 / HM(CPIs)
  - GM(IPCs) = 1 / GM(CPIs)

$$\frac{1}{n}\sum_{i=1}^{n} x_i \qquad \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i}} \qquad \left(\prod_{i=1}^{n} x_i\right)^{1/n}$$

Speedup vs. Percentage

- Speedup = old execution time / new execution time
- Improvement = (new performance - old performance)/old performance
- My old and new computers run a particular program in 80 and 60 seconds; compute the followings
  - speedup = 80/60
  - percentage increase in performance = 33%
  - reduction in execution time = 20/80 = 25%

Example Problem

- A new computer has an IPC that is 20% worse than the old one. However, it has a clock speed that is 30% higher than the old one. If running the same binaries on both machines. What speedup is the new computer providing? Speedup = 1/0.96 = 1.04

	OLD	NEW
IPC	1	0.8
Frequency	1	1.3
IC	1	1
CPI	1/1	1/0.8 = 1.25
CT	1/1	1/1.3 ~ 0.77
CPU Time	1	~0.96

Principles of Computer Design

- Designing better computer systems requires better utilization of resources
  - Parallelism
    - Multiple units for executing partial or complete tasks
  - Principle of locality (temporal and spatial)
    - Reuse data and functional units
  - Common Case
    - Use additional resources to improve the common case

Amdahl's Law

- The law of diminishing returns

$$\text{Execution time}_{\text{new}} = \text{Execution time}_{\text{old}} \times \left((1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right)$$

$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

Example Problem

- Our new processor is 10x faster on computation than the original processor. Assuming that the original processor is busy with computation 40% of the time and is waiting for IO 60% of the time, what is the overall speedup?

$f=0.4 \quad s=10$

Speedup = $1 / (0.6 + 0.4/10) = 1/0.64 = 1.5625$

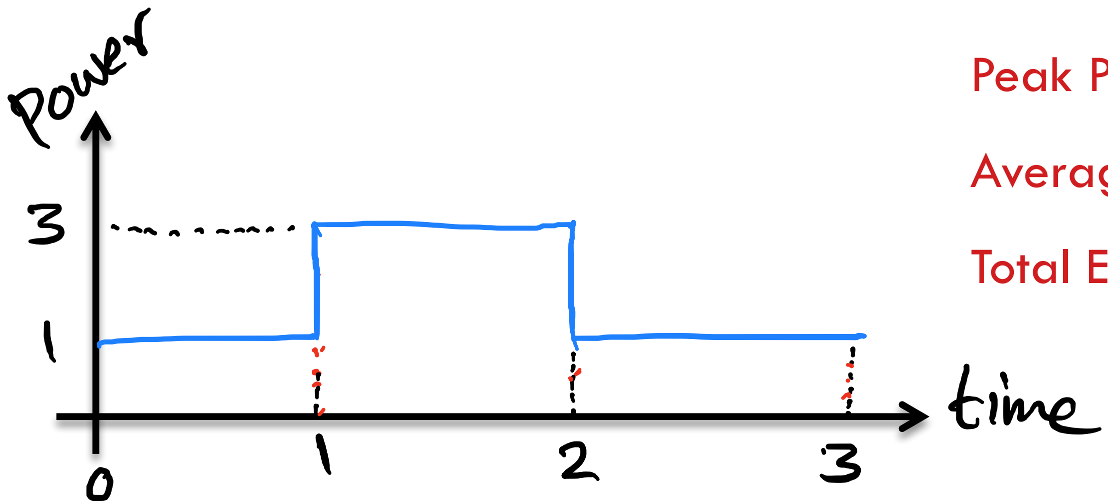
Power and Energy

Power and Energy

- Power = Voltage x Current ($P = VI$)
  - Instantaneous rate of energy transfer (Watt)
- Energy = Power x Time ($E = PT$)
  - The cost of performing a task (Joule)

Power and Energy

- Power = Voltage x Current (P = VI)
  - Instantaneous rate of energy transfer (Watt)
- Energy = Power x Time (E = PT)
  - The cost of performing a task (Joule)



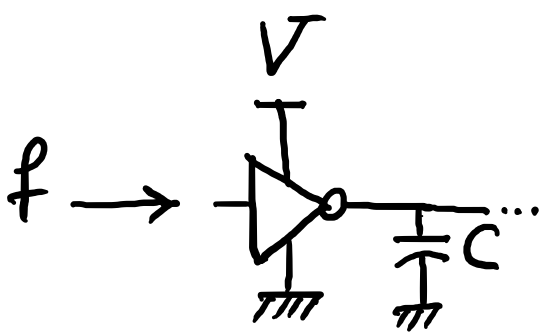
Peak Power = 3W

Average Power = 1.66W

Total Energy = 5J

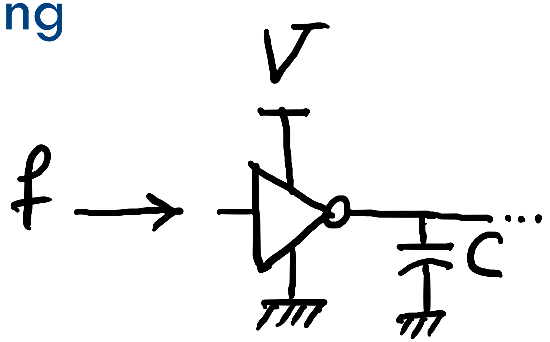
CPU Power and Energy

- All consumed energy is converted to heat
  - CPU power is the rate of heat generation
  - Excessive peak power may result in burning the chip
- Static and dynamic energy components
  - Energy = ($Power_{Static}$ + $Power_{Dynamic}$) x Time
  - $Power_{Static}$ = Voltage x $Current_{Static}$
  - $Power_{Dynamic}$ = Activity x Capacitance x $Voltage^2$ x Frequency



Power Reduction Techniques

- Reducing capacitance (C)
  - Requires changes to physical layout and technology
- Reducing voltage (V)
  - Negative effect on frequency
  - Opportunistically power gating (wakeup time)
  - Dynamic voltage and frequency scaling
- Reducing frequency (f)
  - Negative effect on CPU time
  - Clock gating in unused resources
- Points to note
  - Utilization directly effects dynamic power
  - Lowering power does NOT mean lowering energy



Example Problem

- For a processor running at 100% utilization and consuming 60W, 30% of the power is attributed to leakage. What is the total power dissipation when the processor is running at 50% utilization?

Example Problem

- For a processor running at 100% utilization and consuming 60W, 30% of the power is attributed to leakage. What is the total power dissipation when the processor is running at 50% utilization?
- @100%
  - Power = 18W + 42W = 60W
- @50%
  - Power = 18W + 21W = 39W

Example Problem

- A processor consumes 80W of dynamic power and 20W of static power at 3GHz. It completes a program in 20 seconds. What is the energy consumption if frequency scales down by 20%?

Example Problem

- A processor consumes 80W of dynamic power and 20W of static power at 3GHz. It completes a program in 20 seconds. What is the energy consumption if frequency scales down by 20%?
- @3GHz
  - Energy = (80W + 20W) x 20s = 2000J
- @2.4GHz
  - Energy = (0.8x80W + 20W) x 20/0.8 = 2100J

Example Problem

- A processor consumes 80W of dynamic power and 20W of static power at 3GHz. It completes a program in 20 seconds. What is the energy consumption if frequency scales down by 20%?
- What is the energy consumption if voltage and frequency scale down by 20%?

Example Problem

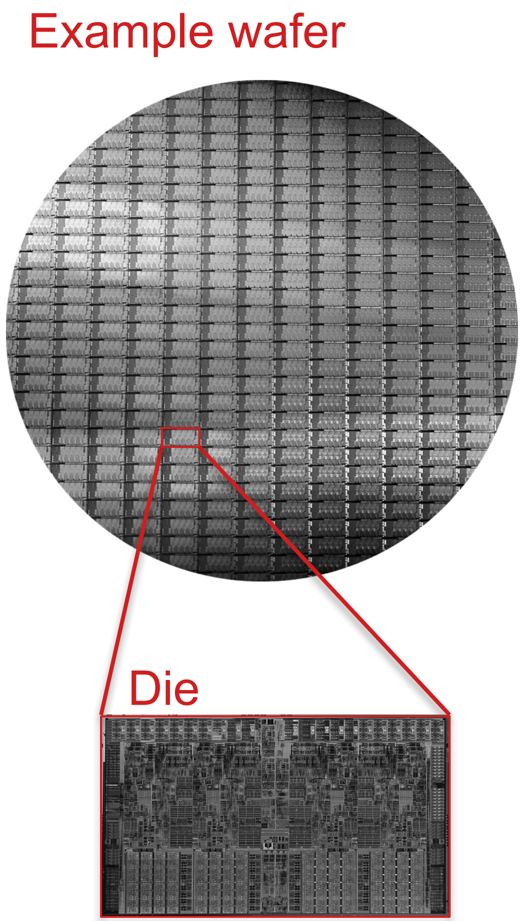
- A processor consumes 80W of dynamic power and 20W of static power at 3GHz. It completes a program in 20 seconds. What is the energy consumption if frequency scales down by 20%?
- What is the energy consumption if voltage and frequency scale down by 20%?

- @ 80%V and 80%f
  - Energy = $(80 \times 0.8^2 \times 0.8 + 20 \times 0.8) \times 20/0.8 = 1424$J

Cost and Reliability

Cost of Integrated Circuit

- Cost of die
  - $$\frac{wafer\ cost}{dies\ per\ wafer \times die\ yield}$$
- Yield of die
  - $$\frac{wafer\ yield}{(1+defect\ per\ unit\ area \times die\ area)^N}$$
- N: process-complexity factor
  - Specified by chip manufacturer



Example Problem

- Defect rate for a $144mm^2$ die is 0.5 per cm^2. Assuming that we use a 40nm technology node (N=11) with 100% wafer yield, find the die yield.

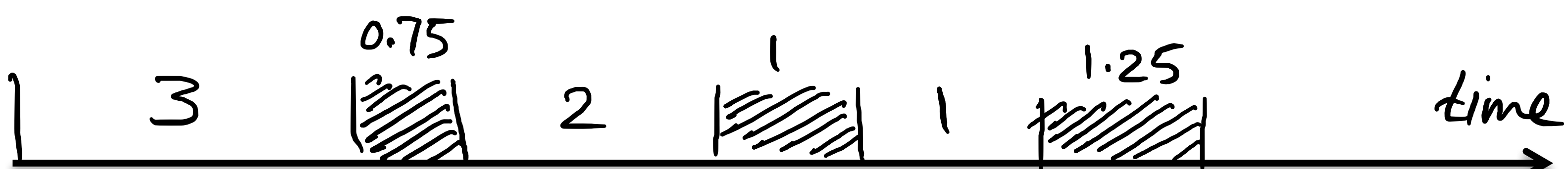
Example Problem

- Defect rate for a 144mm^2 die is 0.5 per cm^2. Assuming that we use a 40nm technology node (N=11) with 100% wafer yield, find the die yield.

- Die yield = $1/(1 + 0.5 \times 1.44)^{11}$

Dependability

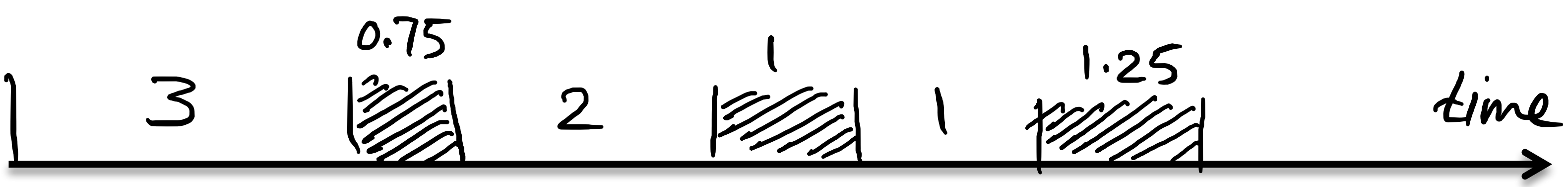
- A measure of system's reliability and availability
- System reliability
  - A measure of continuous service (time-to-failure)
  - Mean Time To Failure (MTTF)
  - Mean Time To Repair (MTTR)



- System availability

Dependability

- A measure of system's reliability and availability
- System reliability
  - A measure of continuous service (time-to-failure)
  - Mean Time To Failure (MTTF) = (3+2+1)/3 = 2
  - Mean Time To Repair (MTTR) = (0.75+1+1.25)/3 = 1



- System availability

$$\frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} = 2/(2+1) = 0.67$$