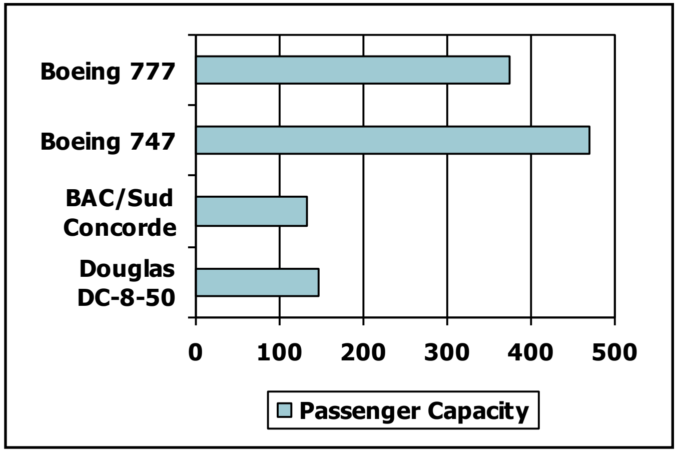
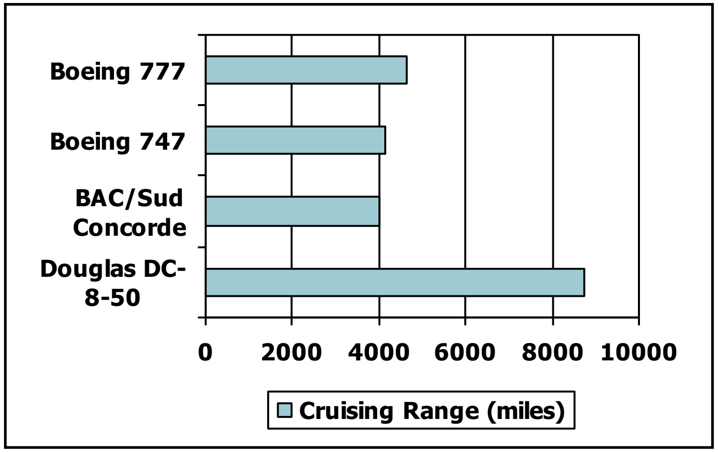
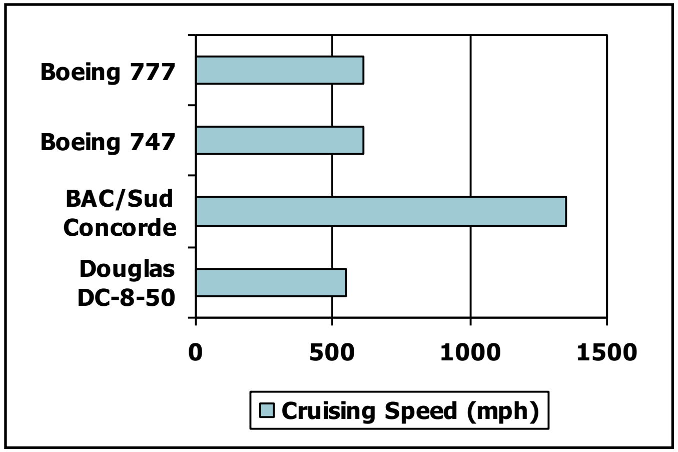
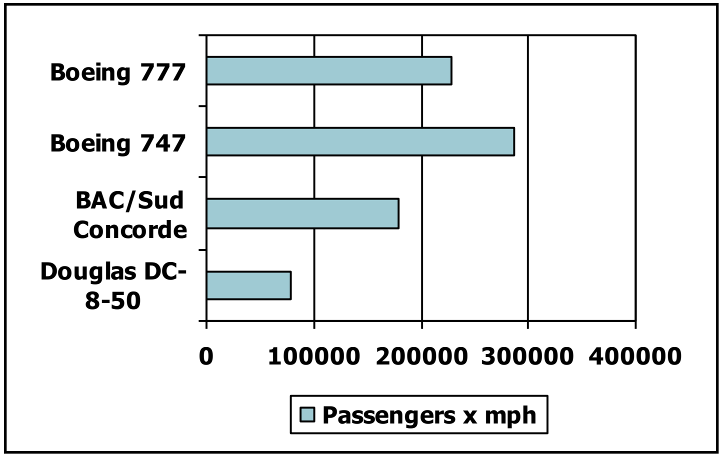
Unit 2 Computer Performance

- Which airplane has the best performance?









Response Time and Throughput

- Response time
  - How long it takes to do a task
- Throughput
  - Total work done per unit time
    - e.g., tasks/transactions/… per hour
- How are response time and throughput affected by
  - Replacing the processor with a faster version?
  - Adding more processors?
- We'll focus on response time for now…

Relative Performance

- Define Performance = 1/Execution Time
- "X is n time faster than Y"

$$\text{Performance}_X / \text{Performance}_Y$$
$$= \text{Execution time}_Y / \text{Execution time}_X = n$$

- Example: time taken to run a program
  - 10s on A, 15s on B
  - Execution Time_B / Execution Time_A = 15s / 10s = 1.5
  - So A is 1.5 times faster than B

Measuring Execution Time

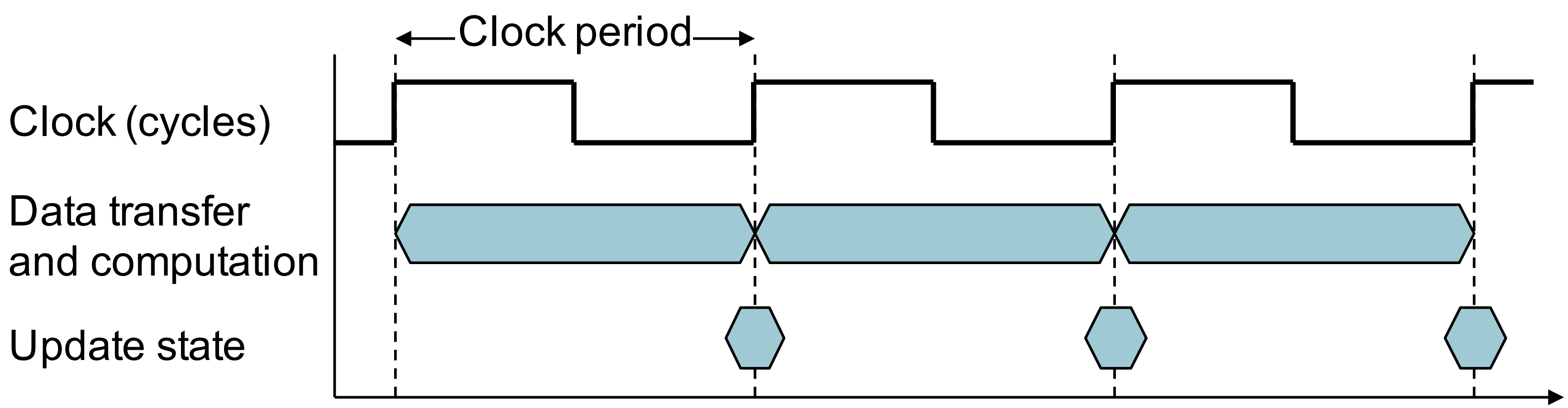
- Elapsed time
  - Total response time, including all aspects
    - Processing, I/O, OS overhead, idle time
  - Determines system performance
- CPU time
  - Time spent processing a given job
    - Discounts I/O time, other jobs' shares

      do not count I/O
  - Comprises user CPU time and system CPU time
  - Different programs are affected differently by CPU and system performance

CPU Clocking

Clock period is the duration of a clock cycle

- Operation of digital hardware governed by a constant-rate clock



- Clock period: duration of a clock cycle
  - e.g., 250ps = 0.25ns = 250×10^{-12}s

$1 ps = 10^{-12} s.$

- Clock frequency (rate): cycles per second
  - e.g., 4.0GHz = 4000MHz = 4.0×10^{9}Hz

$1 Mhz = 10^6 Hz.$
$1 GHz = 10^9 Hz$

CPU Time

$t = \frac{CC}{CR} = CC \times CT$

number of cycles

Period of each cycle $= \frac{1(s)}{\text{Clock rate}}$

$$\text{CPU Time} = \text{CPU Clock Cycles} \times \text{Clock Cycle Time}$$
$$= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}}$$

the frequency of the CPU chip, i.e. clock rates per second, e.g. 4 GHz

- Performance improved by
  - Reducing number of clock cycles → beat faster?
  - Increasing clock rate → better chip?
  - Hardware designer must often trade off clock rate against cycle count

CPU Time Example

$t = \frac{CC}{CR}$

- Computer A: 2GHz clock, 10s CPU time
- Designing Computer B
  - Aim for 6s CPU time
  - Can do faster clock, but causes 1.2 × clock cycles
- How fast must Computer B clock be?

$\Rightarrow CC_A = t_A \times CR_A = 10 \times 2 \times 10^9 = 20 \times 10^9$

$\Rightarrow CC_B = 1.2 \times 20 \times 10^9 = 24 \times 10^9$

$CR_B = \frac{CC_B}{t_B} = \frac{24 \times 10^9}{6} = 4 \times 10^9 = 4\text{GHz}$

$$\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6\text{s}}$$

$$\text{Clock Cycles}_A = \text{CPU Time}_A \times \text{Clock Rate}_A$$

$$= 10\text{s} \times 2\text{GHz} = 20 \times 10^9$$

$$\text{Clock Rate}_B = \frac{1.2 \times 20 \times 10^9}{6\text{s}} = \frac{24 \times 10^9}{6\text{s}} = 4\text{GHz}$$

Instruction Count and CPI

$\boxed{CC = IC \times CPI} \Rightarrow \boxed{t = \frac{CC}{CR.} = \frac{IC \times CPI.}{CR.}}$

$$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$$

$$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$$

$$= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

- Instruction Count for a program
  - Determined by program, ISA and compiler
- Average cycles per instruction
  - Determined by CPU hardware
  - If different instructions have different CPI
    - Average CPI affected by instruction mix

II depends → program, ISA, compiler

Avg CPI depends → CPU hardware, instruction mix.

CPI Example

$$t = \frac{CC}{CR} = \frac{IC \times CPI}{CR} = IC \times CPI \times CT$$

$CC = IC \times CPI$

$CT = \frac{1}{CR}$

$t_A = IC \times 2.0 \times 250 \times 10^{-12}$
$= IC \times 500 \times 10^{-12}$

- Computer A: Cycle Time = 250ps, CPI = 2.0
- Computer B: Cycle Time = 500ps, CPI = 1.2
- Same ISA $\longrightarrow$ same IC
- Which is faster, and by how much?

$t_B = IC \times 1.2 \times 500 \times 10^{-12}$
$= IC \times 600 \times 10^{-12}$

$\Rightarrow t_B > t_A$

$\Rightarrow$ A is faster

$\frac{600}{500} = 1.2$ times

$$\text{CPU Time}_A = \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A$$
$$= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}$$

A is faster…

$$\text{CPU Time}_B = \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B$$
$$= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}$$

$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{I \times 600\text{ps}}{I \times 500\text{ps}} = 1.2$$

…by this much

MORGAN KAUFMANN

CPI in More Detail

- If different instruction classes take different numbers of cycles

$$\text{Clock Cycles} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{Instruction Count}_i)$$

- Weighted average CPI

$$\text{Avg CPI} = \frac{\Sigma\ \text{Clock Cycles}}{\Sigma\ \text{Instruction Count}} = \sum_{i=1}^{n} \left(\text{CPI}_i \times \underbrace{\frac{\text{Instruction Count}_i}{\text{Instruction Count}}}_{\text{Relative frequency}} \right)$$

$CC = IC \times CPI$

$\text{Relative Freq} = \sum_{i=1}^{n} \frac{IC_i}{IC}$

CPI Example

- Alternative compiled code sequences using instructions in classes A, B, C

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2 $\times 1 = 2$	1 $\times 2 = 2$	2 $\times 3 = 6$
IC in sequence 2	4 $\times 1 = 4$	1 $\times 2 = 2$	1 $\times 3 = 3$

$CC_1 = 10$

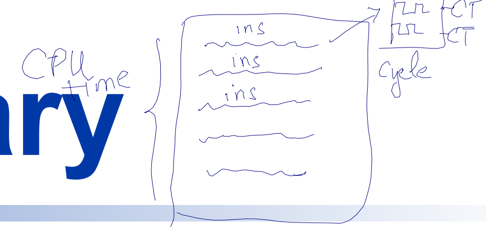
$Avg\,CPI_1 = \frac{CC_1}{IC_1} = \frac{10}{5} = 2$

$CC_2 = 9$

$Avg\,CPI_2 = \frac{CC_2}{IC_2} = \frac{9}{6} = 1.5$

- Sequence 1: IC = 5
  - Clock Cycles
    = 2×1 + 1×2 + 2×3
    = 10
  - Avg. CPI = 10/5 = 2.0
- Sequence 2: IC = 6
  - Clock Cycles
    = 4×1 + 1×2 + 1×3
    = 9
  - Avg. CPI = 9/6 = 1.5

Performance Summary



The BIG Picture

instructions per program

IC

cycles per instruction

CPI

time per cycle (cycle time)

CT

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

- Performance depends on
  - Algorithm: affects IC, possibly CPI
  - Programming language: affects IC, CPI
  - Compiler: affects IC, CPI
  - Instruction set architecture: affects IC, CPI, T_c

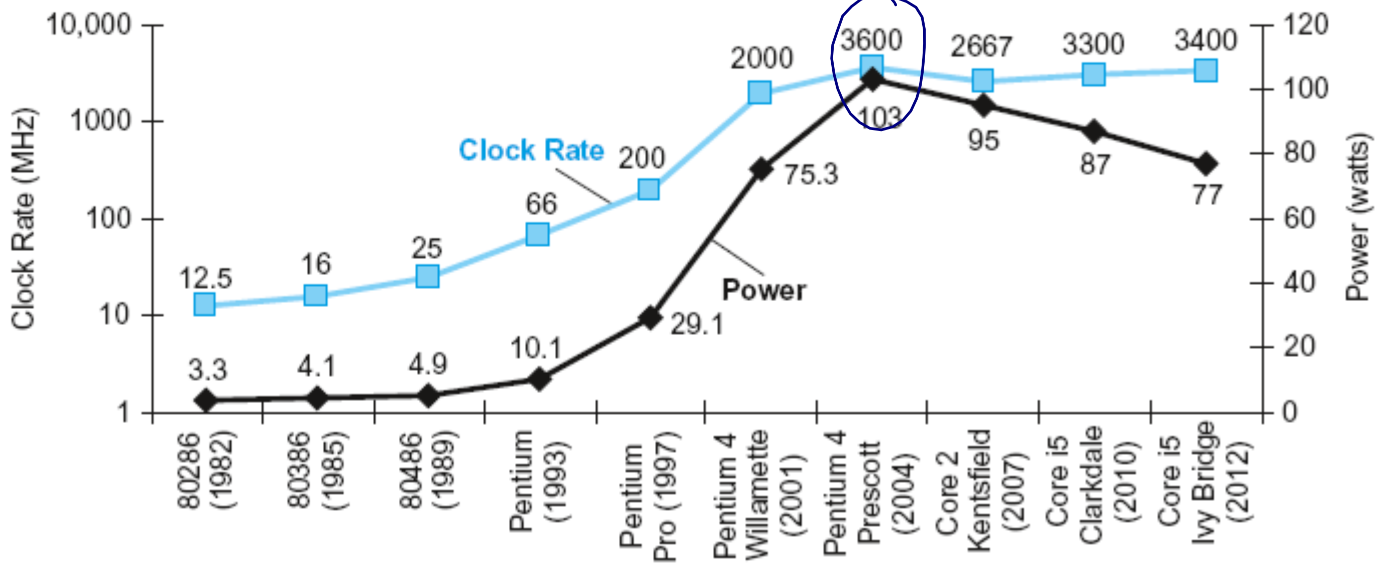
ISA

ISA can affect T_c

Power Trends

Power = Capacity Load × Voltage² × Frequency

Pentium 4: peak in CR & power consumption
2004



- In CMOS IC technology

$$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

×30

5V → 1V

×1000

Reducing Power

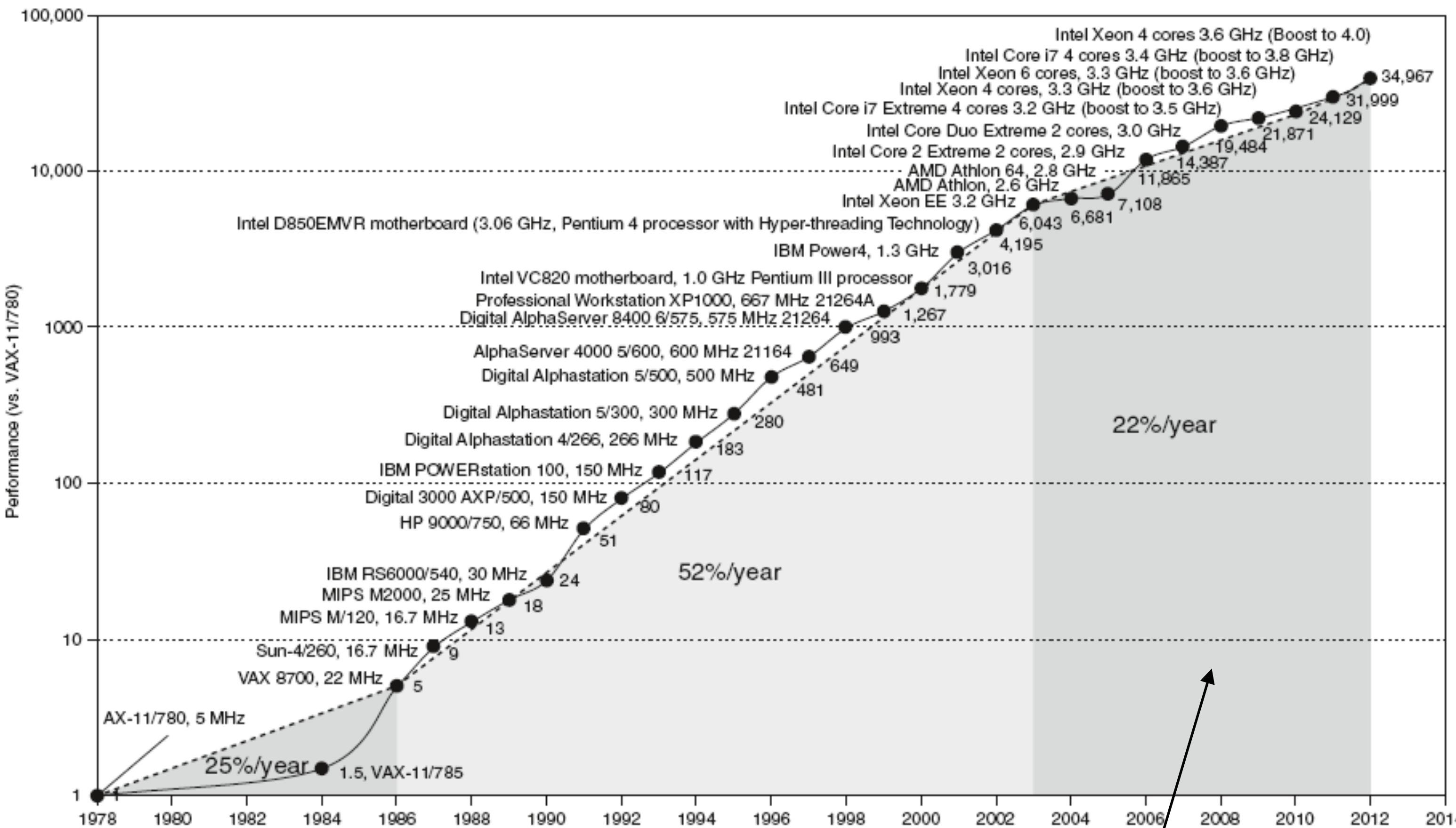
- Suppose a new CPU has
  - 85% of capacitive load of old CPU
  - 15% voltage and 15% frequency reduction

$$\frac{P_{new}}{P_{old}} = \frac{C_{old} \times 0.85 \times (V_{old} \times 0.85)^2 \times F_{old} \times 0.85}{C_{old} \times V_{old}{}^2 \times F_{old}} = 0.85^4 = 0.52$$

less power usage than old computer

- The power wall
  - We can't reduce voltage further
  - We can't remove more heat
- How else can we improve performance?

Uniprocessor Performance



Constrained by power, instruction-level parallelism, memory latency

Multiprocessors

- Multicore microprocessors
  - More than one processor per chip
- Requires explicitly parallel programming
  - Compare with instruction level parallelism
    - Hardware executes multiple instructions at once
    - Hidden from the programmer
  - Hard to do
    - Programming for performance
    - Load balancing
    - Optimizing communication and synchronization

SPEC CPU Benchmark

- Programs used to measure performance
  - Supposedly typical of actual workload
- Standard Performance Evaluation Corp (SPEC)
  - Develops benchmarks for CPU, I/O, Web, …
- SPEC CPU2006
  - Elapsed time to execute a selection of programs
    - Negligible I/O, so focuses on CPU performance
  - Normalize relative to reference machine
  - Summarize as geometric mean of performance ratios
    - CINT2006 (integer) and CFP2006 (floating-point)

$$\sqrt[n]{\prod_{i=1}^{n} \text{Execution time ratio}_i}$$

CINT2006 for Intel Core i7 920

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	–	–	–	–	–	–	25.7

SPEC Power Benchmark

- Power consumption of server at different workload levels
  - Performance: ssj_ops/sec
  - Power: Watts (Joules/sec)

$$\text{Overall ssj\_ops per Watt} = \left(\sum_{i=0}^{10} \text{ssj\_ops}_i \right) \Big/ \left(\sum_{i=0}^{10} \text{power}_i \right)$$

SPECpower_ssj2008 for Xeon X5650

Target Load %	Performance (ssj_ops)	Average Power (Watts)
100%	865,618	258
90%	786,688	242
80%	698,051	224
70%	607,826	204
60%	521,391	185
50%	436,757	170
40%	345,919	157
30%	262,071	146
20%	176,061	135
10%	86,784	121
0%	0	80
Overall Sum	4,787,166	1,922
Σssj_ops/Σpower =		2,490

Pitfall: Amdahl's Law

- Improving an aspect of a computer and expecting a proportional improvement in overall performance

$$T_{\text{improved}} = \frac{T_{\text{affected}}}{\text{improvement factor}} + T_{\text{unaffected}}$$

- Example: multiply accounts for 80s/100s
  - How much improvement in multiply performance to get 5× overall?

$\frac{100}{5}$

$$= 20 = \frac{80}{n} + 20$$

  - Can't be done!
- Corollary: make the common case fast

Fallacy: Low Power at Idle

- Look back at i7 power benchmark
  - At 100% load: 258W
  - At 50% load: 170W (66%)
  - At 10% load: 121W (47%)
- Google data center
  - Mostly operates at 10% – 50% load
  - At 100% load less than 1% of the time
- Consider designing processors to make power proportional to load

Pitfall: MIPS as a Performance Metric

- MIPS: Millions of Instructions Per Second
  - Doesn't account for
    - Differences in ISAs between computers
    - Differences in complexity between instructions

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} = \frac{IC}{t \times 10^6} = \frac{IC}{\frac{IC \times CPI}{CR} \times 10^6} = \frac{CR}{CPI \times 10^6}$$

per second

no of instructions per sec.

= how many clock rate per sec. / how many clock rate per inst. in ave.

$$= \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

- CPI varies between programs on a given CPU

Concluding Remarks

- Cost/performance is improving
  - Due to underlying technology development
- Hierarchical layers of abstraction
  - In both hardware and software
- Instruction set architecture
  - The hardware/software interface
- Execution time: the best performance measure
- Power is a limiting factor
  - Use parallelism to improve performance

Unit 2 Homework

1. How to measure a computer performance ? By CPU time
2. Suppose we have a program runs in 100 seconds on a computer, and the multiplication work takes 40 seconds. If we want the program to run 30% faster, how many seconds do we have to improve the multiplication work ?
3. There are 2 computers. Computer A clock cycle time is 350ps, CPI is 3. Computer B clock cycle time is 400ps, CPI is 2.5. Which computer runs faster on same program ?
4. There are 3 instructions types A B C, each instruction type CPI is A:4, B:2, C:3. If we have two programs, program 1 has 2 of A type instructions, 3 of B type instructions, 4 of C type instructions; program 2 has 3 of A type instructions, 4 of B type instruction, 3 C type instruction. Which program runs faster on the same computer ?
5. Why does the computer microprocessor vendors switch to multi-processors ?
6. What are the software factors which will affect the program performance ?

reach the speed

parallel makes the processing faster

increase throughput

(2)

100 sec → multiplication 40 secs
→ others 60 secs.

↓ 70%

$70\text{ sec} = \frac{40}{n} + 60.$

$\Rightarrow 70 = \frac{40 + 60n}{n}.$

$70n = 40 + 60n$

$10n = 40$

$n = 4$

⇒ multiplication must improve 4 times so the overall program can run 30% faster

(Q3)

$CT_A = 350 \times 10^{12}, CPI_A = 3$

$CT_B = 400 \times 10^{12}, CPI_B = 2.5$

$\frac{T_A}{T_B} = \frac{IC \times CPI_A \times CT_A}{IC \times CPI_B \times CT_B}$

$= \frac{350 \times 3}{400 \times 2.5}$

$= \frac{1050}{1000} = 1.05$

Machine A is 5% faster than machine B

(Q4)

	A	B	C
CPI	4	2	3
Prg 1	2	3	4
Prg 2	3	4	3

$t_1 = (4 \times 2 + 2 \times 3 + 3 \times 4) \times CT$

$= (8 + 6 + 12) \times CT$

$= 26\,CT$

$t_2 = (4 \times 3 + 2 \times 4 + 3 \times 3) \times CT$

$= (12 + 8 + 9) \times CT$

$= 29\,CT$

$t_2 > t_1$, therefore program 1 runs faster than program 2.