Internet Hoax
- This photo was entered in a photo doctoring contest (before Adobe Photoshop was created)
- The “computer” is actually part of a submarine control console

Ken Olsen’s Famous Remark
- Ken Olsen, President of Digital Equipment Corp, 1977: “there is no reason for any individual to have a computer in his home.”
- Remark was taken out of context - he was referring to the computer of science fiction: “…the fully computerized home that automatically turned lights on and off and that prepared meals and controlled daily diets …”

The Turing Bombe

ENIAC - background
- Electronic Numerical Integrator And Computer
- Eckert and Mauchly, University of Pennsylvania
- Designed to calculate trajectory tables for weapons
- Started 1943
- Finished 1946
  - Too late for war effort
- Used until 1955
ENIAC - details
- Decimal (not binary)
- 20 accumulators of 10 digits
  - Each accumulator had 100 vacuum tubes
  - One decimal digit was represented by a ring of 10 tubes; only one “on” at a time
- Programmed manually by switches
- 18,000 vacuum tubes
- 30 tons
- 15,000 square feet
- 140 kW power consumption
- 5,000 additions per second

Programming in the ‘40s

von Neumann/Turing
- Stored Program concept
- Features
  - Data and instructions (programs) are stored in a single read/write memory
  - Memory contents are addressable by location, regardless of the content itself
  - Sequential execution
  - ALU operating on binary data
  - Control unit interpreting instructions from memory and executing
  - Input and output equipment operated by control unit
- Princeton Institute for Advanced Studies (IAS)
- Completed 1952

Paraphrased from von Neumann’s paper:
- 1. Since the device is primarily a computer it will have to perform the elementary operations of arithmetic most frequently. Therefore, it should contain specialized organs for just these operations, i.e., addition, subtraction, multiplication, and division.
- 2. The logical control of the device (i.e., proper sequencing of its operations) can best be carried out by a central control organ.
- 3. A device which is to carry out long and complicated sequences of operation must have a considerable memory capacity.

Von Neumann (cont’d)
- 4. The device must have organs to transfer information from the outside recording medium of the device into the central arithmetic part and central control part, and the memory. These organs form its input.
- 5. The device must have organs to transfer information from the central arithmetic part and central control part, and the memory into the outside recording medium. These organs form its output.
**IAS Machine**

- This architecture came to be known as the “von Neumann” architecture and has been the basis for virtually every machine designed since then.

**IAS - details**

- 1000 x 40 bit words
  - Binary number
  - 2 x 20 bit instructions
- Set of registers (storage in CPU)
  - Memory Buffer Register
  - Memory Address Register
  - Instruction Register
  - Instruction Buffer Register
  - Program Counter
  - Accumulator
  - Multiplier Quotient

**IAS Instruction Set**

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>Op code (2 bit)</th>
<th>Instruction description</th>
<th>Function(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Transfer</td>
<td>00, 01</td>
<td>MOV, ADD, SUB, MUL</td>
<td>Data Transfer</td>
</tr>
<tr>
<td>Arithmetic</td>
<td>10</td>
<td>ADD, SUB, MUL</td>
<td>Operations</td>
</tr>
<tr>
<td>Shift</td>
<td>11</td>
<td>SHL, SHR, ROR</td>
<td>Shift</td>
</tr>
</tbody>
</table>

**IAS Operation**

- [Diagram of IAS Operation]

**Structure of von Neumann (IAS) machine**

- Central Processing Unit (CPU)
- Main Memory (M)
- I/O Equipment (I/O)
- Arithmetic-Logic Unit (CA)
- Program Control Unit (CC)

- [Diagram of von Neumann Structure]
**Commercial Computers**
- 1947 - Eckert-Mauchly Computer Corp
- UNIVAC I (Universal Automatic Computer)
  - Commissioned by US Bureau of Census 1950
  - First successful commercial computer
- Became part of Sperry Rand Corporation
- Late 1950s - UNIVAC II
  - Faster and with more memory than UNIVAC I
  - Backwards compatible
- Developed 1100 series for scientific computation
- Scientific/text processing split

**IBM**
- The major manufacturer of punched-card processing equipment
- 1953 - the 701
  - IBM’s first stored program computer
  - Scientific calculations
- 1955 - the 702
  - Business applications
- Lead to 700/7000 series
- Established IBM dominance

**Transistors: 2nd generation**
- Replaced vacuum tubes
- Smaller, cheaper, less heat dissipation
- Solid State device, made from silicon (sand)
- Invented 1947 at Bell Labs by William Shockley et al.
- Used in computers in the late 1950’s
- Defined the “second generation” of computers

**Transistor Based Computers**
- Second generation machines
- NCR & RCA produced small transistor machines
- IBM 7000
- DEC - 1957
  - Produced PDP-1
  - First in series of PDP machines that marked a radical shift in philosophy of computer design: focus on interaction with user rather than efficient use of computer cycles

**Hacker culture**
- “Hacker” used to mean someone who loved to play with computers
- DEC PDP computers led to the development of hacker cultures at MIT, BBN and elsewhere
- Developed music programs and the first computer game Spacewar!
- The hacker culture lives on in the open source movement

**The IBM 7094**
- Last of the 7000 series
- Illustrates evolution of a computer family
An IBM 7094 Configuration

Integrated Circuits; 3rd Generation
• Transistors are “discrete components”
• Manufactured in own container; soldered to circuit board
• Early 2nd generation: 10,000 discrete transistors
• Later computers several hundred thousand
• 1958: Integrated circuits (ICs)
  — Many transistors on piece of silicon
  — Defines 3rd generation of computers
• Two most important 3rd generation computers were the IBM System 360 and the DEC PDP-8

How many transistors on one chip?
• Intel: see http://www.intel.com/pressroom/kits/quickreffam.htm
• 8086: 29,000
• 80386: 275,000
• Original Pentium: 3.1 million
• Pentium IV: up to 169 million
• Itanium: up to 1.72 billion

Computer Components
• Only two basic components are needed to create a computer: gates and memory cells
• Each component is composed of multiple transistors

Four basic functions
• Data Storage: memory cells
• Data Processing: gates
• Data Movement: connections (paths) between gates and memory cells
• Control: control signals activate gates and memory and determine whether memory access is read or write
• Gates, memory cells and interconnections can be etched into a single piece of silicon

Chip manufacturing
**Moore’s Law**

- Gordon Moore - co-founder of Intel
- Predicted that the number of transistors on a chip will double every year
- Since 1970’s development has slowed a little
  - Number of transistors doubles every 18 months
- Cost of a chip has remained almost unchanged
- Higher packing density means shorter electrical paths, giving higher performance
- Smaller size gives increased flexibility
- Reduced power and cooling requirements
- Fewer interconnections increases reliability

**Generations of Computers**

- Vacuum tube - 1946-1957
- Transistor - 1958-1964
- Small scale integration - 1965 on
  - Up to 100 devices on a chip
- Medium scale integration - to 1971
  - 100-3,000 devices on a chip
- Large scale integration - 1971-1977
  - 3,000 - 100,000 devices on a chip
- Very large scale integration - 1978 - 1991
  - 100,000 - 100,000,000 devices on a chip
- Ultra large scale integration - 1991 -
  - Over 100,000,000 devices on a chip

**Growth in CPU Transistor Count**

**The System 360 Family**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Model 30</th>
<th>Model 40</th>
<th>Model 50</th>
<th>Model 60</th>
<th>Model 70</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum memory size (bytes)</td>
<td>64K</td>
<td>256K</td>
<td>512K</td>
<td>2M</td>
<td>4M</td>
</tr>
<tr>
<td>Data rate from memory (MB/sec)</td>
<td>0.5</td>
<td>0.8</td>
<td>2.0</td>
<td>4.0</td>
<td>8.0</td>
</tr>
<tr>
<td>Processor cycle time (ns)</td>
<td>1.0</td>
<td>0.625</td>
<td>0.625</td>
<td>0.625</td>
<td>0.2</td>
</tr>
<tr>
<td>Relative speed</td>
<td>1</td>
<td>3.5</td>
<td>10</td>
<td>21</td>
<td>50</td>
</tr>
<tr>
<td>Maximum number of I/O channels</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Maximum data rate on one channel (KB/sec)</td>
<td>250</td>
<td>400</td>
<td>1000</td>
<td>1250</td>
<td>1250</td>
</tr>
</tbody>
</table>

**IBM 360 series**

- Introduced in 1964
- Replaced (but not compatible with) 7000 series
- First planned “family” of computers
  - Similar or identical instruction sets
  - Similar or identical O/S
  - Increasing speed
  - Increasing number of I/O ports (i.e. more terminals)
  - Increased memory size
  - Increased cost of higher family members

**System 360**

- Differences were:
  - Complexity of ALU circuitry allowing parallel execution
  - Increased bus width allowing faster data transfer
- The 360 family gave IBM a huge market share
- Architecture still used for IBM mainframes
DEC PDP-8

- 1964
- First minicomputer (named after the miniskirt!)
- Did not need air conditioned room
- Small enough to sit on a lab bench
- $16,000
  - vs. $100k+ for IBM 360
- Embedded applications & OEM
- Introduced the BUS structure (vs. the central-switched IBM architecture)

Beyond 3rd Generation

- Classification by generation became less meaningful after the early 1970’s due to rapid pace of technological development
- Two main developments are still being worked out:
  - Semiconductor memory
  - Microprocessors

Magnetic Core Memory

- Each bit stored in a small iron ring (about 1/16" diameter)
- Mostly assembled by hand
- Reads are destructive
- See [http://www.columbia.edu/acis/history/core.html](http://www.columbia.edu/acis/history/core.html)

Semiconductor Memory

- First developed in 1970 by Fairchild
- Size of a single core
  - i.e. 1 bit of magnetic core storage
- Held 256 bits
- Non-destructive read
- Much faster than core
- 1974: became cheaper than core memory
- Capacity approximately doubles each year
- 2GB chips now available

Intel Microprocessors

- 1971 - 4004
  - First microprocessor
  - All CPU components on a single chip
  - 4 bit
- Followed in 1972 by 8008
  - 8 bit
  - Both designed for specific applications
- 1974 - 8080
  - Intel’s first general purpose microprocessor
16 and 32 bit processors

- 16-bit processors were introduced in the late 1970’s
- First 32-processors were introduced in 1981 (Bell Labs and Hewlett-Packard)
- Intel’s 32-bit processor did not arrive until 1985 - and we’re still using that architecture

Intel x86 Processors 1971 - 1989

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>8008</td>
<td>8088</td>
<td>80286</td>
<td>80386</td>
<td>80486</td>
</tr>
<tr>
<td>Cach</td>
<td>1 KB</td>
<td>16 KB</td>
<td>16 KB</td>
<td>32 KB</td>
<td>32 KB</td>
</tr>
<tr>
<td>Speed</td>
<td>MHz</td>
<td>MHz</td>
<td>MHz</td>
<td>MHz</td>
<td>MHz</td>
</tr>
</tbody>
</table>

Intel x86 Processors 1990 - 2002

<table>
<thead>
<tr>
<th>Model</th>
<th>1990</th>
<th>1995</th>
<th>2002</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>P6</td>
<td>Pentium 4</td>
<td>Core 2</td>
</tr>
<tr>
<td>Cach</td>
<td>64 KB</td>
<td>1 MB</td>
<td>1 MB</td>
</tr>
<tr>
<td>Speed</td>
<td>GHz</td>
<td>GHz</td>
<td>GHz</td>
</tr>
</tbody>
</table>

Processing Power

- Today’s $400 personal computer has more computing power than a 1990 mainframe
- Much of this is gobbled up by bloated operating systems (e.g., Vista)
- But many PC applications are only possible with great processing capability:
  - Image processing
  - Video Editing
  - Speech recognition
  - Simulation modeling
  - ...

Speeding it up

- Pipelining
- On board cache
- On board L1 & L2 cache
- Branch prediction
- Data flow analysis
- Speculative execution

Pipelining

- Like an assembly line
- Instruction execution is divided into stages
- When one stage has completed, that circuitry is free to operate on the "next" instruction
**Cache Memory**
- Very fast memory stores copies of data in slow main memory
- Can be onboard (same chip as processor) or in separate chips

**Branch Prediction**
- Processor looks ahead in instruction stream and attempts to predict which branch will be taken by conditional code
- Execute predicted branch and hope it’s right
- Wrong guess leads to pipeline stall

**Data Flow Analysis**
- Processor examines code to determine which instructions are dependent on results of instructions in pipeline
- Reorders instructions in code so that independent instructions can be executed in parallel pipelines

**Speculative Execution**
- With branch prediction and data flow analysis, processor can execute some instructions before they appear in the execution stream
- Results stored in temporary locations

**Performance Balance**
- Processor speed has increased
- Memory capacity has increased
- BUT: Memory speed lags behind processor speed

**Logic and Memory Performance Gap**
**Solutions**
- Increase number of bits retrieved at one time
  - Make DRAM “wider” rather than “deeper”
- Change DRAM interface
  - Cache
- Reduce frequency of memory access
  - More complex cache and cache on chip
- Increase interconnection bandwidth
  - High speed buses
  - Hierarchy of buses

**I/O Devices**
- Peripherals with intensive I/O demands
- Large data throughput demands
- Processors can handle this
- But there are problems moving data between processor and peripheral
- Solutions:
  - Caching
  - Buffering
  - Higher-speed interconnection buses
  - More elaborate bus structures
  - Multiple-processor configurations

**Typical I/O Device Data Rates**

**Key is Balance**
- Processor components
- Main memory
- I/O devices
- Interconnection structures

**Improvements in Chip Organization and Architecture**
- Increase hardware speed of processor
  - Fundamentally due to shrinking logic gate size
    - More gates, packed more tightly, increasing clock rate
    - Propagation time for signals reduced
- Increase size and speed of caches
  - Dedicating part of processor chip
    - Cache access times drop significantly
- Change processor organization and architecture
  - Increase effective speed of execution
  - Parallelism

**Problems with Clock Speed and Logic Density**
- Power
  - Power density increases with density of logic and clock speed
  - Dissipating heat
- RC delay
  - Speed at which electrons flow limited by resistance and capacitance of metal wires connecting them
  - Delay increases as RC product increases
  - Wire interconnects thinner, increasing resistance
  - Wires closer together, increasing capacitance
- Memory latency
  - Memory speeds lag processor speeds
- Solution:
  - More emphasis on organizational and architectural approaches
Intel Microprocessor Performance

**Increased Cache Capacity**
- Typically two or three levels of cache between processor and main memory
- Chip density increased
  - More cache memory on chip
  - Faster cache access
- Pentium chip devoted about 10% of chip area to cache
- Pentium 4 devotes about 50%

**More Complex Execution Logic**
- Enable parallel execution of instructions
- Pipeline works like assembly line
  - Different stages of execution of different instructions at same time along pipeline
- Superscalar processing allows multiple pipelines within single processor
  - Instructions that do not depend on one another can be executed in parallel

**Diminishing Returns**
- Internal organization of processors complex
  - Can get a great deal of parallelism
  - Further significant increases likely to be relatively modest
- Benefits from cache are reaching limit
- Increasing clock rate runs into power dissipation problem
  - Some fundamental physical limits are being reached

**New Approach - Multiple Cores**
- Multiple processors on single chip
  - Large shared cache
- Within a processor, increase in performance proportional to square root of increase in complexity
- If software can use multiple processors, doubling number of processors almost doubles performance
- So, use two simpler processors on the chip rather than one more complex processor
- With two processors, larger caches are justified
  - Power consumption of memory logic less than processing logic
- Example: IBM POWER4
  - Two cores based on PowerPC

**POWER4 Chip Organization**
Pentium Evolution (1)

- 8080
  - First general purpose microprocessor
  - 8 bit data bus
  - Used in first personal computer - Altair
- 8086
  - Much more powerful
  - 16 bit processor and bus
  - Instruction cache, prefetch few instructions
  - 8086 (8 bit external bus) used in first IBM PC
- 80286
  - 16 Mbyte memory addressable
  - Up from 1Mb
- 80386
  - 32 bit
  - Support for multitasking

Pentium Evolution (2)

- 80486
  - Sophisticated powerful cache and instruction pipelining
  - Built in floating point co-processor
- Pentium
  - Superscalar
  - Multiple instructions executed in parallel
- Pentium Pro
  - Increased superscalar organization
  - Aggressive register renaming
  - Branch prediction
  - Data flow analysis
  - Speculative execution

Pentium Evolution (3)

- Pentium II
  - MMX technology
  - Graphics, video & audio processing
- Pentium III
  - Additional floating point instructions for 3D graphics
- Pentium 4
  - Used Arabic rather than Roman numerals
  - Further floating point and multimedia enhancements
- Itanium
  - 64 bit
  - See chapter 15
- Itanium 2
  - Hardware enhancements to increase speed
- See Intel web pages for detailed information on processors

Embedded Systems

- Refers to the use of electronics and software within a product or device, designed to perform a specific function
- Examples include automotive systems (ignition, engine control, braking); consumer electronics and household appliances; industrial control; medical devices; office automation
- There are far more embedded systems in existence than general purpose computers

Embedded Systems

<table>
<thead>
<tr>
<th>Market</th>
<th>Embedded Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automotive</td>
<td>Ignition systems</td>
</tr>
<tr>
<td></td>
<td>Engine control</td>
</tr>
<tr>
<td></td>
<td>Brakes systems</td>
</tr>
<tr>
<td>Consumer electronics</td>
<td>Digital and analog electronics</td>
</tr>
<tr>
<td></td>
<td>Audio and video systems</td>
</tr>
<tr>
<td></td>
<td>Transportation systems</td>
</tr>
<tr>
<td>Industrial control</td>
<td>Automotive and control systems</td>
</tr>
<tr>
<td>Medical</td>
<td>Diagnostics</td>
</tr>
<tr>
<td></td>
<td>Monitors</td>
</tr>
<tr>
<td></td>
<td>Product design</td>
</tr>
<tr>
<td></td>
<td>Quality control</td>
</tr>
<tr>
<td>Office automation</td>
<td>Robots</td>
</tr>
<tr>
<td></td>
<td>Office automation</td>
</tr>
</tbody>
</table>

Constraints on Embedded Systems

- Requirements and constraints vary
  - Small systems may have radically different cost constraints (pennies) from large systems (millions of dollars)
  - Stringent to non-existent quality requirements (safety, reliability, legislated requirements)
  - Short (single use) to long (decades) lifetimes
  - Resistance to environmental conditions such as moisture, pressure, temperature, humidity, radiation
**Constraints on Embedded Systems**

- Different application characteristics depending on static or dynamic loads, sensor speed, response time requirements, computation versus I/O intensive tasks
- Different models of computation ranging from discrete asynchronous event systems to systems with continuous time dynamics

**Environmental Coupling**

- Embedded systems are typically tightly coupled with their environment
- Most systems are subject to real-time response restraints generated by interaction with the environment
  - speed of motion, measurement precision, reactive latency dictate software requirements

**Possible Embedded System Organization**

**ARM Microprocessors**

- ARM is a family of RISC microprocessors and microcontrollers designed by ARM Inc
  - the company designs architectures and licenses them to manufacturers
- ARM processors are widely used in PDAs, games, phones and consumer electronics, including iPod and iPhone

**ARM Evolution**

- Early 1980s: Acorn Inc contracted with BBC to develop a new microcomputer architecture
- Acorn RISC Machine (ARM1) released 1985
  - VLSI Technology was the silicon partner
  - High performance with low power consumption
  - ARM2 added multiply and swap instructions
  - ARM3 added a small cache
- Acorn, VLSI and Apple started ARM Ltd: Advanced RISC Machine: ARM6

**ARM Evolution**

<table>
<thead>
<tr>
<th>Family</th>
<th>32-bit RISC</th>
<th>Cache</th>
<th>Typical MIPS @ MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM1</td>
<td>32</td>
<td>None</td>
<td>7 MIPS @ 12 MHz</td>
</tr>
<tr>
<td>ARM2</td>
<td>32</td>
<td>4 KB</td>
<td>7 MIPS @ 12 MHz</td>
</tr>
<tr>
<td>ARM3</td>
<td>32</td>
<td>4 KB</td>
<td>28 MIPS @ 33 MHz</td>
</tr>
<tr>
<td>ARM6</td>
<td>32</td>
<td>4 KB</td>
<td>28 MIPS @ 33 MHz</td>
</tr>
<tr>
<td>ARM7</td>
<td>32</td>
<td>8 KB</td>
<td>60 MIPS @ 60 MHz</td>
</tr>
<tr>
<td>ARM8</td>
<td>32</td>
<td>8 KB</td>
<td>84 MIPS @ 72 MHz</td>
</tr>
<tr>
<td>ARM9</td>
<td>32</td>
<td>16 KB/16 KB</td>
<td>300 MIPS @ 300 MHz</td>
</tr>
<tr>
<td>ARM9E</td>
<td>Enhanced DSP</td>
<td>16 KB/16 KB</td>
<td>220 MIPS @ 200 MHz</td>
</tr>
<tr>
<td>ARM10E</td>
<td>32</td>
<td>32 KB/32 KB</td>
<td>320 MIPS @ 320 MHz</td>
</tr>
<tr>
<td>ARM11</td>
<td>32</td>
<td>32 KB/32 KB</td>
<td>Variable</td>
</tr>
<tr>
<td>Cortex</td>
<td>32</td>
<td>13 KB</td>
<td>2000 MIPS @ 1 GHz</td>
</tr>
</tbody>
</table>

DSP = digital signal processor
**Processor Assessment**

- What criteria do we use to assess processors?
  - Cost
  - Size
  - Reliability
  - Security
  - Backwards compatibility
  - Power consumption
  - Performance

**Performance Assessment**

- Performance is not easy compare; it varies with
  - Processor Speed
  - Instruction Set
  - Application
  - Choice of programming language
  - Quality of compiler
  - Skill of the programmer

**Processor Speed**

- To measure processor speed, we can consider
  - Clock speed
  - Millions of instructions per second (MIPS)
  - Millions of floating point instructions per second (MFLOPS)

**Clock speed**

- The system clock provides a series of 1/0 pulses (square wave) that drives circuitry in the processor
- Changes in state of circuitry are synchronized with the signal changes (1 to 0 or 0 to 1)
- A quartz crystal is used to generate a sine wave
- Analog to Digital (A/D) converter turns this into a square wave
- One pulse is a clock cycle or clock tick
  - A 1GHz clock produces one billion ticks per second
  - But a machine with a 2GHz is not necessarily twice as fast

**Instruction Execution**

- Machine instructions are operations such as load from memory, add, store to memory
- Instruction execution is a set of discrete steps, such as fetch, decode, fetch operands, execute, write operands
  - Some instructions may take only a few clock cycles while other require dozens
  - When pipelining is used several instructions may be executed in parallel
- So we cannot simply compare clock speeds
Instruction Execution Rate

- If all instructions required the same number of clock ticks then the average execution rate would be a constant dependent on clock speed
- In addition to variation in instruction clocks, we also have to take into account the time needed to read and write memory
  - This can be significantly slower than the system clock

Performance Factors and System Attributes

Where $I_c =$ Instruction Count

$p = $ processor cycles needed to decode / execute

$m = $ number of memory references needed

$k = $ ratio between processor cycle and memory cycle

$\tau = $ 1 / processor cycle (clock time)

Total time $T = I_c * [p + (m * k)] * \tau$

Instructions per Second

- Commonly measured in millions
  
  $\text{MIPS} = \frac{I_c}{(T * 10^6)}$

- Example: program executes 2 million instructions on 400MHz processor. Program has four major types of instructions

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>CPI</th>
<th>Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic and logic</td>
<td>1</td>
<td>60%</td>
</tr>
<tr>
<td>Load/Store with cache hit</td>
<td>2</td>
<td>18%</td>
</tr>
<tr>
<td>Branch</td>
<td>4</td>
<td>12%</td>
</tr>
<tr>
<td>Load/Store with cache miss</td>
<td>8</td>
<td>10%</td>
</tr>
</tbody>
</table>

Avg Cycles/Instruction = $0.6 + 0.36 + 0.48 + .8 = 2.24$

MIPS = $(400 * 10^6) / (2.24 * 10^6) = 178$

MFLOPS

- For heavy-duty “number crunching” applications the number of floating point operations per second is of more interest than MIPS

  $\text{MFLOPS} = \text{Millions of Floating Point Operations} / \text{Second}$

  Important for scientific applications and games

Instruction Set Differences

- MIPS and MFLOPS have limited utility as performance assessments
- Consider $A = B + C$ in a high-level language (HLL)
- Might be translated as one instruction (CISC)
  
  ```
  add mmx(B), mmx(C), mmx(A)
  ```

  Or three (Intel)

  ```
  mov eax, B
  add eax, C
  mov A, eax
  ```

  Or four (typical RISC)

  ```
  load R1, B
  load R2, C
  add R3, R2, R1
  store A, R3
  ```

Benchmarks

- A program designed to test timing for a particular type of application
  - Written in a high level language for portability
  - Representative of a particular programming style (number crunching, business database, systems, ...)
  - Designed for easily measured timing

- Standard Performance Evaluation Corp (SPEC) has produced a widely recognized set of benchmarks
  - Measurements across a variety of benchmarks are averaged using a geometric mean

- See [http://www.spec.org/benchmarks.html](http://www.spec.org/benchmarks.html)
A note on measures of central tendencies

- The conventional arithmetic average is only one of many measures of central tendencies. Three classical or Pythagorean means are (for n observations):
  - Arithmetic mean (sum of the observations divided by n)
  - Geometric mean (n<sup>th</sup> root of the product of the observations)
  - Harmonic mean (n divided by the sum of the reciprocal of the observations)
- It is always true that the arithmetic mean > geometric mean > harmonic mean
- Reasons for selection of a particular measure are beyond the scope of this course.

Amdahl’s Law

- Proposed by Gene Amdahl in 1967 to predict the maximum speedup of a system using multiple processors compared to a single processor
- For any give program some fraction f involves code that is amenable to parallel execution while the remainder of the code (1-f) can only be executed in a serial fashion
- In its simplest form Amdahl’s law for N processors can be expressed as
  Speedup = 1 / [(1-f) + f / N]

Amdahl’s Law

- This leads the rather pessimistic chart below for 1,024 processors
- The very steep curve near the Y axis suggests that very few programs will achieve a speedup proportional to the number of processors

Gustafson’s Law

- John Gustafson argued that in practice the problem size is not independent of the number of processors and that massively parallel systems can achieve near optimal speedup with bounded problem size
- Note that both arguments can be applied to improvements other than parallelism; e.g., computation of floating point arithmetic in hardware rather than software