Computer Organization and Architecture

Chapter 2
Computer Evolution and Performance

First Generation

- ENIAC: Electronic Numerical Integrator And Computer
- Decimal (not binary)
- 20 accumulators of 10 digits
- Programmed manually by switches
- 18,000 vacuum tubes
- 30 tons
- 15,000 square feet

Transistor Based Computers

- Transistors replaced vacuum tubes
- Smaller
- Cheaper
- Less heat dissipation
- Solid State device
- Made from Silicon (Sand)
- Invented 1947 at Bell Labs
- William Shockley et al.

Microelectronics Based Computers

- Literally - “small electronics”
- A computer is made up of gates, memory cells and interconnections
- These can be manufactured on a semiconductor
- e.g. silicon wafer

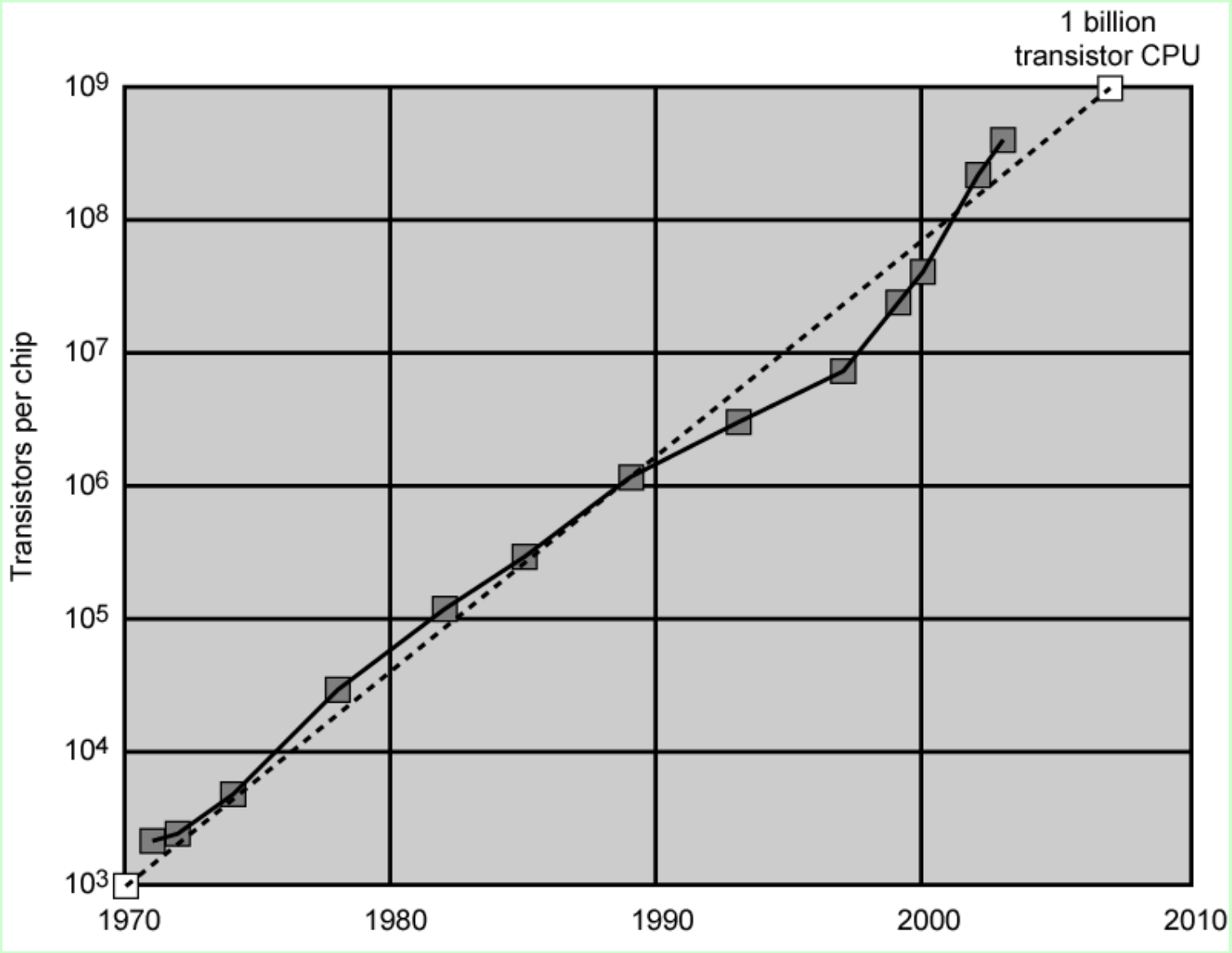
Generations of Computer

- 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

Moore's Law

- Increased density of components on chip
- Gordon Moore – co-founder of Intel
- 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

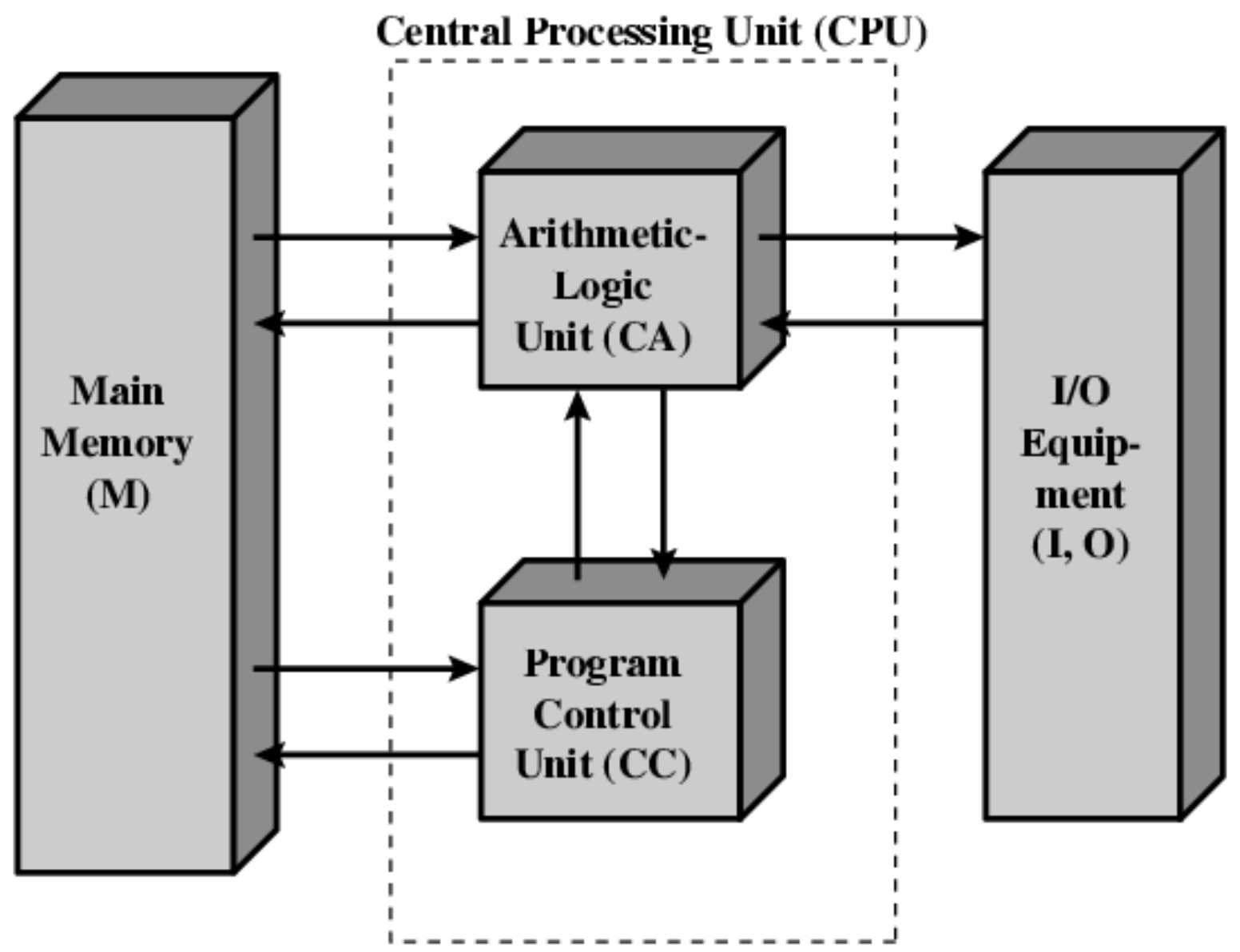
Growth in CPU Transistor Count



Von Neumann

- Stored Program concept
- Main memory storing programs and data
- 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

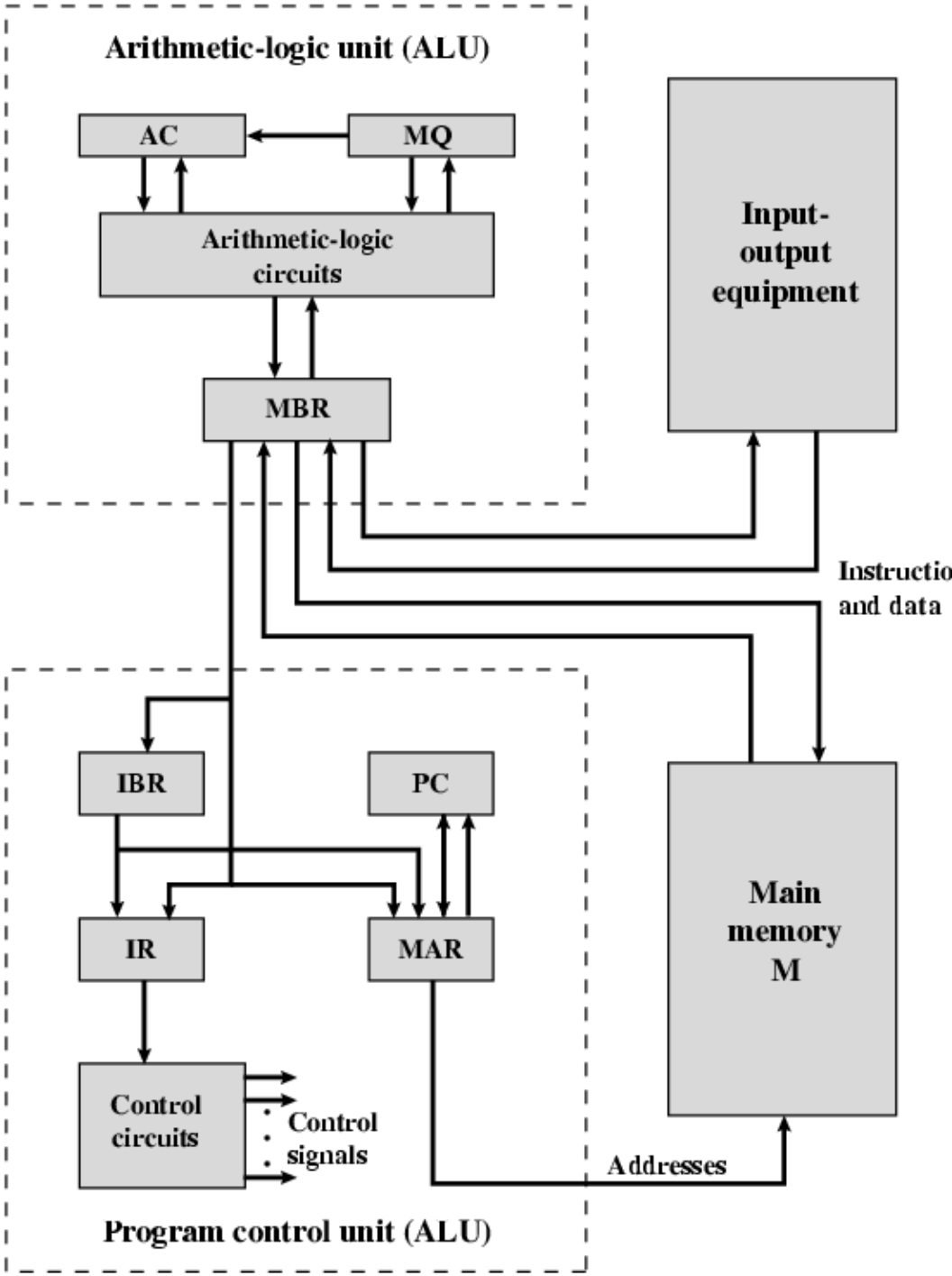
Structure of von Neumann machine



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

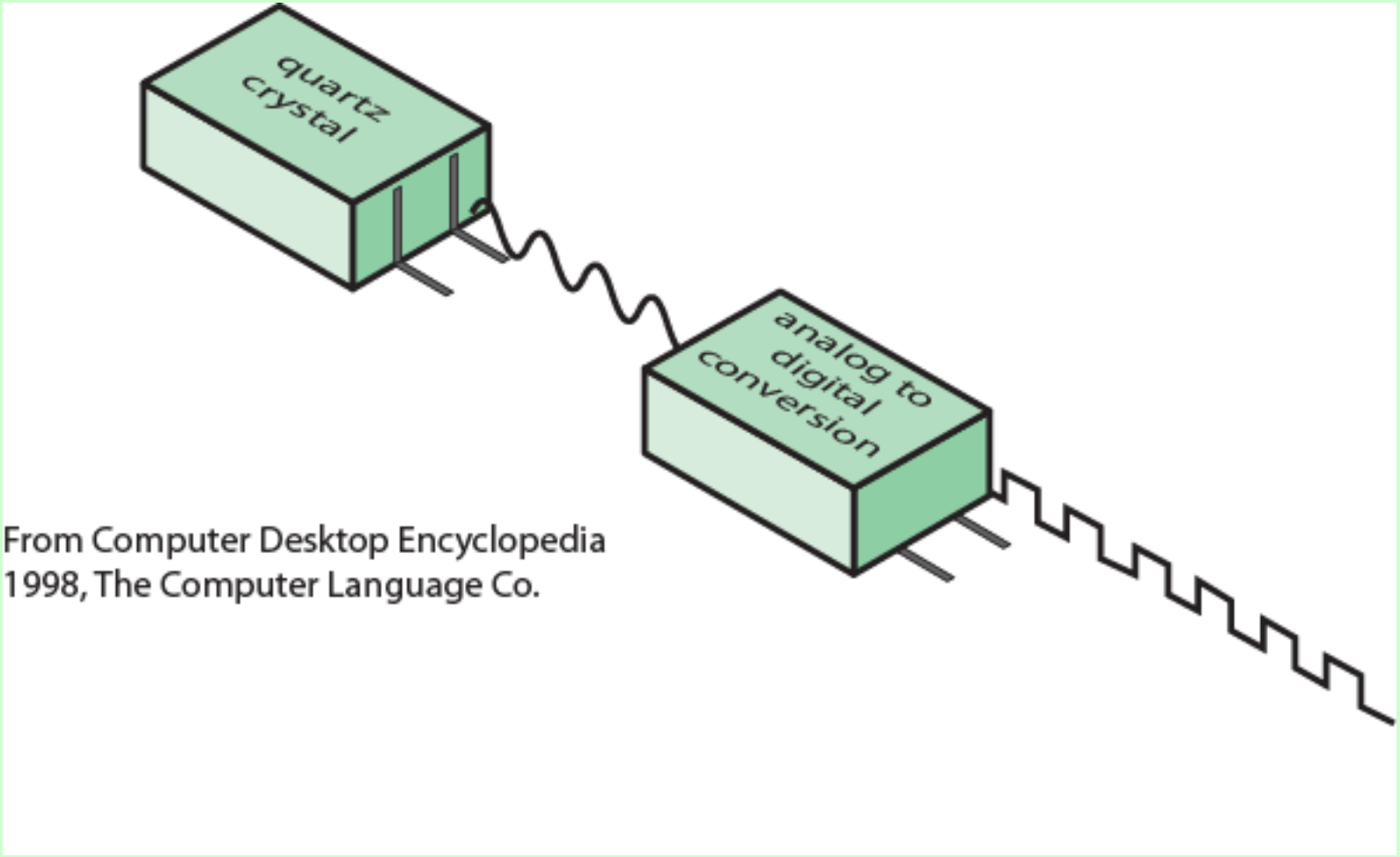
Structure of IAS – detail



Performance Assessment
Clock Speed

- Key parameters
  - Performance, cost, size, security, reliability, power consumption
- System clock speed
  - In Hz or multiples of
  - Clock rate, clock cycle, clock tick, cycle time
- Signals in CPU take time to settle down to 1 or 0
- Signals may change at different speeds
- Operations need to be synchronised
- Instruction execution in discrete steps
  - Fetch, decode, load and store, arithmetic or logical
  - Usually require multiple clock cycles per instruction
- Pipelining gives simultaneous execution of instructions
- So, clock speed is not the whole story

System Clock



From Computer Desktop Encyclopedia
1998, The Computer Language Co.

Instruction Execution Rate

- Millions of instructions per second (MIPS)
- Millions of floating point instructions per second (MFLOPS)
- Heavily dependent on instruction set, compiler design, processor implementation, cache & memory hierarchy

Benchmarks

- Programs designed to test performance
- Written in high level language
  - Portable
- Represents style of task
  - Systems, numerical, commercial
- Easily measured
- Widely distributed
- E.g. System Performance Evaluation Corporation (SPEC)
  - CPU2006 for computation bound
    - 17 floating point programs in C, C++, Fortran
    - 12 integer programs in C, C++
    - 3 million lines of code
  - Speed and rate metrics
    - Single task and throughput

SPEC Speed Metric

- Single task
- Base runtime defined for each benchmark using reference machine
- Results are reported as ratio of reference time to system run time
  - $Tref_i$ execution time for benchmark i on reference machine
  - $Tsut_i$ execution time of benchmark i on test system

$$r_i = \frac{Tref_i}{Tsut_i}$$

- Overall performance calculated by averaging ratios for all 12 integer benchmarks
  - Use geometric mean
    - Appropriate for normalized numbers such as ratios

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

SPEC Rate Metric

- Measures throughput or rate of a machine carrying out a number of tasks
- Multiple copies of benchmarks run simultaneously
  - Typically, same as number of processors
- Ratio is calculated as follows:
  - $Tref_i$ reference execution time for benchmark i
  - N number of copies run simultaneously
  - Tsuti elapsed time from start of execution of program on all N processors until completion of all copies of program
  - Again, a geometric mean is calculated

$$r_i = \frac{N \times Tref_i}{Tsut_i}$$

Amdahl's Law

- Gene Amdahl [AMDA67]
- Potential speed up of program using multiple processors
- Concluded that:
  - Code needs to be parallelizable
  - Speed up is bound, giving diminishing returns for more processors
- Task dependent
  - Servers gain by maintaining multiple connections on multiple processors
  - Databases can be split into parallel tasks

Amdahl's Law Formula

- For program running on single processor
  - Fraction *f* of code infinitely parallelizable with no scheduling overhead
  - Fraction (1-*f*) of code inherently serial
  - T is total execution time for program on single processor
  - N is number of processors that fully exploit parralle portions of code

$$Speedup = \frac{\text{time to execute program on a single processor}}{\text{time to execute program on } N \text{ parallel processors}} = \frac{T(1-f)+Tf}{T(1-f)+\frac{Tf}{N}} = \frac{1}{(1-f)+\frac{f}{N}}$$

- Conclusions
  - *f* small, parallel processors has little effect
  - *N* ->∞, speedup bound by 1/(1 – *f*)
    - Diminishing returns for using more processors

Internet Resources

- http://www.intel.com/
  - —Search for the Intel Museum
- http://www.ibm.com
- http://www.dec.com
- Charles Babbage Institute
- PowerPC
- Intel Developer Home

References

- AMDA67 Amdahl, G. “Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capability”, *Proceedings of the AFIPS Conference, 1967.*