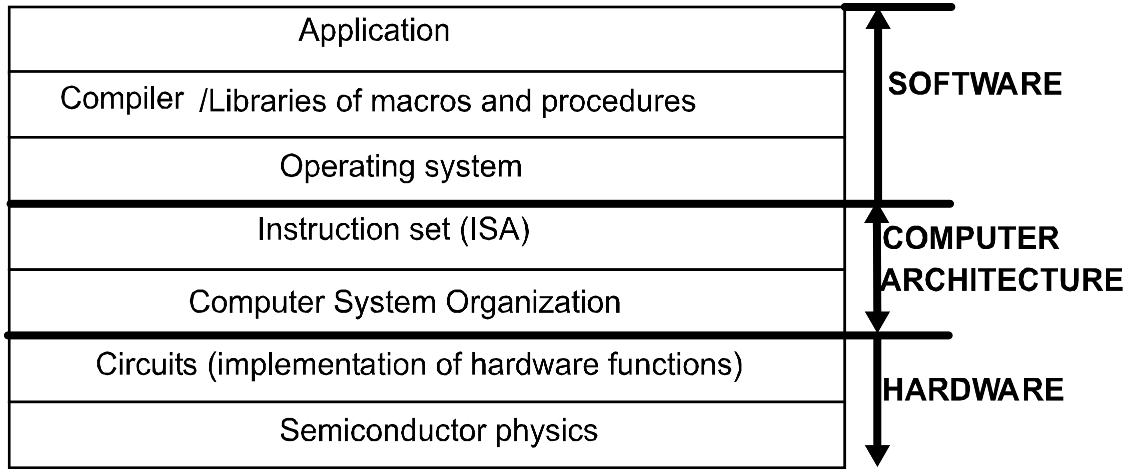
CHAPTER 1

INTRODUCTION

- COMPUTER ARCHITECTURE: DEFINITION
- SYSTEM COMPONENTS
- TECHNOLOGICAL FACTORS AND TRENDS
- PERFORMANCE METRICS AND EVALUATION
- QUANTITATIVE PRINCIPLES OF COMPUTER DESIGN

WHAT IS COMPUTER ARCHITECTURE?

- OLD DEFINITION: INSTRUCTION SET ARCHITECTURE (ISA)
- TODAY'S DEFINITION IS MUCH BROADER: HARDWARE ORGANIZATION OF COMPUTERS (HOW TO BUILD COMPUTER)--INCLUDES ISA
- LAYERED VIEW OF COMPUTER SYSTEMS



Application

Compiler /Libraries of macros and procedures

Operating system

Instruction set (ISA)

Computer System Organization

Circuits (implementation of hardware functions)

Semiconductor physics

SOFTWARE

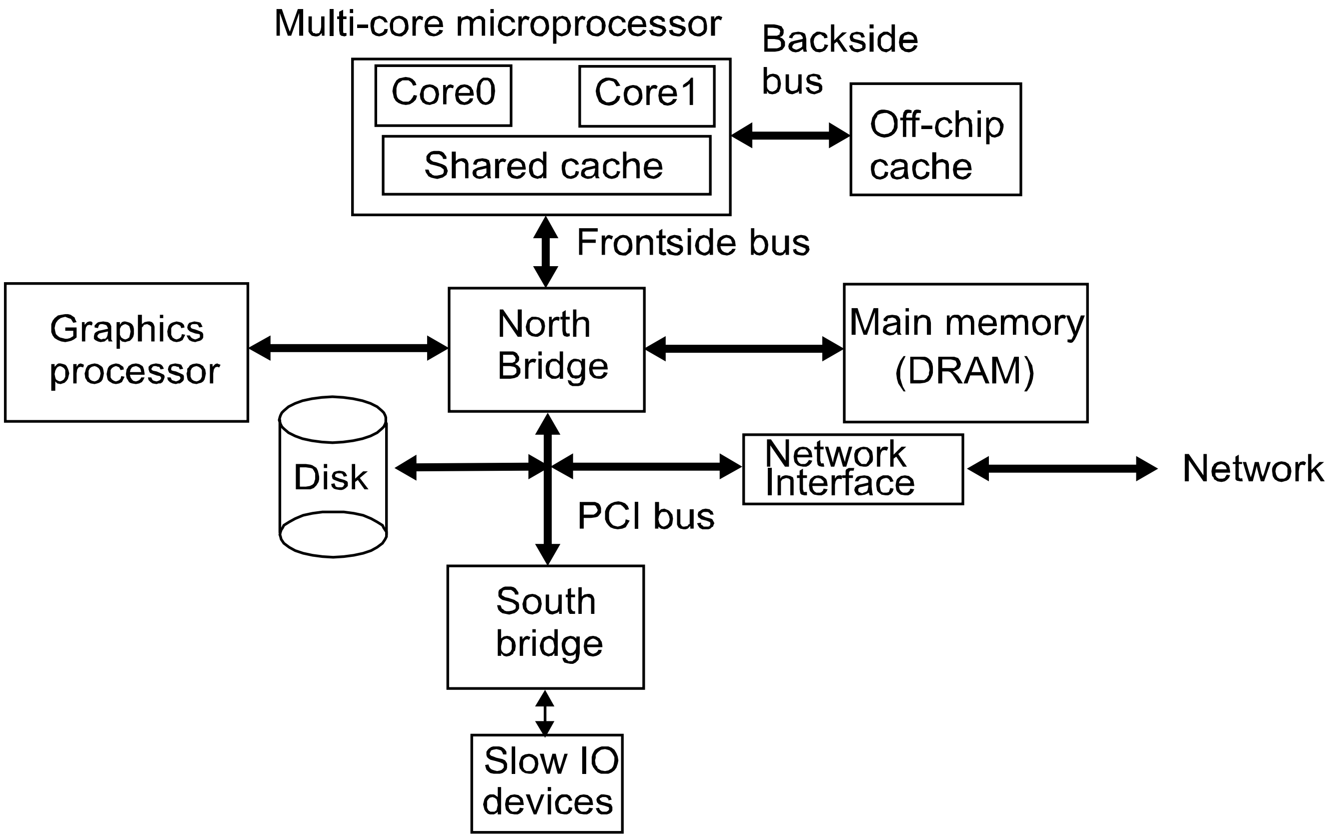
COMPUTER ARCHITECTURE

HARDWARE

- ROLE OF THE COMPUTER ARCHITECT:
  - TO MAKE DESIGN TRADE-OFFS ACROSS THE HW/SW INTERFACE TO MEET FUNCTIONAL, PERFORMANCE AND COST REQUIREMENTS

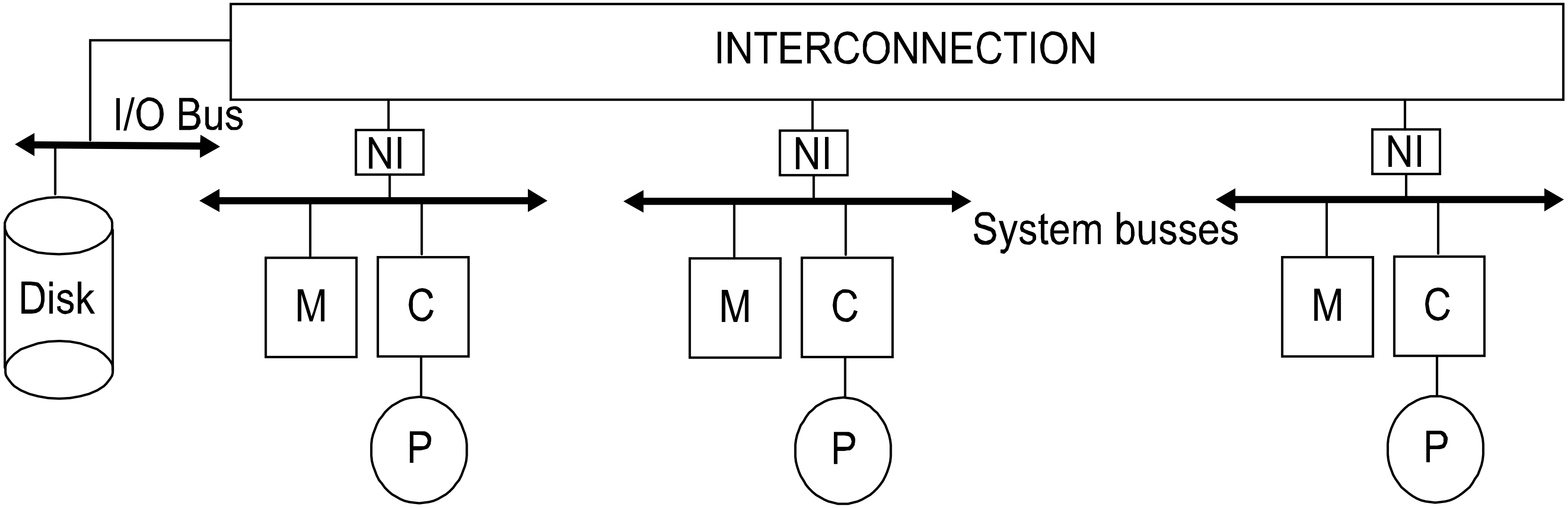
COMPUTER ORGANIZATION

- **MODERN PC ARCHITECTURE**



COMPUTER ORGANIZATION

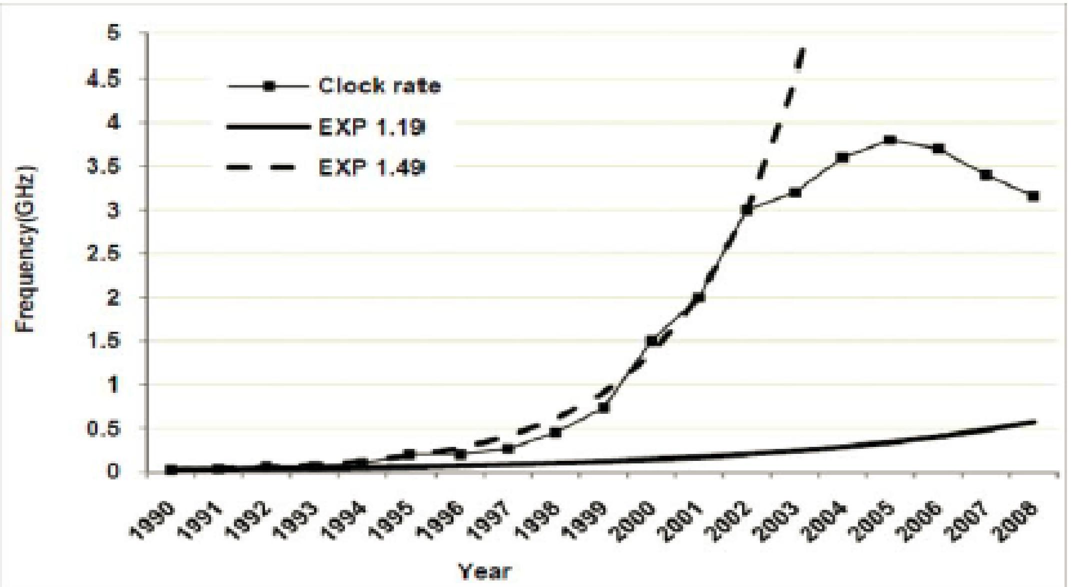
- GENERIC HIGH-END PARALLEL SYSTEM:



- MAIN COMPONENTS: PROCESSOR, MEMORY SYSTEMS, I/O AND NETWORKS,

PROCESSOR ARCHITECTURE

- HISTORICALLY THE CLOCK RATES OF MICROPROCESSORS HAVE INCREASED EXPONENTIALLY
  - Highest clock rate of Intel processors in every year from 1990 to 2008



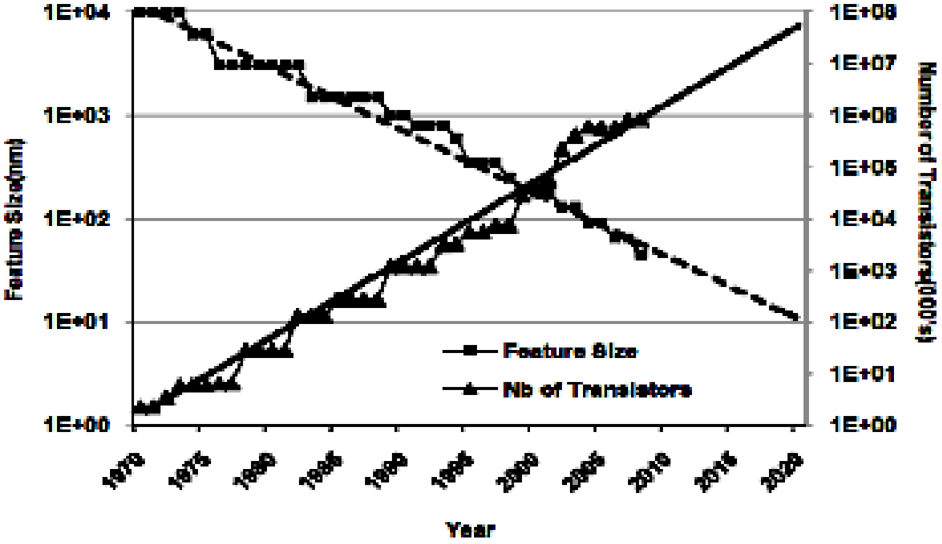
- DUE TO PROCESS IMPROVEMENTS
- DEEPER PIPELINE
- CIRCUIT DESIGN TECHNIQUES

THIS HISTORICAL TREND HAS SUBSIDED OVER THE PAST 10 YEARS

IF IT HAD KEPT UP, TODAY'S CLOCK RATES WOULD BE MORE THAN 30GHz!!!!!

PROCESSOR ARCHITECTURE

- PIPELINING (I.E., ARCHITECTURE) AND CIRCUIT TECHNIQUES HAVE GREATLY CONTRIBUTED TO THE DRAMATIC RISE OF THE CLOCK RATE
  - THE 1.19 CURVE CORRESPONDS TO PROCESS IMPROVEMENTS ALONE
  - REST IS DUE TO ARCHITECTURE AND CIRCUITS
- ADDITIONALLY COMPUTER ARCHITECTS TAKE ADVANTAGE OF THE GROWING NUMBER OF CIRCUITS



New process every 2 year

feature size reduced by 30% every process

or halved every 5 years

Number(b) of transistor doubles every 2 years (Moore's law)

1B transistors reached in 2008

100B in 2021

- A SANDBOX TO PLAY IN SO TO SPEAK
- HOW DO WE USE 100B TRANSISTORS????

CAN THIS TREND CONTINUE?

MEMORY SYSTEMS

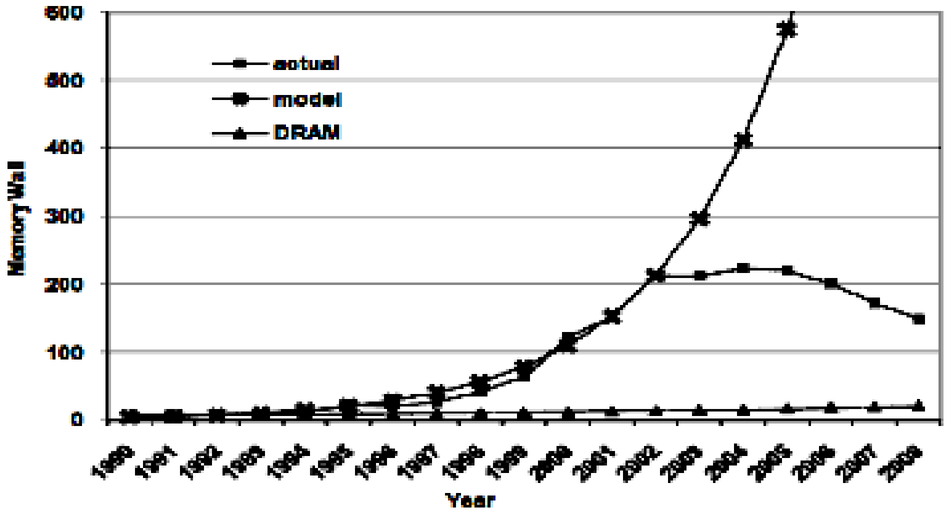
- **MAIN MEMORY SPEED IS NOT GROWING AS FAST AS PROCESSORS' SPEED.**
  - GROWING GAP BETWEEN PROCESSOR AND MEMORY SPEED (THE SO-CALLED "MEMORY WALL")
- **ONE WANTS TO DESIGN A MEMORY SYSTEM THAT'S BIG, FAST AND CHEAP**
  - THE APPROACH IS TO USE A MULTI-LEVEL HIERARCHY OF MEMORIES
  - MEMORY HIERARCHIES RELY ON PRINCIPLE OF LOCALITY
  - EFFICIENT MANAGEMENT OF THE MEMORY HIERARCHY IS KEY
  - COST AND SIZE OF MEMORIES IN A BASIC PC (2008)

Memory	Size	Marginal Cost	Cost per MB	Access Time
L2 Cache (on chip)	1MB	$20/MB	$20	5 nsec
Main Memory	1GB	$50/GB	5c	200 nsec
Disk	500GB	$100/500GB	0.02c	5msec

MEMORY WALL?? WHICH MEMORY WALL??

HISTORICALLY, MICROPROCESSOR SPEED HAS INCREASED BY 50% A YEAR

- WHILE DRAM PERFORMANCE IMPROVED BY 7% A YEAR
  - ALTHOUGH DRAM DENSITY KEEPS INCREASING BY 4X EVERY 3 YEARS
- THIS CREATED THE PERCEPTION THAT THIS PROBLEM WOULD LAST FOREVER
- HOWEVER TRENDS HAVE CHANGED DRAMATICALLY IN THE PAST 6 YEARS
  - THE “MEMORY WALL” (RELATIVE PERFORMANCE OF PROCESSORS VS DRAM)



DRAM: 1.07 CGR

**Memory wall =
memory_cycle/processor_cycle**

**In 1990, it was about 4 (25MHz,150ns).
Grew to 200 exponentially until 2002
Has tappered off since then**

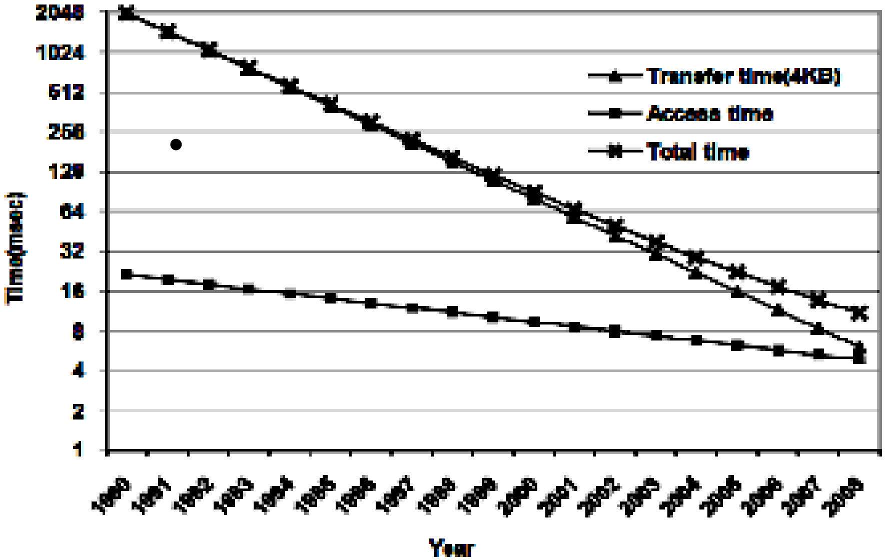
- **ALTHOUGH STILL A BIG PROBLEM, THE MEMORY WALL STOPPED GROWING AROUND 2002.**

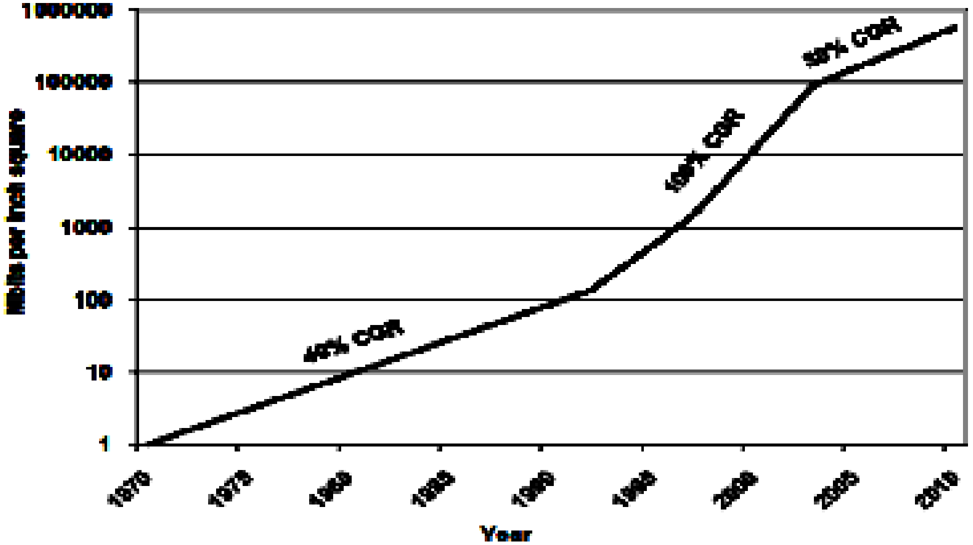
WITH THE ADVENT OF MULTICORE MICROARCHITECTURES THE MEMORY PROBLEM HAS SHIFTED FROM LATENCY TO BANDWIDTH

DISK

- HISTORICALLY DISK PERFORMANCE & DENSITY IMPROVED BY 40%

DISK TIME = ACCESS TIME + TRANSFER TIME





- HISTORICALLY TRANSFER TIMES HAVE DOMINATED
- BUT TODAY TRANSFER AND ACCESS TIMES ARE OF THE SAME ORDER
- IN FUTURE ACCESS TIME WILL DOMINATE (MUCH SLOWER CURVE)

NOTE: ALL THESE TIMES ARE STILL IN THE ORDER OF MILLISECONDS
➔MUST SWITCH CONTEXT

NETWORKS

- NETWORKS ARE PRESENT AT MANY LEVELS
- ON-CHIP INTERCONNECTS forward values from and to different stages of a pipeline and among execution units AND connect cores to shared cache banks.

- SYSTEM INTERCONNECTS connect processors (CMPs) to memory & I/O

- I/O INTERCONNECTS (usually a bus such as e.g., PCI) connect various I/O devices to the system bus.

- INTER-SYSTEM INTERCONNECTS connect separate systems (separate chassis or box) and includes
  - SANs (System-Area networks --connecting systems at very short distances),
  - LAN (Local Area Networks --connecting systems within an organization or a building),
  - WAN (Wide Area networks --connecting multiple LAN at long distances).

- INTERNET. Most computing systems are connected to the Internet, which is a global, worldwide interconnect.

PARALLELISM IN ARCHITECTURES

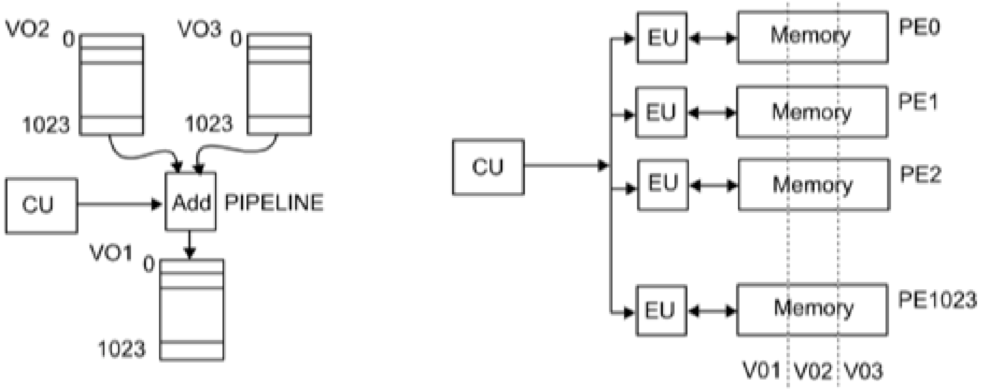
- **THE MOST SUCCESSFUL MICROARCHITECTURE HAS BEEN THE SCALAR PROCESSOR**
  - A TYPICAL SCALAR INSTRUCTION OPERATES ON SCALAR OPERANDS

    ADD O1,O2,O3 /O2+O3=>O1
  - EXECUTE MULTIPLE SCALAR INSTRUCTIONS AT A TIME
    - PIPELINING
    - SUPERSCALAR
    - SUPERPIPELINING
    - TAKES ADVANTAGE OF **ILP**, I.E., INSTRUCTION-LEVEL PARALLELISM, THE PARALLELISM EXPOSED IN SINGLE THREAD OR SINGLE PROCESS EXECUTION
- **CMPs (CHIP MULTIPROCESSORS) EXPLOITS PARALLELISM EXPOSED BY DIFFERENT THREADS RUNNING IN PARALLEL**
  - THREAD LEVEL PARALLELISM OR TLP
  - CAN BE SEEN AS MULTIPLE SCALAR PROCESSORS RUNNING IN PARALLEL

PARALLELISM IN ARCHITECTURES

- **VECTOR AND ARRAY PROCESSORS**
  - A TYPICAL VECTOR INSTRUCTION EXECUTES DIRECTLY ON VECTOR OPERANDS

    VADD VO1,VO2,VO3 /VO2+VO3=>VO1
    - VOk IS A VECTOR OF SCALAR COMPONENTS
    - EQUIVALENT TO COMPUTING
      - VO2[i]+VO3[i]=>VO1[i], i=0,..,N
- **VECTOR INSTRUCTIONS ARE EXECUTED BY PIPELINES OR PARALLEL ARRAYS**



(a) Vector

(b) Array

POWER

- **TOTAL POWER: DYNAMIC + STATIC(LEAKAGE)**

$$P_{dynamic} = \alpha C V^2 f$$

$$P_{static} = VI_{sub} \approx Ve^{-KVt/T}$$

- **DYNAMIC POWER FAVORS PARALLEL PROCESSING OVER HIGHER CLOCK RATE**
  - DYNAMIC POWER ROUGHLY PROPORTIONAL TO F^3
  - TAKE A U.P. AND REPLICATE IT 4 TIMES: 4X SPEEDUP & 4X POWER
  - TAKE A U.P. AND CLOCK IT 4 TIMES FASTER: 4X SPEEDUP BUT 64X DYNAMIC POWER!
- **STATIC POWER**
  - BECAUSE CIRCUITS LEAK WHATEVER THE FREQUENCY IS.
- **POWER/ENERGY ARE CRITICAL PROBLEMS**
  - POWER (IMMEDIATE ENERGY DISSIPATION) MUST BE DISSIPATED
    - OTHERWISE TEMPERATURE GOES UP (AFFECTS PERFORMANCE, CORRECTNESS AND MAY POSSIBLY DESTROY THE CIRCUIT, SHORT TERM OR LONG TERM)
    - EFFECT ON THE SUPPLY OF POWER TO THE CHIP
  - ENERGY (DEPENDS ON POWER AND SPEED)
    - COSTLY; GLOBAL PROBLEM
    - BATTERY OPERATED DEVICES

RELIABILITY

- **TRANSIENT FAILURES (OR SOFT ERRORS)**
  - CHARGE Q = C X V
    - IF C AND V DECREASE THEN IT IS EASIER TO FLIP A BIT
  - SOURCES ARE COSMIC RAYS AND ALPHA PARTICULES RADIATING FROM THE PACKAGING MATERIAL
  - DEVICE IS STILL OPERATIONAL BUT VALUE HAS BEEN CORRUPTED
  - SHOULD DETECT/CORRECT AND CONTINUE EXECUTION
  - ALSO: ELECTRICAL NOISE CAUSES SIMILAR FAILURES
- **INTERMITTENT/TEMPORARY FAILURES**
  - LAST LONGER
  - DUE TO
    - TEMPORARY: ENVIRONMENTAL VARIATIONS (EG, TEMPERATURE)
    - INTERMITTENT: AGING
  - SHOULD TRY TO CONTINUE EXECUTION
- **PERMANENT FAILURES**
  - MEANS THAT THE DEVICE WILL NEVER FUNCTION AGAIN
  - MUST BE ISOLATED AND REPLACED BY SPARE

PROCESS VARIATIONS INCREASE THE PROBABILITY OF FAILURES

WIRE DELAYS

- WIRE DELAYS DON'T SCALE LIKE LOGIC DELAYS
- PROCESSOR STRUCTURES MUST EXPAND TO SUPPORT MORE INSTRUCTIONS
- THUS WIRE DELAYS DOMINATE THE CYCLE TIME; SLOW WIRES MUST BE LOCAL

DESIGN COMPLEXITY

- PROCESSORS ARE BECOMING SO COMPLEX THAT A LARGE FRACTION OF THE DEVELOPMENT OF A PROCESSOR OR SYSTEM IS DEDICATED TO VERIFICATION
- CHIP DENSITY IS INCREASING MUCH FASTER THAN THE PRODUCTIVITY OF VERIFICATION ENGINEERS (NEW TOOLS, SPEED OF SYSTEMS)
-

CMOS ENDPOINT

- CMOS IS RAPIDLY REACHING THE LIMITS OF MINIATURIZATION
  - FEATURE SIZES WILL REACH ATOMIC DIMENSIONS IN LESS THAN 15 YEARS
  - OPTIONS????
    - QUANTUM COMPUTING
    - NANOTECHNOLOGY
    - ANALOG COMPUTING

PERFORMANCE REMAINS A CRITICAL DESIGN FACTOR

PERFORMANCE METRICS (MEASURE)

- **METRIC #1: TIME TO COMPLETE A TASK (T_{exe}): EXECUTION TIME, RESPONSE TIME, LATENCY**
  - "X IS N TIMES FASTER THAT Y" MEANS Texe(Y)/Texe(X) = N
  - THE MAJOR METRIC USED IN THIS COURSE

- **METRIC #2: NUMBER OF TASKS PER DAY, HOUR, SEC, NS**
  - THE THROUGHPUT FOR X IS N TIMES HIGHER THAN Y IF THROUGHPUT(X)/THROUGHPUT(Y) = N
  - NOT THE SAME AS LATENCY (EXAMPLE OF MULTIPROCESSORS)

- **EXAMPLES OF UNRELIABLE METRICS:**
  - MIPS: MILLION OF INSTRUCTIONS PER SECOND
  - MFLOPS: MILLION OF FLOATING POINT OPERATIONS PER SECOND

EXECUTION TIME OF A PROGRAM IS THE ULTIMATE MEASURE OF PERFORMANCE

BENCHMARKING

WHICH PROGRAM TO CHOOSE?

- **REAL PROGRAMS:**
  - PORTING PROBLEM; COMPLEXITY; NOT EASY TO UNDERSTAND THE CAUSE OF RESULTS
- **KERNELS**
  - COMPUTATIONALLY INTENSE PIECE OF REAL PROGRAM
- **TOY BENCHMARKS (E.G. QUICKSORT, MATRIX MULTIPLY)**
- **SYNTHETIC BENCHMARKS (NOT REAL)**
- **BENCHMARK SUITES**
  - SPEC: STANDARD PERFORMANCE EVALUATION CORPORATION
    - SCIENTIFIC/ENGINEEING/GENERAL PURPOSE
    - INTEGER AND FLOATING POINT
    - NEW SET EVERY SO MANY YEARS (95,98,2000,2006)
  - TPC BENCHMARKS:
    - FOR COMMERCIAL SYSTEMS
    - TPC-B, TPC-C, TPC-H, AND TPC-W
  - EMBEDDED BENCHMARKS
  - MEDIA BENCHMARKS

REPORTING PERFORMANCE FOR A SET OF PROGRAMS

LET Ti BE THE EXECUTION TIME OF PROGRAM i:

1. (WEIGHTED) ARITHMETIC MEAN OF EXECUTION TIMES:

$$\sum_i T_i / N \quad \text{OR} \quad \sum_i T_i \times W_i$$

THE PROBLEM HERE IS THAT THE PROGRAMS WITH LONGEST EXECUTION TIMES DOMINATE THE RESULT

2. DEALING WITH SPEEDUPS

- SPEEDUP MEASURES THE ADVANTAGE OF A MACHINE OVER A REFERENCE MACHINE FOR A PROGRAM i

$$S_i = \frac{T_{R,i}}{T_i}$$

- ARITHMETIC MEAN OF SPEEDUPS
- HARMONIC MEAN

$$\bar{S} = \frac{N}{\sum_i^N \frac{1}{S_i}} = \frac{N}{\sum_i^N \frac{T_i}{T_{R,i}}}$$

REPORTING PERFORMANCE FOR A SET OF PROGRAMS

- GEOMETRIC MEANS OF SPEEDUPS

$$\bar{S} = \sqrt[N]{\prod_{i=1}^{N} S_i}$$

  - MEAN SPEEDUP COMPARIONS BETWEEN TWO MACHINES ARE INDEPENDENT OF THE REFERENCE MACHINE
  - EASILY COMPOSABLE
  - USED TO REPORT SPEC NUMBERS FOR INTEGER AND FLOATING POINT

	Program A	Program B	Arithmetic Mean	Speedup (ref 1)	Speedup (ref 2)
Machine 1	10 sec	100 sec	55 sec	91.8	10
Machine 2	1 sec	200 sec	100.5 sec	50.2	5.5
Reference 1	100 sec	10000 sec	5050 sec		
Reference 2	100 sec	1000 sec	550 sec		

		Program A	Program B	Arithmetic	Harmonic	Geometric
Wrt Reference 1	Machine 1	10	100	55	18.2	31.6
	Machine 2	100	50	75	66.7	70.7
Wrt Reference 2	Machine 1	10	10	10	10	10
	Machine 2	100	5	52.5	9.5	22.4

FUNDAMENTAL PERFORMANCE EQUATIONS FOR CPUs:

$$Texe = IC \times CPI \times Tc$$

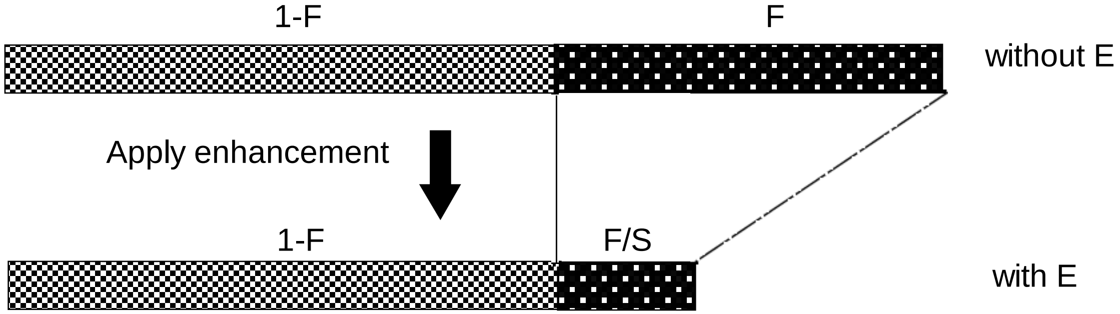
- IC: DEPENDS ON PROGRAM, COMPILER AND ISA
- CPI: DEPENDS ON INSTRUCTION MIX, ISA, AND IMPLEMENTATION
- Tc: DEPENDS ON IMPLEMENTATION COMPLEXITY AND TECHNOLOGY

CPI (CLOCK PER INSTRUCTION) IS OFTEN USED INSTEAD OF EXECUTION TIME

- **WHEN PROCESSOR EXECUTES MORE THAN ONE INSTRUCTION PER CLOCK USE IPC (INSTRUCTIONS PER CLOCK)**

$$Texe = (IC \times Tc)/IPC$$

AMDAHL'S LAW



- ENHANCEMENT E ACCELERATES A FRACTION F OF THE TASK BY A FACTOR S

$$T_{exe}(withE) = T_{exe}(withoutE) X \left[(1-F) + \frac{F}{S}\right]$$

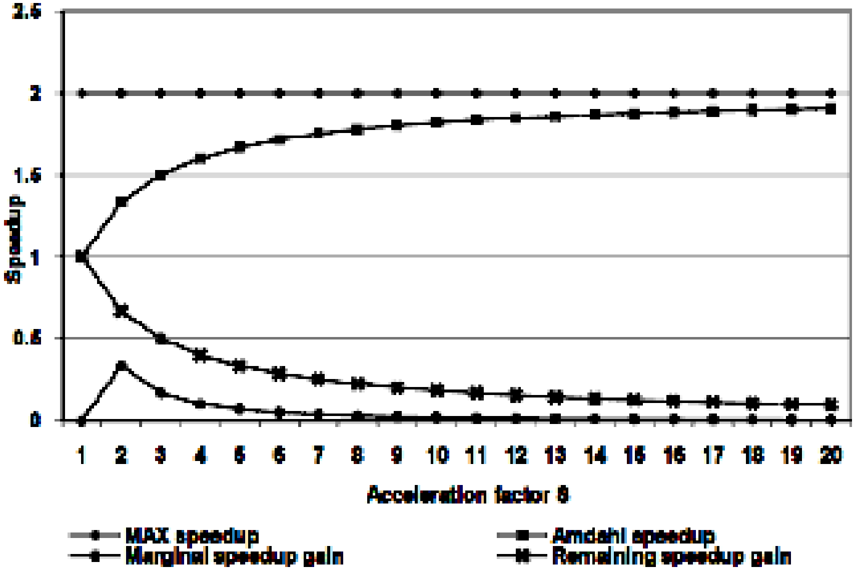
$$Speedup(E) = \frac{T_{exe}(withoutE)}{T_{exe}(withE)} = \frac{1}{(1-F) + \frac{F}{S}}$$

LESSONS FROM AMDAHL'S LAW

- IMPROVEMENT IS LIMITED BY THE FRACTION OF THE EXECUTION TIME THAT CANNOT BE ENHANCED

$$\text{SPEEDUP}(E) < \frac{1}{1-F}$$

  - LAW OF DIMINISHING RETURNS

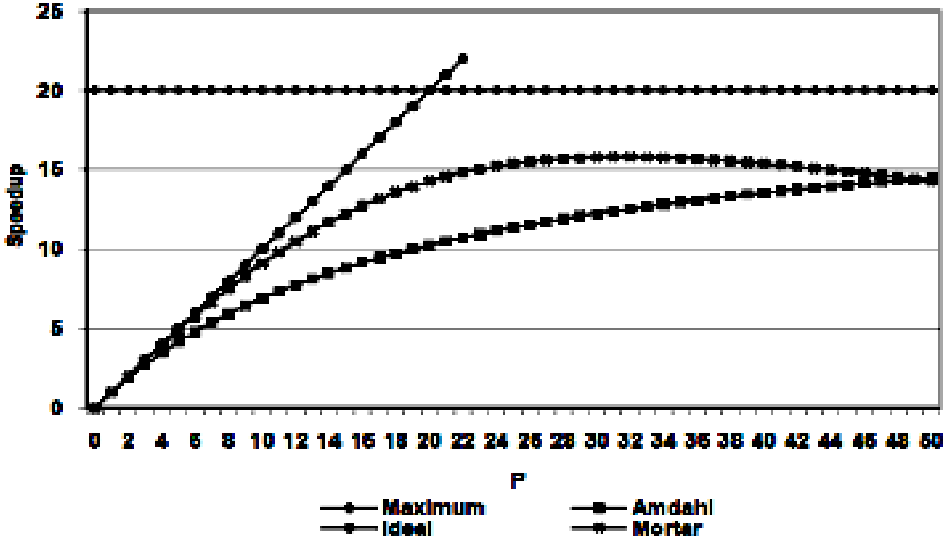


F=0.5

- OPTIMIZE THE COMMON CASE
  - EXECUTE THE RARE CASE IN SOFTWARE (E.G. EXCEPTIONS)

PARALLEL SPEEDUP

$$S_P = \frac{T_1}{T_P} = \frac{1}{1-F+F \S P} = \frac{P}{F+P(1-F)} < \frac{1}{1-F}$$



F=0.95

- NOTE: SPEEDUP CAN BE SUPERLINEAR. HOW CAN THAT BE??

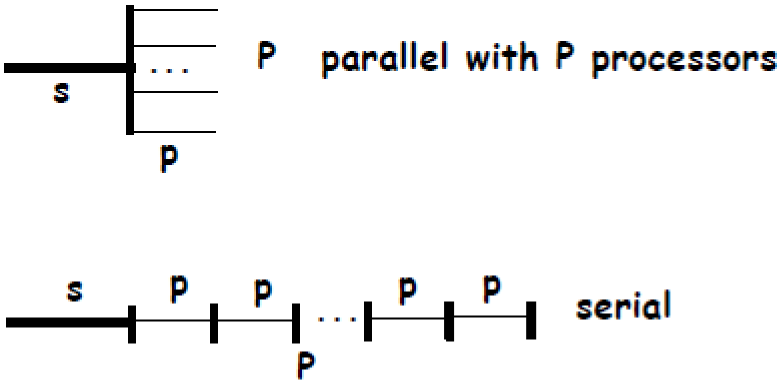
OVERALL NOT VERY HOPEFUL

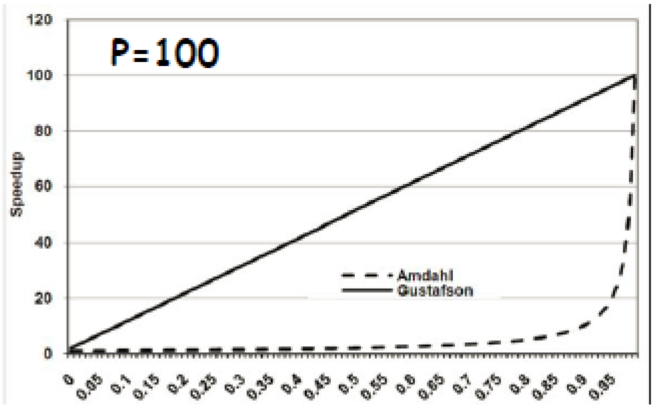
GUSTAFSON'S LAW

- REDEFINE SPEEDUP
  - THE RATIONALE IS THAT, AS MORE AND MORE CORES ARE INTEGRATED ON CHIP OVER TIME, THE WORKLOADS ARE ALSO GROWING
  - STARTS WITH THE EXECUTION TIME ON THE PARALLEL MACHINE WITH P PROCESSORS:

$$T_P = s + p$$

    - *s* IS THE TIME TAKEN BY THE SERIAL CODE AND *p* IS THE TIME TAKEN BY THE PARALLEL CODE
  - EXECUTION TIME ON ONE PROCESSOR IS $T_1 = s + pP$
  - Let $F=p/(s+p)$. Then $S_P = (s+pP)/(s+p) = 1-F+FP = \mathbf{1+F(P-1)}$


s
...
P
parallel with P processors
p
s
p
p
...
p
p
P
serial


P=100
Speedup
Amdahl
Gustafson