Server Architectures: Processors and Memories

January 2005

René J. Chevance

Foreword

- **This presentation belongs to a set of presentations about server architectures.**
  They are based on the following book:

  Serveurs Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, Storage Solutions
  René J. Chevance
  Digital Press December 2004 ISBN 1-55558-333-4
  http://books.elsevier.com/

  This book has been derived from the following one:

  Serveurs multiprocesseurs, clusters et architectures parallèles
  René J. Chevance
  Eyrolles Avril 2000 ISBN 2-212-09114-1
  http://www.eyrolles.com/

  The English version integrates a lot of updates as well as a new chapter on Storage Solutions.

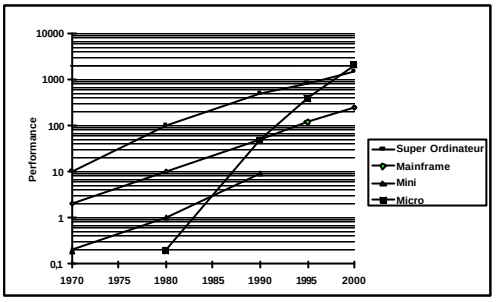
Contact: www.chevance.com **rjc@chevance.com**

Contents

Processors and Memory

Semiconductors and Microprocessors

- **Evolution of Processor Performance**



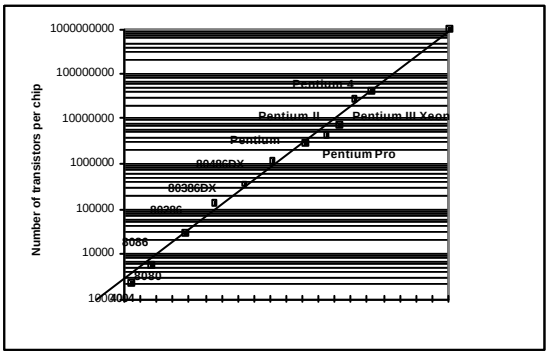
Note: The performance numbers in the chart are given as multiples of the processing capability of a minicomputer of the end of the 1970's. As can be seen on this chart, the growth of microprocessor performance far outpaces that of the other types of processor. It should be noted that the figure plots raw performance, a measure of the intrinsic power of the processors rather than reflecting the processing capability available to actual applications

- **Moore's First Law:**
  - **Density of integrated circuits doubles every eighteen months**
- **Derivation: microprocessor performance doubles every eighteen months**
- **Observation: microprocessor performance doubles every:**
  - **~22 months**
  - **~19 months according to the « Road Maps » of microprocessor vendors**

Page 5

Semiconductors

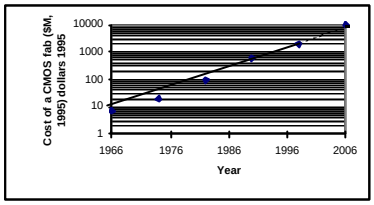
- **Moore's Law illustrated**
  - **Growth of the number of transistors in successive Intel processors (after Intel)**

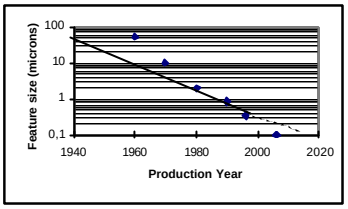


  - ***Similar patterns can be observed for microprocessors from other vendors***

Page 6

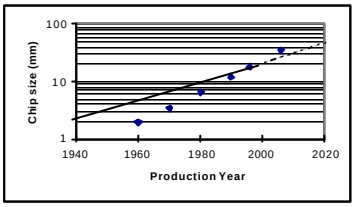
Semiconductors(2)





In October 2001, Intel announced that it had invested $2B in the construction of a new fab at Chandler (Arizona). This fab will construct chips on 8 inch wafers using a 0.13µ technology. Constructing the fab took 18 months and 4200 workers.

For comparison, note that the diameter of a human hair is of the order of 70 µ (70 microns)



If n is the minimum feature size:

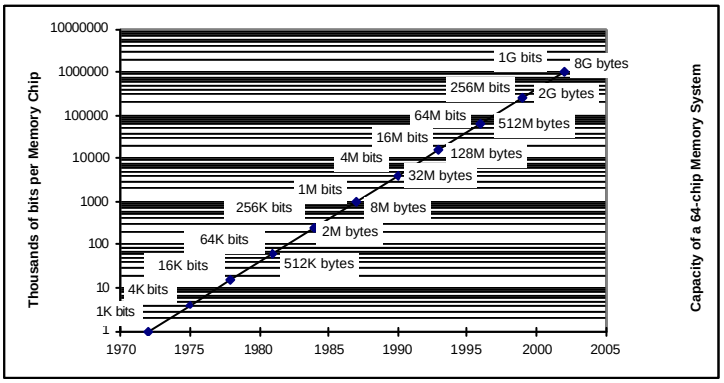
- ***maximum frequency of operation varies as 1/n***
- ***maximum number of devices per unit area varies as $1/n^2$.***

Thus improvement in lithography has a potential for an $1/n^3$ improvement in microprocessor capability.

Page 7

Semiconductors(3)

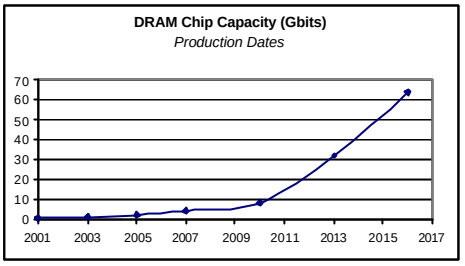
- **Evolution of the capacity of DRAM chips and a 64-chip Memory System**
  - **Memory system built with 64 memory chips (excluding chips for data integrity)**



Page 8

Semiconductors(4)

- **Semiconductor Technology Projections**
  - **DRAM Capacity Projections (Source: [ITR03])**



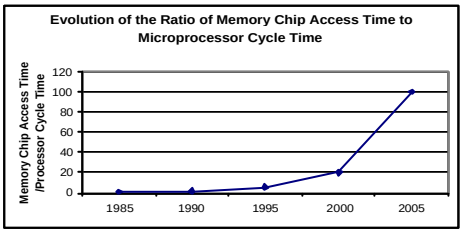
  - **Microprocessor Characteristics Projection (source [ITR03])**

	2004	2006	2008	2010	2012	2016	2018
Transistors (millions)	553	878	1393	2212	3511	8848	14405
Total number of pins	1600	1936	2354	2782	3338	4702	5426
Thermal Dissipation (W)	158	180	200	218	240	288	300
On-chip Frequency (Mhz)	4171	3906	10972	15079	20065	39683	53207

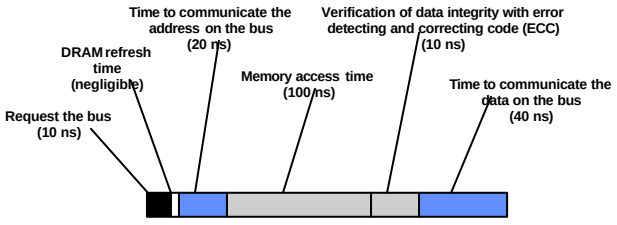
Page 9

Memory Hierarchy

- **Fact: Growing gap between memory chip access time and processor cycle time**



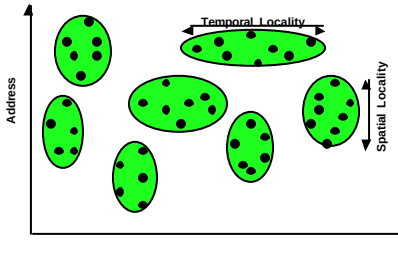
- **Chip access time is just a part of memory access time e.g.**



Page 10

Memory Hierarchy[2]

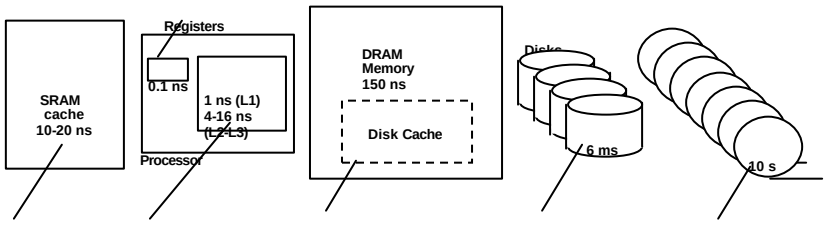
- **Exploiting space-time locality properties**
  - **if a piece of information is accessed, there is a high probability that it will be accessed again in the near future (temporal locality);**
  - **if a piece of information is accessed, there is a high probability that nearby (in memory) information will be also accessed in the near future (spatial locality)**



- **Cache principle : maintaining, in a level of fast memory, the data which are most frequently accessed at a given moment (implemented in hardware for efficiency)**
- **Echange between memory levels on a granule basis (or cache block) e.g. 64 or 128 bytes**

Memory Hierarchy[3]

- **Levels of memory in a system**



- **Illustration of the differences between the characteristics of the various levels of the memory hierarchy**

Technology	Typical Access Time	Human Scale	Approximate Capacity	Approximate Price ($/MB)
Processor Register	100ps	0.1 s	64 x 64 bits	(part of microproces-sor)
Integrated Cache	L1: ~ 1ns L2-L3 4-16 ns depending on cache size	16 s	fraction of a MB up to several MB	(part of microprocessor)
External cache	10-20 ns	~10-20 s	4-8 MB	~$10
Main mem-ory	~150 ns	~25 min	>= 1GB	$0.125
Disk	~6 ms	~700 days	> 70 GB/disk	~$0.005
Tape (in a robot)	~10 s	~ 3200 years	~100GB/tape	< $0.001

Memory Hierarchy[4]

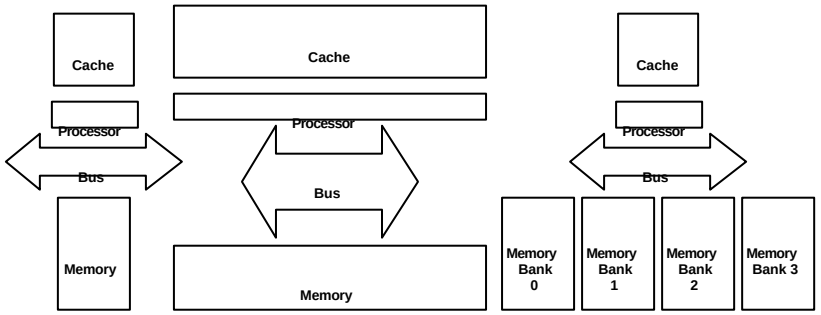
- **Example of memory apparent access time (Level 1 and Level 2 caches, DRAM memory) assuming hit ratios of 95% et 98% (Level 1 and 2 respectively)**

  memory_apparent_access_time = 0.95 x 1 + 0.03 x 10 + 0.02 x 150 = 4.25 ns

- **Cache design parameters:**
  - **Cache block size**
  - **Separate caches or unified cache**
  - **Placement of blocks within a cache**
    - **Fully Associative**
    - **Direct Mapped**
    - **Set Associative**
  - **Virtual or physical addressing**
  - **Block replacement strategy**
  - **Write strategy (Write Through, Write Back)**
  - **Write to a missing block**
  - **….**
- **Cache coherency problems (see SMP)**

Memory Hierarchy[5]

- **Problem: memory throughput demand from microprocessors is growing faster than memory capabilites**
- **Organizations for the improvement of memory throughput:**



(a) - One word wide Memory (b) - Multi-word Wide Memory (c) - Interleaved Memory

- **Design issues:**
  - **Wide memories: modularity, cost**
  - **Interleaved memories: modularity**

Memory Hierarchy[6]

- **Synthesis**

Cache Type and Properties	Level 1 and Level 2	External	Disk cache	External Storage Hierachy
Where it's found	internal to the processor	between microprocessor and main memory	in memory	on disk
Technology	SRAM inte-grated into the microprocessor	SRAM	DRAM	Disk
What's cached	External cache or memory contents	DRAM contents	Disk contents	Contents of tape cartridges in a robot
Characteristics				
Capacity	O(10 KB/100 KB)	O(1 MB)	O(100 MB)	O(1 GB)
Granule size	O(10/100 B)	O(100 B)	O(10 KB)	O(100 KB)
Access time	3 ns	15 ns	~ 180 ns	~ 6 ms
bandwidth	O(GB/s)	O(GB/s)	O(1 GB/s)	O(100 MB/s)
Who manages the cache	Hardware (internal to the microprocessor)	Hardware (internal to the microprocessor)	Software, either the Operating System, the file system or the DBMS	Software: the memory hierarchy manager

Remark: The notation O(N) means "of the order of N"; so that O(100 B), for example, means "of the order of 100 bytes - certainly more than 10, and certainly less than 1000".

Page 15

Microprocessors

Page 16

Microprocessors

- Several years ago, there was a strong debate about RISC (Reduced Instruction Set Computer) and CISC (Complex Instruction Set Computer) approaches. Due to their larger market and the associated revenues, a CISC architecture (x86) was able to recover the intial performance gap with RISC architectures.
- Extension of IA-32 architecture to support 64 bits (both AMD and Intel) will extent the life of this architecture
- Symmetric Multiprocessors have specific requirements on processors:
  - Memory coherence
  - Memory consistency
  - Synchronization mechanisms

  This will be discussed in the SMP presentation.
- Performance benchmarks will be discussed in the « System Performance and Estimation Techniques » presentation

Microprocessor Performance

- Processor Performance Équation

Time/Task = Instructions/Task x Cycles/Instruction x Time/Cycle

  - Contributing factors
    - Instructions per Task:
      - Choice of algorithm, optimizing compiler, ...
      - Architecture suitability to support the application
    - Cycles per instruction:
      - Optimizing compiler, architecture characteristics, architecture implementation (or micro-architecture) with features like pipeline, superscalar, super-pipeline, out-of-order, multi-threading,....)
    - Cycle time:
      - Technology, architecture characteristics, architecture implementation
- Performance improvement through parallelism. Two complementary approaches:
  - At instruction level: ILP (Instruction Level Parallelism), simultaneous execution of instructions (decreasing the cycles/instruction component of the equation)
  - At process/thread level: TLP (Thread Level Parallelism), simultaneous execution of several instruction flows

Microprocessor Performance[2]

- ILP (Instruction Level Parallelism). According to measurements on current architectures, limited to 5 to 6 instructions
  - Possible ways to increase ILP:
    - Creating a new ISA (Instruction Set Architecture) e.g. IA-64 (Itanium) from Intel/HP, VLIW (Very Large Instruction Word) such as Transmeta's Crusoe
    - Improving architecture implementation (micro-architecture). *Note: such techniques apply both to existing ISA as well as to new ISAs*
      - Out-of-order execution
      - Register renaming
      - Speculative execution
      - Branch prediction….
- TLP pour Thread Level Parallelism (improving system throughput)
  - Possible ways (non-exclusive)
    - SMT - Simultaneous MulTithreading
      - Several threads are sharing the same processing resources (e.g. commuting processing from one thread to another one once a thread is waiting for DRAM memory)
      - Example: Pentium 4 (Hyperthreading)
    - MPC - MultiProcessor Chip (several processor cores on the same chip)
      - Several independent processors (cores) one the same chip
      - Example: IBM's Power5 which is a dual core chip, each processor being a 2-way SMT

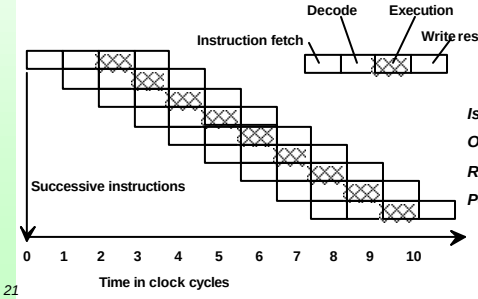
Microprocessor Performance[3]

- **Limiting factors for microprocessor performance:**
  - Data level dependencies
  - Branches:
    - Unconditional branch (target instruction not present in the cache)
    - Conditional branch:target instruction is determined by evaluation of a condition (this is an example of a data dependency)
  - Waits caused by interactions with the memory hierarchy
  - Sequential nature of the instruction flow imposed by many ISAs

Illustration of parallelism in a sequence of instructions

Sequence 1	Sequence 2
load r1<-(r2)	add r3<-r3+1
add r3<-r3+1	add r4<-r3+r2
fpadd r5<-r6+r7	store r4-> (r5)
Level of parallelism = 3	Level of parallelism = 1

Basic Pipelined Processor



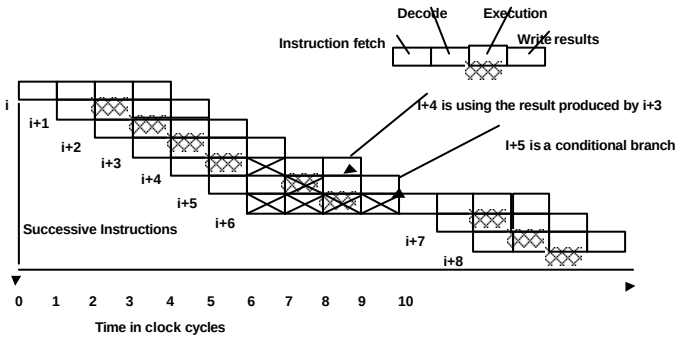
Issue rate = 1

Operation latency = 1

Required parallelism = 1

Potential throughput = 4 x unpipelined processor

Illustration of (two simples cases of) conflicts in a pipeline

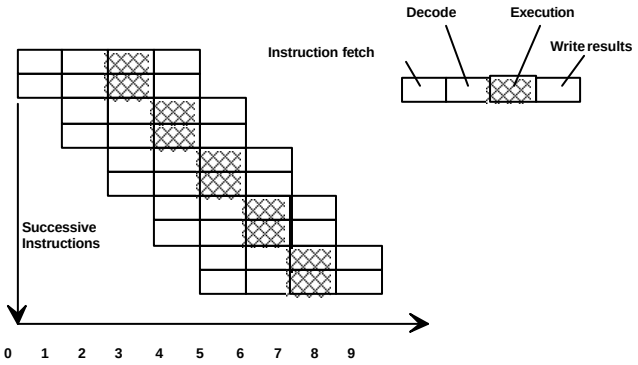


Super-pipelined

- Improving the number of instructions executed per cycle through increasing the number of stages in the pipeline

Micro-architectures[3]

- **Superscalar**



Issue rate = N (2 here)

Operation latency = 1

Required parallelism = 2 (2 here)

Potential throughput = N x basic pipelined processor

Superpipeline/Superscalar Comparison

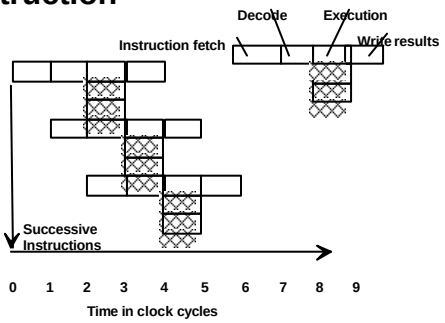
- **Super-pipelined**
  - **Moderate increase in chip complexity**
  - **Critical elements:**
    - **Interfaces with caches and address translation**
    - **Need for storage elements (latches) in front of each pipe stage**
    - **Usually requires high frequencies to get high performance**
- **Superscalar**
  - **Increasing the degree of superscalarity increases complexity super-linearly**

Very Long Instruction Word (VLIW)

- **Principle: Architecture with instructions composed by several primitives operations able to be executed in parallel**
- **Primitive operations within an instruction are determined at compile time:**
  - **Microprocessor complexity, by comparison with a superscalar implementation is decreased since parallelism is detected by the compiler**
- **Difficulties:**
  - **Compilers must extract enough parallelism from source programs (otherwise instructions are composed mostly of no-operations (NOPs))**
  - **Generated code must take into account the degree of parallelism supported by the processor. So, it is necessary to recompile programs or to use a binary translator (detrimental to performance)**

VLIW(2)

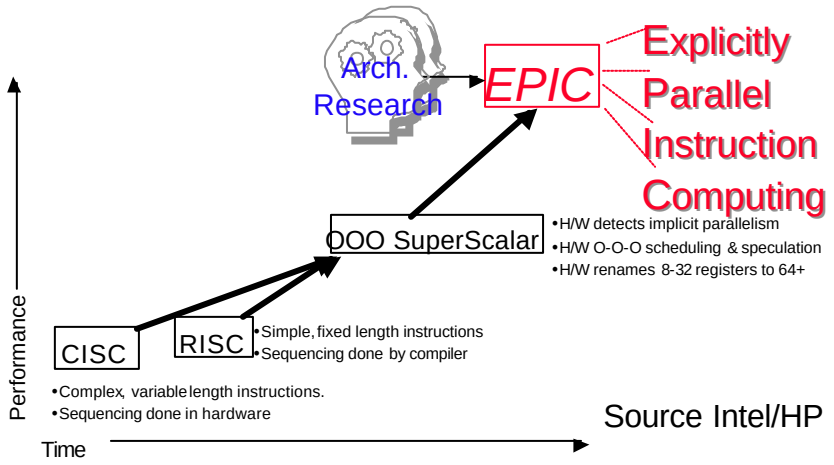
- **Exemple of a VLIW processor with 3 operations per instruction**



- **Few probant experiences for general purpose computing:**
  - **Multiflow (out of business)**
  - **Source of inspiration for IA-64**
  - **Transmeta (IA-32 compatible)**

Itanium

- **Concept**



Itanium(2)

- **EPIC Objectives**
  - **Performance and the potential for continued performance improvement over a long period**
  - **Software Compatibility:**
    - **IA-32 (direct support)**
    - **HP-PA (through code translation)**
  - **Scalability: different implementations may have different levels of parallelism**
  - **Large address space**
  - **Performance: EPIC addresses the following limiting factors:**
    - **Branches (predication)**
    - **Memory latency (speculative load)**
    - **Instruction level parallelism:**
      - **exhibited by the compilers**
      - **instructions are grouped in bundles (3 instructions) and a descriptor tells the processor about instruction dependencies**

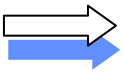
Performance improvement Techniques

- **Overview of performance improvement techniques (at single instruction flow level)**
  - **Speculative execution**
    - **A guess is made about the outcome of a conditional branch and execution proceeds as if the guess were true; to handle the cases when the guess is wrong, the processor has resources which allow it, effectively, to backtrack and try again.**
  - **Out of order execution and register renaming**
    - **Out of order is a technique which aims to minimize the effects of inter-instruction data dependencies and waits caused by cache misses. Register renaming is a companion technique. A machine with out of order execution may execute instructions in an order quite different from that obtained by simple sequential execution of the program (although its internal mechanisms will ensure that the results it gets are the same as sequential execution would have provided, of course).**
    - **Execution of instructions waiting for data is suspended and the execution continue with instructions which follows. Execution of suspended instructions is resumed when data is available**

Performance improvement Techniques(2)

- **Out of order execution**

```
load r1<-x
load r2<-y
add r3<-r6+r7
load r16<-b
store r16->a
store r3->z
add r8<-r1+r2
```

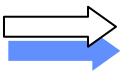


If we assume that either or both x and y are not present in the data cache, the microprocessor starts execution of the following instructions before the memory hierarchy has delivered the values for the first two instructions

- **Register renaming**

Original Instruction Sequence

```
e1 :  load r3, ....
      add r2,r3, ....
      ....
      ....
e2 :  load r3, ...
      sub r3,r4, ...
      ....
```



Mapped sequence – the two sub-sequences can be executed in parallel (architected registers are mapped to implementation registers

```
e1 :  load ir17, ....
      add ir18,ir17, ....
      ....
      ....
e2 :  load ir37, ...
      sub ir37,ir21, ...
      ....
```

Register renaming extends the number of registers (at implementation level) providing capability for parallel execution in an instruction stream. This technique is particularly useful for CISC architectures where the number of architected registers is small.

Performance improvement Techniques(3)

- **Branch prediction**
  - The microprocessor maintains a table (a cache) of most probable target instruction for each conditional branch. The microprocessor is then able to start excution of the probable target instruction without waiting for the test to decide. Of, course, in case of miss prediction, the microprocessor will have to « undo ».
- **Dynamic Code Translation**
  - Instructions are translated into simpler instructions
  - Examples:
    - Intel Pentium Pro and follow-on
    - AMD
    - Transmeta's Crusoe (IA-32 Þ VLIW)
- **Trace cache**
  - Refinement on branch prediction technique: simply identify sequences of instructions which are executed - omitting the branches - and cache the most frequent sequences (called traces, the cache being the trace cache)

Page 31

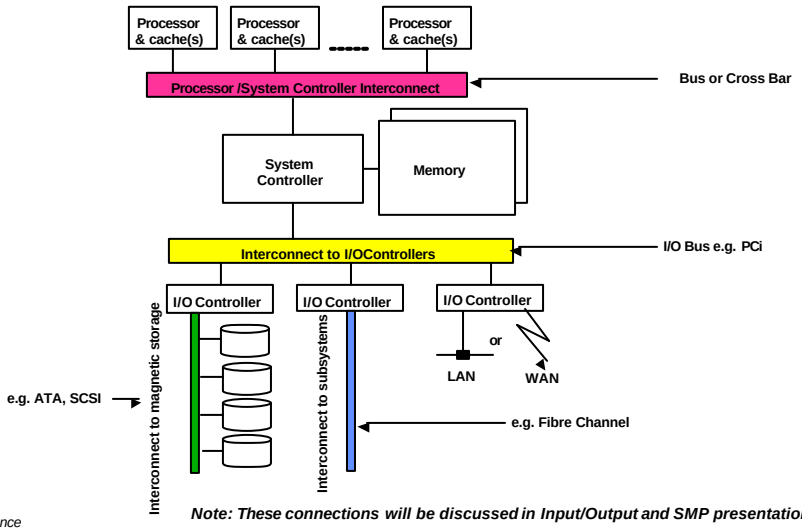
Architecture Evolution Process

- **Looking at the historical record, processor architectural evolution tends to develop as a repeating two-phase phenomenon - a stability phase being followed by a breakaway phase**
  - Stability phase:
    - Concentration on implementation refinement while the architectures remain essentially unchanged
    - Such an evolution does not change the established equilibrium
    - Example: evolution of CISC at the beginning of the 80's
- **In parallel, technology evolution changes the equilibrium and led to a breakaway phase (non-homotetic evolutions):**
  - Evolution of memory capacity and optimizing compilers which led to RISC architectures
- **In a breakaway phase, there is a flock of new ideas , as usual, the market will « make the selection »**

Page 32

Processor-Memory-I/O Interconnect

- **Connections in a generic system architecture**

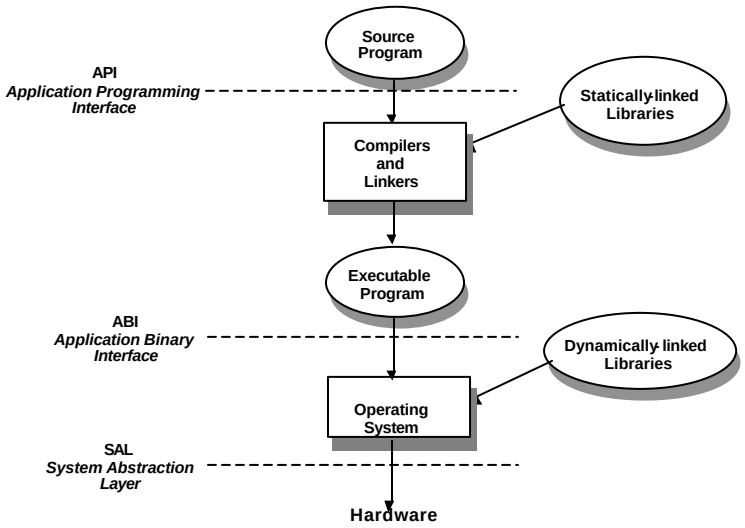


Note: These connections will be discussed in Input/Output and SMP presentations

Binary Compatibility
Java
Architecture Retargeting

Binary Compatibility

- **Levels of compatibility**

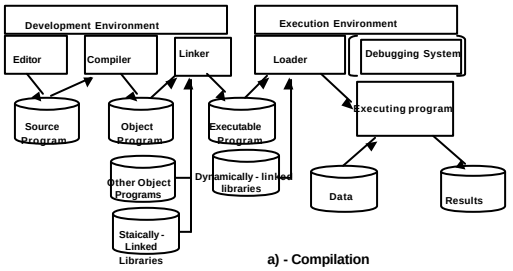


Binary Compatibility(2)

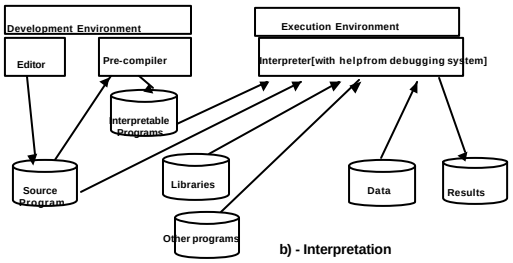
- **Binary compatibility is a consquence of:**
  - **Processor architecture (Instruction Set Architecture)**
  - **OS-imposed addressing conventions**
  - **Interfaces between the application and the OS and with the libraries**
  - **Conventions for data representation**
- **Binary level is the standard level for software distribution**
- **Binary compatibility is an obstacle to the introduction of new architectures and/or new OSes**
- **An attempt to relieve the problems of binary compatibility: JAVA (Sun)**
  - **Derived from C++**
  - **Generation of portable code (called byte code) to interpreted by a virtual machine (JVM for Java Virtual Machine) independent from architectures**
  - **Applications may be downloaded (over a network) on demand:**
    - **Applets (client side applications)**
    - **Servlets (server side applications)**

Binary Compatibility(3)

- **Differences between compilation and interpretation**



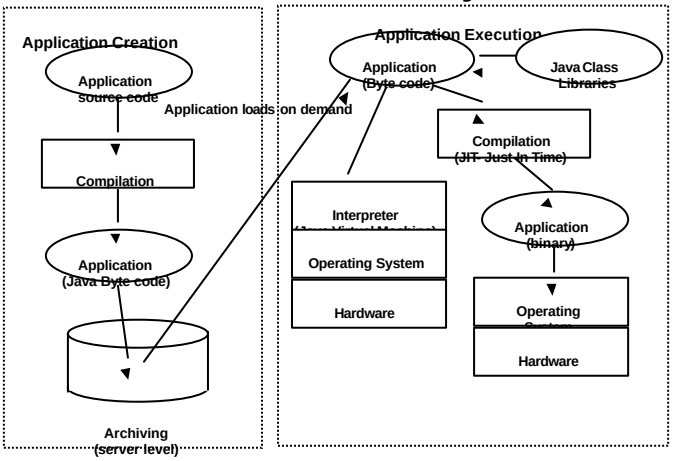
a) - Compilation



b) - Interpretation

Binary Compatibility(4)

- **Java : Write Once, Run Everywhere**

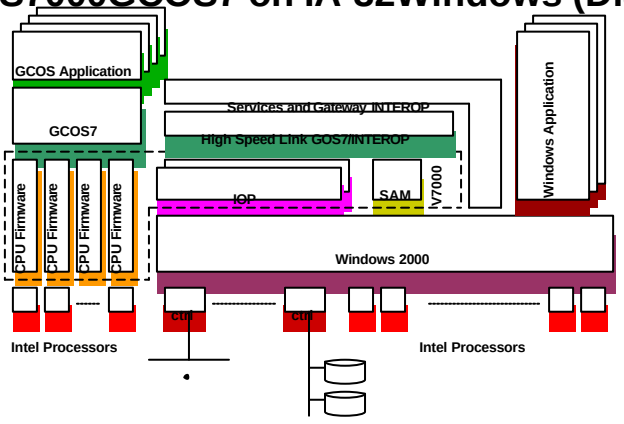


Architecture Retargeting

- **The processing power of current microprocessors allows the support of aplications and OS developed for other architectures**
- **Several approaches have been used:**
  - **Retargeting compilers to a new architecture**
    - **e.g. Tandem (proprietary to RISC),**
  - **Emulation**
    - **Many proprietary architectures**
  - **Binary Translation: static or dynamic (DOCT for Dynamic Object Code Translation)**
    - **IA-32 on RISC or on VLIW (Transmeta)**
    - **Proprietary architectures**

Architecture Retargeting(2)

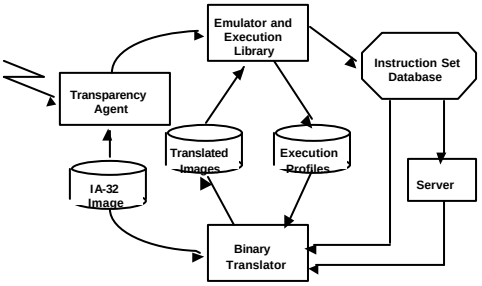
- **Example of emulation: Bull's DPS7000GCOS7 on IA-32Windows (Diane)**



Diane General Architecture

Architecture Retargeting(3)

- **FX!32: IA-32 to RISC (Alpha)**
  - **Because it is impossible for a static analysis to distinguish between code and data in an IA-32 binary, FX!32 uses a mixed approach:**
    - **Program execution begins in emulation mode, which makes it possible to identify the portions of the program which are instructions**
    - **Once the executable instructions are identified, they can be translated into directly-executable Alpha code**

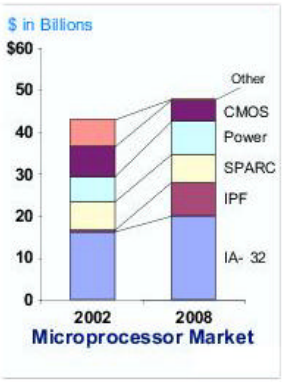


General diagram of FX!32

Economic Aspects of Microprocessors

Economic Aspects of Microprocessors

- **Evolution of the server market share for various architectures (Source Gartner 2003)**

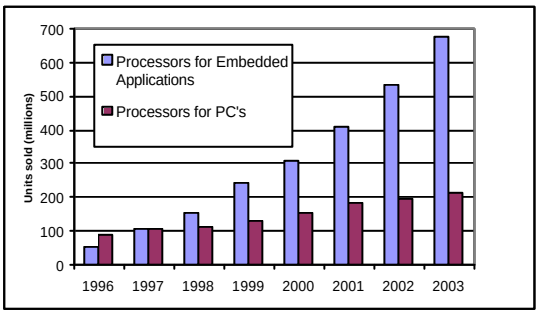


Notes:

- ***By 2003: number of server units sold was 5.3 millions of which IA-32 based servers represented 1.4 million units (IDC)***
- ***By 2003: server revenue was $45.7B (IDC)***
- ***PC market in 2004: 191.4 millions units and $132.8B (ISuppli)***
- ***According to Intel, 1 billion x86 processors have been sold by april 2003 and about 2 billions will have been sold by 2008***

Economic Aspects of Microprocessors[2]

- **32-64 bit Microprocessor Market for embedded applications**



- **8 and 16 bit microprocessors dominates 32-64 bit microprocessors in terms of volumes**
- **Success on one market does not implies success in the other one**
- **The market for embedded microprocessors is expected to reach $8.6B by 2006 (inStat)**

Economic Aspects of Microprocessors[3]

- **Evolution of development costs [HEN99]:**

Microprocessor	Year of Introduction	Millions of Transistors	Number of Developers	Project Duration (months)	Project Headcount Cost Estimate ($M)	Verification Cost (Percentage of the project cost)
R2000	1985	0.1	20	15	2.5	15%
R4000	1991	0,25	55	24	11	20%
R10000	1996	6.8	>100	36	30	>35%

- **To remain a credible contender, a vendor must (historically) introduce a new version every 18 to 24 months.**
- **For a given micro-architecture, it is possible to take advantage of one or two semiconductor technology steps without making significant changes to the micro-architecture. So, a given micro-architecture (for a given general- purpose microprocessor) can therefore expect a competitive life of three or four years.**
- **Furthermore, a new micro-architecture may imply the development of companion chips (e.g. system controllers) and compiler enhancements**
- **Because design and verification is taking increasing amounts of time, and to meet the need of a new design every three or four years a vendor must have several designs proceeding concurrently, but at different phases of their life cycles**

Economic Aspects of Microprocessors[4]

- **Current estimate**
  - **Improving existing micro-architecture: O(10M of $)**
  - **New micro-architecture : O(100M of $)**
  - **New architecture: O(B of $)**
- **Manufacturing cost of a microprocessor O(10$) or O(100$)**
- **Practical consequences:**
  - **Development cost dominates. Only large volumes may amortize development costs**
  - **Industry is going to concentrate over only 2 or 3 architectures**