

The architecture of a computer defines:

the view of the computer from the perspective of an assembly language or machine language programmer.

The instruction set architecture (ISA)

defines the software hardware boundary

It includes:

- the instruction set
- the machine instruction formats
- the available addressing techniques
- the operational register set
- the format of the available data types

The architecture specifies what the computer can do

“Computer Organization” is sometimes used interchangeably with  
“Computer Architecture”  
but they have different meanings

Computer Organization is also called the microarchitecture  
describes how the capability defined by the architecture is  
implemented

Architecture may define 32-bit memory word transfers  
the organization may internally perform two 16-bit transfers

Computer models that share a common architecture may have  
different microarchitectures (i.e., organizations).

The architecture of a clock is defined by the movement of hands on a marked dial to indicate the time

However, one clock may be driven by a wind-up spring, while another, with the same architecture, may be driven by a crystal oscillator

The user may be unaware of the internal timing mechanism

A machine code program can run on different machines with the same architecture without change  
organizational details may differ

- what types of operations are available?  
(integer, floating point, etc.)
- which instructions are allowed to reference memory?
- do operands have to reside in registers?  
or can one or more reside in memory?
- are vector type instructions and vector registers available?
- how many bits do the instruction operands require?
- is a segmented or flat memory model used?
- how many operands can instructions employ?

- are the instructions pipelined?
- is a single-cycle or multi-cycle datapath used?
- how many cache levels are employed?
- is secondary storage provided by magnetic disks or by flash?
- are there multiple buses?
- is the control unit microprogrammed or is hardwired logic used?
- are multiple execution units included?
- how many memory accesses are used to retrieve data or instructions?
- are vector type instructions provided by an array processor or by a vector processor?

Computers with different **organizations**, but with a common shared architecture can execute the same machine code program.

Machine code programs are what the compiler produces when it translates a high level language source program (such as C or C++).

High level languages are designed to abstract away or hide the architecture

The underlying machine architecture is only revealed at the assembly language and machine language levels

Assemblers translate symbolic assembly language into machine code  
Assembly instructions are a higher level representation of machine instructions  
machine instructions are binary patterns understood by the computer hardware

Assembly language instructions and native built-in machine instructions have a one-to-one correspondence

A set of computers that share a common architecture is referred to as a computer family

Examples of computer families include:  
the Intel x86, Motorola 68000 and MIPS processor families

Members of a computer family can all run the same machine language programs although they tend to vary greatly in the speed with which the programs are executed.

Different technologies may be used to implement the various models within the family.

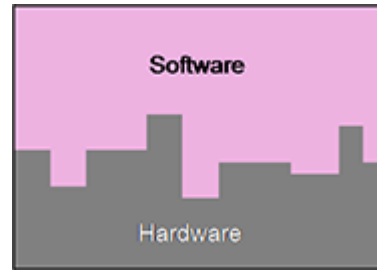
Computer families usually include lower cost lower performance models as well as more powerful, faster and more expensive models.



New members of a computer family perform better  
but execute programs written for older members  
this is called “*backwards compatibility*”

Backwards compatibility is desirable  
runs existing software base on newer more powerful  
models allows for easier hardware upgrades

Programs must be re-compiled to run on a machine with a  
different architecture



Initially programs were written to interact directly with a machine's unique hardware

Each update or improvement in the hardware required rewriting programs that had been previously produced

Machine programs run on all machines with a standard shared ISA



This saves on development time and on cost

manufacturers can innovate and fine-tune the hardware for performance without breaking the existing software base