DATORARKITEKTUR

(Advanced Computer Architecture)

Paul Pop
Institutionen för Datavetenskap (IDA)
Linköpings Universitet

email: paupo@ida.liu.se
http://www.ida.liu.se/~paupo
phone: 28 5628
E++ building

Course Information

Web page: http://www.ida.liu.se/~TDTS51

Examination: written, 19 december 2000 kl. 14 - 18

Lecture notes: available from the web page at least 24 hours before the lecture

Text book: William Stallings: *Computer Organization and Architecture*, 5th edition, Prentice Hall International, Inc., 2000.

Preliminary Course Plan

Lecture 1.

Introduction: Outline, Basic computer architecture and organization, Basic functions of a computer and its main components, The von Neumann architecture.
This is to referesh our memory!

The Memory System: Memory hierarchy, Cache memories, Virtual memories, Memory management.

Lecture 2.

The Memory System: continuation

Lectures 3 and 4.

Instruction Pipelining: Organization of pipelined units, Pipeline hazards, Reducing branch penalties, Branch prediction strategies.

Lectures 5 and 6.

RISC Architectures: An analysis of instruction execution for code generated from high-level language programs, Compiling for RISC architectures, Main characteristics of RISC architectures, RISC-CISC trade-offs.

Preliminary Course Plan (cont'd)

Lectures 7 and 8.

Superscalar Architectures: Instruction level parallelism and machine parallelism, Hardware techniques for performance enhancement, Data dependencies, Policies for parallel instruction execution, Limitations of the superscalar approach.

Lectures 9 and 10.

VLIW Architectures: The VLIW approach - advantages and limitations. Compiling for VLIW architectures. The Merced (Itanium) architecture.

Lectures 11 and 12.

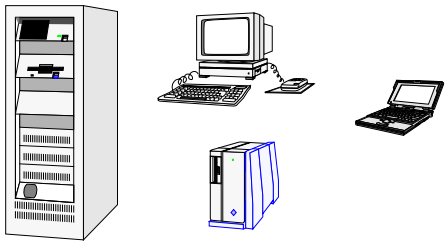
Architectures for Parallel Computation: Parallel programms, Performance of parallel computers, A classification of computer architectures, Array processors, Multiprocessors, Multicomputers, Vector processors.

COMPUTER ARCHITECTURE

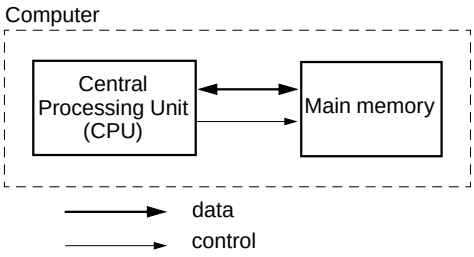
(BASIC ISSUES)

1. What is a Computer/Computer System?
2. The von Neumann Architecture
3. Application Specific vs. General-Purpose
4. Representation of Data and Instructions
5. Instruction Execution
6. The Control Unit
7. The Computer System
8. Main and Secondary Memory
9. Input - Output Devices

What is a computer?

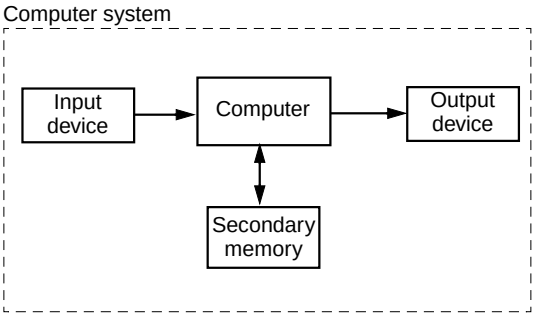


- A **computer** is a data processing machine which is operated automatically under the control of a list of instructions (called a program) stored in its main memory.



What is a computer system?

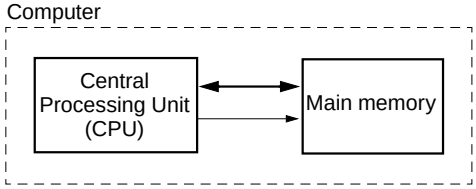
- A **computer system** consists usually of a computer and its peripherals.
- **Computer peripherals** include input devices, output devices, and secondary memories.



The von Neumann Architecture

The principles:

- Data and instructions are both stored in the main memory (stored program concept);
- The content of the memory is addressable by location (without regard to what is stored in that location);
- Instructions are executed sequentially (from one instruction to the next, in order of their location in memory) unless the order is explicitly modified.
- The organization (architecture) of the computer:
  - a *central processing unit* (CPU); it contains the *control unit* (CU), that coordinates the execution of instructions and the *arithmetic/logic unit* (ALU) which performs arithmetic and logic operations;
  - (main) memory.



The von Neumann Architecture (cont'd)

- von Neumann computers are *general purpose computers*.

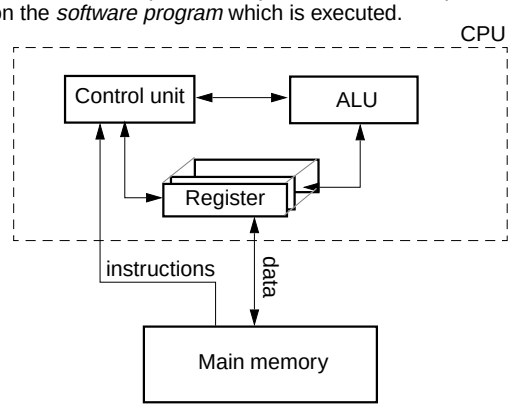
⇩

they can solve very different problems depending on the *program* they got to execute!

- Key concepts here are *program* and *program execution*.

General-purpose (von Neumann) Architectures

In the von-Neumann architecture, a small set of circuits can be driven to perform very different tasks, depending on the *software program* which is executed.



- The primary function of a CPU is to execute the instructions fetched from the main memory.
- An instruction tells the CPU to perform one of its basic operations (an arithmetic or logic operation, to transfer a data from/to main memory, etc.).
- The CU is the one which interprets (decodes) the instruction to be executed and which "tells" the different other components what to do.
- The CPU includes a set of registers which are temporary storage devices typically used to hold intensively used data and intermediate results.

Representation of Data

- Inside a computer, data and control information (instructions) are all represented in **binary format** which uses only two basic symbols: "0" and "1".
- The two basic symbols are represented by electronics signals.

- Numeric data are represented using the **binary system**, in which the positional values are powers of 2:

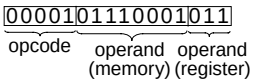
  $100101 = 1*2^0 + 0*2^1 + 1*2^2 + 0*2^3 + 0*2^4 + 1*2^5$

  $10110 = 0*2^0 + 1*2^1 + 1*2^2 + 0*2^3 + 1*2^4$
- Binary numbers are added, subtracted, multiplied and divided (by the ALU) directly; it is not needed to convert them to decimal numbers first.

  100101 + 10110 = 111011

Machine Instructions

- A CPU can only execute *machine instructions*;
- Each computer has a set of specific machine instructions which its CPU is able to recognize and execute.
- A machine instruction is represented as a sequence of bits (binary digits).
  These bits have to define:
  - What has to be done (the operation code)
  - To whom the operation applies (source operands)
  - Where does the result go (destination operand)
  - How to continue *after* the operation is finished



- The representation of a machine instruction is divided into *fields*; each field contains one item of the instruction specification (opcode, operands, etc.); the fields are organized according to the *instruction format*.

Type of Machine Instructions

- Machine instructions are of four types:
  - Data transfer between memory and CPU registers
  - Arithmetic and logic operations
  - Program control (test and branch)
  - I/O transfer

- Important aspects concerning instructions:
  - Number of addresses
  - Types of operands
  - Addressing modes
  - Operation repertoire
  - Register access
  - Instruction format

  } Instruction set design

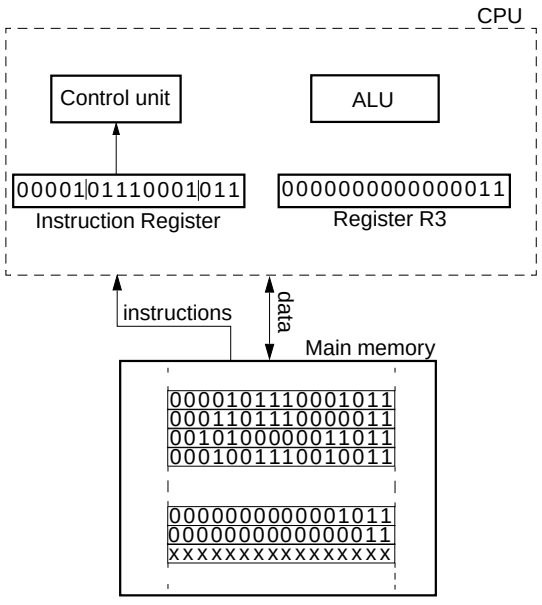
Instruction Execution

The following four instructions perform Z:=(Y+X)*3:

Address	
00001000	00001 01110001 011 (Move addr of Y Reg 3)
00001001	00011 01110000 011 (Add addr of X Reg 3)
00001010	00101 00000011 011 (Mul operand "3" Reg 3)
00001011	00010 01110010 011 (Move addr of Z Reg 3)
.....	.....
01110000	0000000000001011 ← X
01110001	0000000000000011 ← Y
01110010	0000000000101010 ← Z

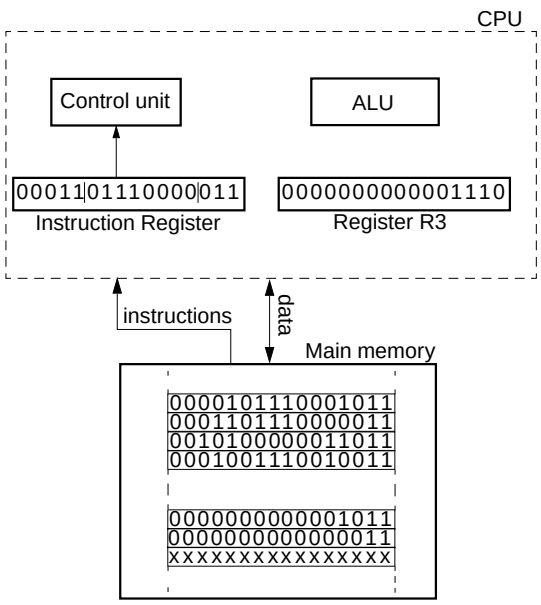
Instruction Execution (cont'd)

First instruction



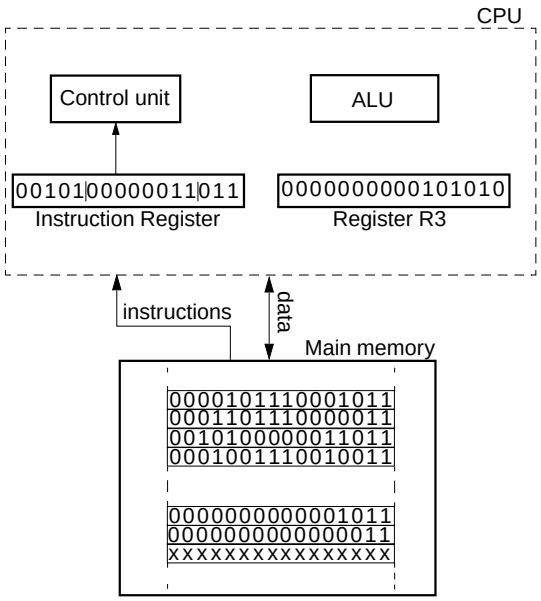
Instruction Execution (cont'd)

Second instruction



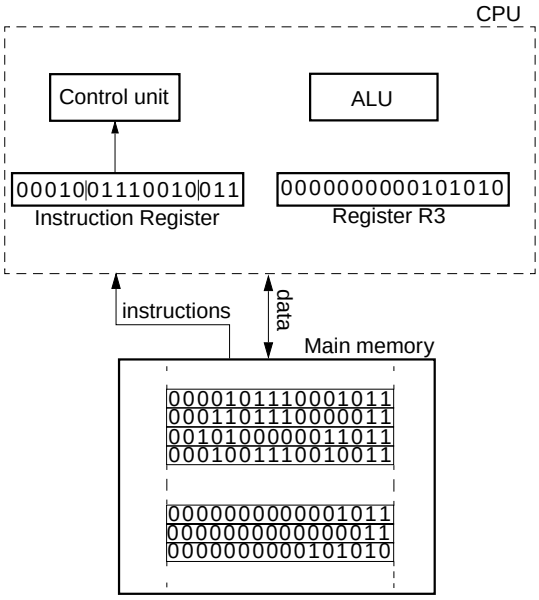
Instruction Execution (cont'd)

Third instruction



Instruction Execution (cont'd)

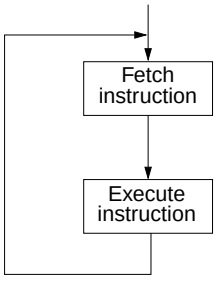
Fourth instruction



The Instruction Cycle

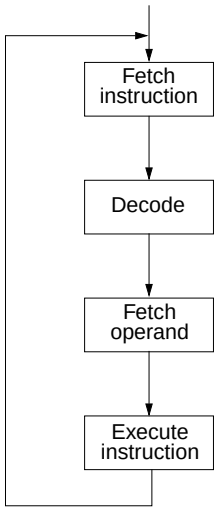
- Each instruction is performed as a sequence of steps; the steps corresponding to one instruction are referred together as an *instruction cycle*.

A simple view of the instruction cycle:

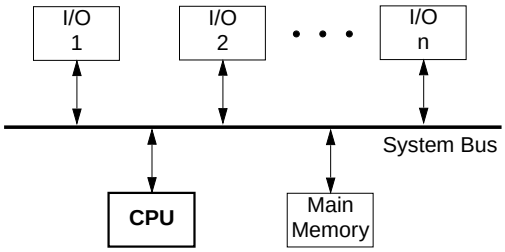


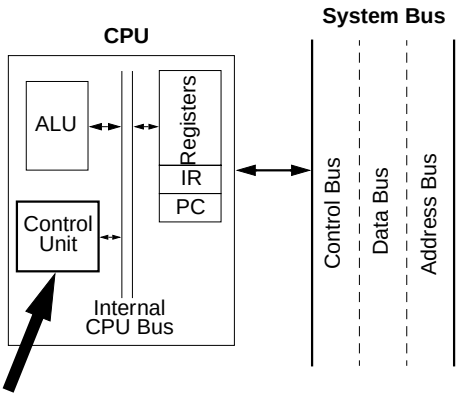
The Instruction Cycle (cont'd)

A refined view of the instruction cycle:



The Control Unit



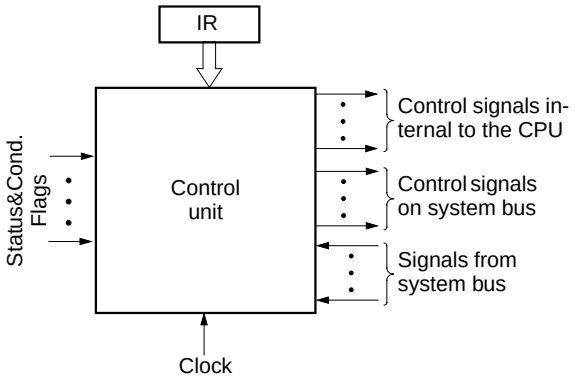


The Control Unit (cont'd)

- How are the elements inside the CPU and the interface to the external datapath controlled (synchronized) in order to work properly?

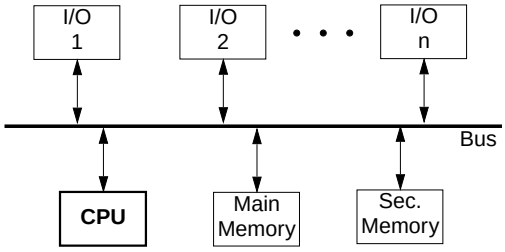
To perform this control, that's the task of the *Control Unit*

The Control Unit (cont'd)



- Techniques for implementation of the control unit:
  1. Hardwired control
  2. Microprogrammed control

The Computer System

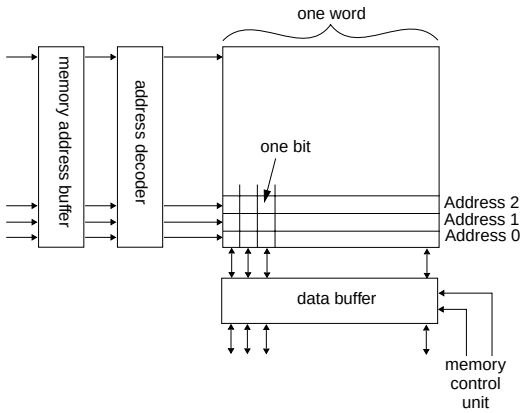


- *CPU* + *main memory* constitute the "core" of the computer system.
- *Secondary memory* + *I/O devices* are the so called peripherals.
- Communication between different components of the system is usually performed using one or several *buses*.

Memories

- The main memory is used to store the program and data which are currently manipulated by the CPU.
- The secondary memory provides the long-term storage of large amounts of data and program.
- Before the data and program in the secondary memory can be manipulated by the CPU, they must first be loaded into the main memory.
- The most important characteristics of a memory is its speed, size, and cost, which are mainly constrained by the technology used for its implementation.
- Typically
  - the main memory is fast and of limited size;
  - secondary memory is relatively slow and of very large size.

The Main Memory



- The main memory can be viewed as a set of storage cells, each of which can be used to store a word.
- Each cell is assigned a unique address and the addresses are numbered sequentially: 0,1,2,... .
- Besides the storage cells, there are a *memory address buffer* (storing the address of the word to be read/written) and a *data buffer* (storing the data read/to be written), the *address decoder* and a *memory control unit*.

The Main Memory (cont'd)

- The most widely used technology to implement main memories is semiconductor memories.
- The most common semiconductor memory type is *random access memory* (RAM).
- The information stored in a RAM semiconductor memory will be lost when electrical power is removed.

Secondary Memory

Hard Disk:

- Data are recorded on the surface of a hard disk made of metal coated with magnetic material.
- The disks and the drive are usually built together and encased in an air tight container to protect the disks from pollutants such as smoke particle and dust. Several disks are usually stacked on a common drive shaft with each disk having its own read/write head.
- Main features:
  - Direct access
  - Fast access:
    seek time ≈ 10 ms
    data transfer rate ≈ 5 MB/s
  - Large storage capacity (8MB - several GB)

Secondary Memory (cont'd)

Diskette:

- Data are recorded on the surface of a floppy disk made of polyester coated with magnetic material.
- A special diskette drive must be used to access data stored in the floppy disk. It works much like a record turntable of gramophone.
- Main features:
  - Direct access
  - Cheap
  - Portable, convenient to use
- Main standards:
  - 5 1/4-inch. Capacity ≈ 360 KB/disk
  - 3 1/2-inch. Capacity ≈ 1.44 MB/disk
    (about 700 pages of A4 text)

Secondary Memory (cont'd)

Magnetic tape:

- Magnetic tape is made up from a layer of plastic which is coated with iron oxide. The oxide can be magnetized in different directions to represent data.
- Its operation uses a similar principle as in the case of a tape recorder.
- Main features:
  - Sequential access (access time about 1-5 s)
  - High value of storage (50 MB/tape)
  - Inexpensive
- It is often used for backup or archive purpose.

Secondary Memory (cont'd)

Optical Memory:

- CD-ROM (Compact Disk ROM): The disk surface is imprinted with microscopic holes which record digital information. When a low-powered laser beam shines on the surface, the intensity of the reflected light changes as it encounters a hole. The change is detected by a photosensor and converted into a digital signal.
  - huge capacity: 775 MB/disk(≈550 diskettes).
  - inexpensive replication, cheap production.
  - removable.
  - read-only.
  - long access time (could be half a second).
- WORM (Write-Once Read-Many) CD: A laser beam of modest intensity equipped in the disk drive is used to imprint the hole pattern.
  - good for archival storage by providing a permanent record of large volumes of data.
- Erasable Optical Disk: combination of laser technology and magnetic surface technique.
  - can be repeatedly written and overwritten
  - high reliability and longer life than magnetic disks.

Input-Output Devices

- Input and output devices provide a means for people to make use of a computer.
- Some I/O devices function also as an interface between a computer system and other physical systems. Such interface usually consists of A/D and D/A converters.

Typical Input Devices

Device	Main features	Advantages	Disadvantages
Keyboard	Like a typewriter	Efficient for inputting text	Relatively slow, speed depends on operator
Light pen	Point at screen	Easy to use	Needs much software to make it versatile
Mouse	Move around on desk	Efficient for icon-based input, and menu selection	Needs much software support
Joystick	Used for games and control	As above Fast	Needs much software support
Graphics tablet	Graphics input	Input picture and freehand sketch	Slow
Scanner	Copy pictures	Fast input of graphics	Bit-mapped graphics only
Voice input	User friendly	No hands needed	Limited vocabulary, Speech recognition software needed

Input-Output Devices (cont'd)

Typical Output Devices

Device	Main features	Advantages	Disadvantages	Speed
Display Screen	Most versatile, both text and graphics	No waste of paper etc.	No hard copy	
Line printer	Impact printer, Very fast.	Can cope with high volume	Large versions are very noisy	up to 6000 cps
Dot matrix printer	Versatile text and graphics	Inexpensive	low quality and speed	up to 200 cps
Inkjet printer	Mechanically similar to above; dot produced by ejected ink droplet	small size; inexpensive	lower quality then laser printers	~20 line/sec
Laser printer	High quality text and graphics	Very fast, high volume	(used to be) expensive	20 000 line/min possible
Plotter	High quality graphics	large graphics output possible	Large machine, expensive	Pen up to 1 meter/s
Voice output	Natural for certain applications	Don't need to use eyes	Limited range of sounds	Normal speech

Summary

- Computer = CPU + Main Memory
  Computer System = Computer + Peripherals
- The CPU executes instructions stored together with data in the main memory.
- Von Neumann computers are general-purpose, programmable computers.
- Data and instructions are represented in binary format.
- Machine instructions are specific to each computer and are organized according to a certain instruction format.
- An instruction is performed as a sequence of steps; this is the instruction cycle.
- The currently manipulated program and data are stored in the main memory. This is organized as a set of storage cells each one having a unique address.
- Secondary memory can be a hard disk, diskette, magnetic tape or an optical device.
- Input-output devices provide a means for people to exchange information with the computer.

What is our Topic in this Course?

We are interested in some advanced issues, typical to modern microprocessors and computer systems.

These advances are at the origin of *high performance achieved with today's computers.*

- Memory hierarchy:
  - cache memory
  - virtual memory
  - memory management;
- Advanced CPU structures and instruction execution strategies:
  - pipelining
  - RISC architectures
  - superscalar architectures
  - VLIW architectures
- System Architectures for parallel computing