Course Information

Web page: http://www.ida.liu.se/~TDDI03

Examination: written, January 12th, 2015, kl. 8 - 12

Lecture notes: available from the web page, latest 24 hours before the lecture.


New from this year: Two big seminars!

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 refresh our memory!!!

Lecture 2 and 3.
The Memory System: Memory hierarchy, Cache memories, Virtual memories, Memory management.

Lectures 4 and 5.
Instruction Pipelining: Organization of pipelined units, Pipeline hazards, Reducing branch penalties, Branch prediction strategies.

Lectures 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.

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.
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. The Intel x86 and ARM Families

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.

The "core" of the computer

Central Processing Unit (CPU)
Main memory

- data
- control

What is a computer (cont’d)?
• Besides the "core" we also have the 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 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*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0
  10110 = 0*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 0*2^1 + 0*2^0
- 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

  0000101110001011
  opcode operand operand (memory) (register)

- 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.
Types 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 Execution

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

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001000</td>
<td>Move</td>
<td>addr Y</td>
</tr>
<tr>
<td>00001001</td>
<td>Add</td>
<td>addr X</td>
</tr>
<tr>
<td>00001010</td>
<td>Mul</td>
<td>operand 3</td>
</tr>
<tr>
<td>00001011</td>
<td>Move</td>
<td>addr Z</td>
</tr>
</tbody>
</table>

01110000 0000000000001011 01110001 0000000000000011 01110010 0000000000101010

Instruction Execution (cont'd)

First instruction

Second instruction

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>00011011</td>
<td>Move</td>
<td>addr X</td>
</tr>
<tr>
<td>00010110</td>
<td>Mul</td>
<td>operand 3</td>
</tr>
<tr>
<td>00010110</td>
<td>Move</td>
<td>addr Z</td>
</tr>
</tbody>
</table>

01110000 0000000000001011 01110001 0000000000000011 01110010 0000000000101010

Instruction Execution (cont'd)

Second instruction

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>00011011</td>
<td>Move</td>
<td>addr R3</td>
</tr>
<tr>
<td>00010110</td>
<td>Mul</td>
<td>operand 3</td>
</tr>
<tr>
<td>00010110</td>
<td>Move</td>
<td>addr R3</td>
</tr>
</tbody>
</table>

01110000 0000000000001011 01110001 0000000000000011 01110010 0000000000101010

Instruction Execution (cont'd)

Second instruction

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>00011011</td>
<td>Move</td>
<td>addr R3</td>
</tr>
<tr>
<td>00010110</td>
<td>Mul</td>
<td>operand 3</td>
</tr>
<tr>
<td>00010110</td>
<td>Move</td>
<td>addr R3</td>
</tr>
</tbody>
</table>

01110000 0000000000001011 01110001 0000000000000011 01110010 0000000000101010

Instruction Execution (cont'd)

Second instruction

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>00011011</td>
<td>Move</td>
<td>addr R3</td>
</tr>
<tr>
<td>00010110</td>
<td>Mul</td>
<td>operand 3</td>
</tr>
<tr>
<td>00010110</td>
<td>Move</td>
<td>addr R3</td>
</tr>
</tbody>
</table>

01110000 0000000000001011 01110001 0000000000000011 01110010 0000000000101010

Instruction Execution (cont'd)

Second instruction

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>00011011</td>
<td>Move</td>
<td>addr R3</td>
</tr>
<tr>
<td>00010110</td>
<td>Mul</td>
<td>operand 3</td>
</tr>
<tr>
<td>00010110</td>
<td>Move</td>
<td>addr R3</td>
</tr>
</tbody>
</table>

01110000 0000000000001011 01110001 0000000000000011 01110010 0000000000101010

Instruction Execution (cont'd)

Second instruction

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>00011011</td>
<td>Move</td>
<td>addr R3</td>
</tr>
<tr>
<td>00010110</td>
<td>Mul</td>
<td>operand 3</td>
</tr>
<tr>
<td>00010110</td>
<td>Move</td>
<td>addr R3</td>
</tr>
</tbody>
</table>

01110000 0000000000001011 01110001 0000000000000011 01110010 0000000000101010
Instruction Execution (cont’d)

Third instruction

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:

- Fetch instruction
- Execute instruction

A refined view of the instruction cycle:

- Fetch instruction
- Decode
- Fetch operand
- Execute instruction
The Control Unit

- 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), 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
  - (Relatively) Fast access
  - Large storage capacity
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 volume of storage
  - Inexpensive
- It is often used for backup or archive purpose.

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
  - inexpensive replication, cheap production
  - removable
  - read-only
  - long access time
- 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.

Flash memory:
- Semiconductor high capacity memory
- Main features
  - Non-volatile
  - Random access
  - Larger delay for write than for read access (erase-before-write)
- Widely used as storage device for mobile systems (e.g. MP3 players, digital camera)
- Replacing hard disk in general purpose computers (PCs).

Case Studies

1. INTEL x86 Family
   - The most successful example of a modern CISC (complex instruction set computer) microprocessor

2. ARM Family
   - The most successful RISC (reduced instruction set computer) microprocessor ever.
The INTEL x86

The number one microprocessor used in non-embedded systems.
First member:
- INTEL 4004 in 1971
First general purpose microprocessor:
- INTEL 8080 in 1974

Milestones

- **8080** (1974)
  - First general purpose microprocessor;
  - 8 bits;
  - Used in first personal computer: Altair.
- **8086** (1978)
  - 16 bits
  - Something like an instruction cache
  - First one used in IBM PC
- **80286** (1982)
  - Huge increase in addressable memory
    (16 MByte instead of 1 MByte)
- **80386** (1985)
  - 32 bits
- **80486** (1989)
  - Complex cache structure and pipelining
  - Math coprocessor

The INTEL x86 (cont’d)

- **Pentium** (1993)
  - Introduces superscalar technology
- **Pentium Pro** (1995)
  - Advanced superscalar techniques
  - Branch prediction and speculative execution
- **Pentium II** (1997)
  - Intel MMX technology (instruction set extension for multimedia)
- **Pentium III** (1999)
  - Additional floating point instructions
  - Support for 3D graphics software
- **Pentium 4** (2000)
  - Further improvements on the line of Pentium III
- **Core** (2006)
  - Solo: single core
  - Duo: Dual core - two processors on a chip
- **Core 2** (2006)
  - 64 bits
  - Dual: two processors on a chip
- **Core 2 Quad** (2007)
  - Four processors on a chip
- **Core i7** (2009/2010), mobile version 2011
  - Six processors, Hyperthreading

Backward compatibility: newer versions can run the programs running on older versions.

The ARM Family

ARM processors are widely used in embedded systems (games, phones, PDAs, multimedia applications, various hand-held devices, consumer products, automotive, wireless etc.).

- High performance, low size/cost, low power consumption

ARM Cambridge, UK, are designing single and multiprocessor architectures and licence them to manufacturers.

Milestones

- **ARM1** and **ARM2** (1985)
  - 32 bit RISC processor
  - 3 stage pipeline
- **ARM3** (1989)
  - cache memory
- **ARM6** (1992)
  - Floating point unit
- **ARM7** (1994)
  - Most successful family.
  - First used as part of complete Systems on Chip (SoC)
- **ARM8** (1996)
  - 5 stage pipeline

The ARM Family (cont’d)

- **ARM9** (1997)
  - separate data/instruction cache
- **StrongArm** (1996)
  - Special version of ARM9; developed with DEC and, later, bought by Intel.
- **ARM10** (2000)
  - 6 stage pipeline
- **ARM11** (2002)
  - 9 stage pipeline
  - Media processing extensions
- **XScale** (2002)
  - Successor to StrongArm, by Intel
  - 7/8 stage pipeline
  - Dynamic voltage & frequency management
- **Cortex** (2005)
  - 13 stage pipeline
  - superscalar
- **ARM11 MPCore** (2005)
  - Multicore based on ARM11; up to 4 cores
- **Cortex-A9 MPCore** (2009)
  - Multicore based on out-of-order superscalar Cortex-A9 core; up to 4 cores
- **Cortex- A15 MPCore** (2011, commercial 2012)
  - Multicore based on an out-of-order superscalar Cortex-A15 core; 2 clusters, 4 cores each.
**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.
- Two important examples: INTEL x86 and ARM

**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
  - performance of parallel computers
  - parallel computer architectures
  - multicore and multithreaded processors
  - general purpose graphic processors