### SDD Notes

### 31/1

* **Algorithm:** A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
  + Set of events = outcomes
  + Follow certain rules
  + Preconceptions
  + Accuracy is key to making show the outcome is correct
* **IPO:** Examples are breathing; lungs process oxygen; Oxygenated blood, carbon dioxide produced.
  + Input
  + Process
  + Output
  + It is a system
* **Abstractions:** a technique for arranging complexity of computer systems. It works by establishing a level of simplicity on which a person interacts with the system, suppressing the more complex details below the current level.

### 1.2.18

#### Control Structures

* Sets the way code flows with statements like, if, and, if else, else id etc.
* algorithms that control a program.

#### Sequence

* Sequence is assumed will be sequential unless directed.
* Code does a certain sequence.

#### Types of Control Structure

* Selecting
* Binary
* Multiway
* Sequence
* Loop/Repetition

### ****7.2.18****

8.1.1 social and ethical issues

When you are planning to build some software - apart from the functionality - what are the questions you need to ask yourself?

#### Questions to consider

* What type of user interface is it going to have?
* Where is it going to be accessed?
* Who is going to use it?
* What is the impact of using it?
* What concerns and issues do you need to care about?

#### Evolution of software application

Before the advent if the personal computer, hardware and software were intimately linked.

* Software applications were written for large military or business organisations
* The hardware and software were purchased together as total solution.
* This is still the case with large business systems.
* Personal computers altered the connection between hardware and software purchases
  + In this section, we examine the origin of some common groups of software and the effect of their evolution on the design features present in today’s software
  + We shall examine the origin of the following influential software areas whose evolution has had a profound impact on the design of software products we use today:
    - Command line and Graphical User Interface (GUI)
    - Internet Applications
      * Email
      * Web Browsers
      * Search Engines
      * Social Networking applications
    - Spreadsheets
* The first type of computing was mechanical
* It moved into electronics like computers, and into pilots *GUI* etc.
* Several developing tools being used to day like c++ etc.

1. Selection -> binary

->

2. loops/repetition

Pretest

Protest

Counting

1. sequel

### 8.2.18

#### Command line vs graphical user interface

* Command line interfaces (CLIs) are text based.
* User presented with a prompt where they can enter commands or inputs into system.
* Users need to be familiar with the syntax.
* Graphical user interfaces (GUIs) have largely replaces CLIs for most users,
* ClIs remain popular when performing specialised or automated tasks. For example many servers such as web and database servers are administered using a CLI.
  + In these cases the user has extensive expertise and knowledge of available commands.
  + For these usersinteraction via CLI is more efficient compared to using a GUI and hence the additional overhead of GUI is unnecessary in terms of usability.
* Using a CLI, a series of commands can be stored within a text (batch\_ file and replayed many times
  + For example all the commands required to back up the databases stored in a server could be saved in a text file.
  + This file can the be replayed to create daily back ups.

#### Graphical User interfaces GUI

* Started during advent of Personal Computers in the 1970’s
  + 1970’s - 80’s Xerox researches and experiments with GUI interfaces - not successful
  + 1970’s - Jobs and Wozniak started Apple computers and invent the Apple 1
  + Bill gates was in awe of Macintosh. Starts Microsoft. Uses GUI to start Windows in 1985.
  + Windows doing better than Apple so Apple commence legal battle, ending with Apple allowing Microsoft a license to use the GUI interface.

### 12.2.18

#### Email

* The origins of email predate the internet. During the early 1960’s electronic mail messages were sent between terminals connected to the same mainframe. During the 1960’s mainframes operated by a variety of organisations were beginning to be connect together and hence emails messages could be sent between users of these systems.
* Started in the 60’s when messaging between 2 connected mainframes
* During 70’s as internet spreading- so the @ symbol was invented/introduced to separate the username and the email host machine
* This has not changed since then
* Only changing is the formatting of email content – allowing us to include attachments as well as rich formatting (HTML) – specified by MIME encoding (multipurpose Internet mail extensions).
* Emails are sent via simple Mail transfer protocol (SMTP)
* Emails are received via (POP) or the IMAP (Internet Message Access Protocol)
  + POP – Copies emails from the central email server and puts a copy locally on your email client(option to delete from central server on download)
  + IMAP – Synch’s your local client to the web server.

#### Web Browsers

* The first web browser was created by Tim Berners-Lee
* It was created in the 1990s
* The first web Browser was Mosaic it ran on a specific platform known as UNIX. (Mosaic A.K.A Mozilla Firefox)
* Under browsers were under development such as the open source code based Amaya, which is still in existence today.in 1994 the web really took off and by the end of it nearly 3 million servers were operating.
* Mosaic Netscape owned more than 80 percent of the browser market. Because of Netscape’s educational roots their browsers was freely distributed to students, teachers and researcher.
* Windows 95 released Microsoft Network (MSN) client software included, the aim being used to establish the Microsoft Network as a Parallel network to the World Wide Web
* Netscape’s Navigator 2 included Email, Frames, Support for progressive Jpegs, Java support as well as SSL (Secure Sockets Layer) encryption. Microsoft Internet explorer 2 existed with a small yet significant following.
* When Internet explorer 3 was released Microsoft offered it for free.

#### Search Engines

* In 1990’s there was no World Wide Web. Instead it was a large amount of files spread across a vast network.
* The method to send this information was FTP (File Transfer Protocol). They acted as repositories for files.
* To find a file you had to know that exact file location.
* Archie = Grandfather of Search Engines.
* Over the next 7 years different versions of search engine were released google was the last. And it is as well by far the most popular search engine.
* Search Engines
  + Archie- Search engine- 1990
  + Veronica (and Jug head) - Search engine- 1993
  + WWW Wanderer- Search engine- 1993
  + ALIEWEB- Search engine- 1993
  + Excite- Search engine- 1993
  + Galaxy- Search engine- 1994
  + Yahoo- Search engine- 1994
  + Web Crawler- Search engine- 1994
  + InfoSeek- Search engine- 1995
  + AltaVista- Search engine- 1995
  + HotBot – Search engine-1996
  + Metacrawler- Search engine – 1996
  + Google – Search engine – 1997

#### A Web Crawler

A Web crawler starts with a list of URL’s to visit, called the *seeds*. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies. If the crawler is performing archiving of web sites it copies and saves the information as it goes. The archives are usually stored in such a way they can be viewed, read and navigated as they were on the live web, but are preserved as ‘snapshots'.

### 14.2.18

Algorithms

5 control structures

1. Repetitions –loop
   1. ‘For’ counting loop
   2. ‘while’ – pre test
   3. ‘repeat’ – post test
2. Selection
   1. Binary
   2. Multiway
3. Sequence

### 16.2.18

#### Intellectual Property

Rights and responsibilities of software developers Rights of the software developer refer to the entitlements that are due to them. Payment for licensed software, credit for work done or trusting that the software project is used in a legal and ethical way. Responsibilities of software developer refer to the need for the developers

to be accountable for the product that has been developed. Good quality

product that is free from viruses and able to meet the customer’s needs.

The customer has the right to expect that problems encountered will be

fixed by the developer.

#### Authorship

Refers to the ownership or origin of a particular piece of work. The author of a piece of software, or the organization for which the author is working, has exclusive rights to designate how and where the program will be used.

#### Reliability

The customer can expect that the product functions as specified. Reliability is the ability of a program to constantly perform the required function understated conditions for a specified time. Computer software failure can always be traced to problems with the way it has been coded.

#### Quality

A good quality product will meet the program requirements in an efficient way, eg on time, on budget and the software is accurate (works properly).

Quality will be enhanced by: ensuring specifications meet the users needs

using a structured approach using accurate and detailed documentation

the enforcement of standards and conventions conducting project reviews

thorough testing

#### Response to problems

Customers will expect software to be reliable and perform the task for which it was purchased. If the software fails to meet expectations, or does not work as specified, the customer can expect the software developer to either replace the software, provide a refund or inform the

customer as to how to use their software correctly.

#### Code of conduct

It is the software developers right to expect that the software product will be used legally and appropriately. It is a standard practice of many companies to ask employees to sign a code of conduct concerning their use of software within the organization. The code of conduct is an official agreement describing ethical and moral obligations of the employee.

#### Spread sheets and presentation

* The idea or aim of a spreadsheet is to perform religious and laborious calculations for the user.
* The first spreadsheet was known as VisiCalc and was written for Apple II computers.
* The copyright laws back then were not as they are today so when Dan Bricklin and Bob Frankston idea of VisiCalc was copied they were only able to get a few royalties.
* What’s still present in today spreadsheet applications;
  + Input, processing and output all merged into a single interface
  + Scrolling ability of the window. Left, Right, Up and Down
  + Instant recalculation of cells as contents change
  + Inclusion of status and or formula line
  + Ability to replicate a range to any other range
  + Relative and absolute referencing
  + Formulas could be entered using minimal keystrokes
  + Cursors are used to select cells and ranges

### Work

#### Intellectual property

* All property is protected by law
* When you purchase any property, whether it is physical goods or software, the law protects the original design.
* E.g. just because you purchase a desk lamp it does not give you the right to claim it as your own property.
* Intellectual property is the property resulting from the fruits of mental labour. The intellectual property laws cover the designs of most products.

#### Software License Agreement

* Software license are intended to enforce the intellectual property rights of software developers. These licences are enforceable by law.
* License agreements also protect developers from legal action should their products result in hardship or financial loss to purchasers.

##### License Terminology

* License agreements result in a legal contract between the purchaser and the seller, it is important to understand the use of the terminology used.
* **License –** Formal permission or authority ti use a product. In most software licenses they are almost non-exclusive; this means it can be licensed to multiple users.
  + The license does not give the user ownership of the software rather it gives them the right to use the software.
* **Agreement –** an agreement is an arrangement or contract between two mutual parties.
  + Installing pre-packaged software often indicates acceptance and agreement between the two parties.
    - **Term-** the period the agreement is enforced. In most cases it is enforced immediately once the terms and conditions have been agreed to.
    - **Warranty-** this is the assurance of some sort – a guarantee. This time in which the product is covered under warranty any product failure can be fixed or replace by the company for cheaper price and or free.
    - **Limited use-** software licenses do not give users unrestricted use of a product. Commonly usage is restricted to a single machine. Copying of the program is not permitted unless for archival back up purposes.
    - **Liability-** an obligation or debt as an consequence of an event. License agreements normally restrict the liability of the software developer by replacing the product or refunding the purchase price.
    - **Program-** refers to computer software (all data files).
    - **Reverse engineer-** in terms of software, this means the process of decompiling the product.
    - **Back up Copy-** A copy of the software made for archive purpose of backing up information in case of media failing. If product license is destroyed, so are the backup copies.

### 20.02.18

#### Questions to consider

When you are planning to build software- apart from functionality- what are the questions you need to ask your self?

1. What type of User Interface is it going to have?
2. Where is it going to be accessed?
3. Who is going to use it?
4. What is the impact of using it?
5. What concerns and issues do you need to care about?

#### Intellectual Property

* Property resulting from the fruits of mental labour.

Law protects all property. Essentially, when you purchase any piece of property, whether it is physical goods or software, law protects the original design.

Intellectual property is property resulting from the fruits of mental labour.

Intellectual property laws cover the design of the products. Laws have now been passed in most countries to protect the intellectual property rights of software developer. These aims to encourage the development if software by ensuring software developers are financially rewarded for their intellectual efforts.

#### Software license agreements

Software licences are intended to enforce the intellectual property rights of software developers.

These are enforceable by law.

Legally correct if the agreement is to stand up to scrutiny of the courts.

Licence agreements protect also against legal action of customers.

#### Licence terminology

Because of the legal nature of licence agreements, they often use the terminology that is difficult to understand.

It is important to know the terminology used.

* **Licence** is a formal permission or authority to use a product
* In relation to software, licences are non-exclusive (can be given to multiple users.)
* **Agreements** are mutual arrangements or contracts between parties.
* Acceptance of software licence agreement.
* **Term** is the period of time where the agreement in in force.
* **Warranty** is an assurance of some sort - a guarantee.
* Usually contain limited warranties.
* **Limited use** are software licenses that do not give purchasers unrestricted use of product.
* Restricted to a single machine, copying not allowed at all.
* **Liability** is an obligation or debt as a consequence of some event.
* Licence agreements normally restrict liability of software agreement ie replacing or refunding product.
* **Program** refers to the computer software.
* This usually includes both executable files and included data files.
* **Reverse engineer** - refers to decompiling the code and attempting a copy.
* Most agreements do not allow licences to reverse engineer a product.
* **Backup copy** is a copy of the software made for archival purposes.
* Should only be used in event of original product failing.

#### Intellectual Property

* **Intellectual property** rights are protected using the laws of copyright
* Software licence agreements are used to create formal contracts that allow the laws of copyright to be better enforced
* There is no need to formally register software products for them to be covered by copyright laws.
* Coverage is automatic for all intellectual property, however selling or changing the owner of the copyrights for a product requires a written contract

Products that no not contain licence agreements or copyright notices are still covered by copyright laws

#### Legal Aspects

* Intellectual property rights are protected using the laws of copyright
* Software licence agreements are used to create formal contracts that allow the laws of copyright to be better enforced
* There is no need to formally register software products for them to be covered by copyright laws.
* Coverage is automatic for all intellectual property, however selling or changing the owner of the copyrights for a product requires a written contract
* Products that no not contain licence agreements or copyright notices are still covered by copyright laws
* The copyright act was created in 1968

#### Copyright Laws

* Shareware:
  + Covered by copyright
  + Copies can be made for archival or distribution purposes
  + The product cannot be modified or reverse engineered
  + Source code is free
* Freeware:
  + Covered by copyright
  + Copies can be made, distributed and altered
  + Modified products must also be freeware
  + Source code may or may not be distributed along with the exe code
* Public domain
  + Not covered by copyright
  + Copies and modifications can be made
* Open source licence
  + Although covered by copyright law, open source licences, such as the GNU GPL (general purpose licence)
  + Modifications can be made and distributed
* Site license
  + Covered by copyright
  + Can be found for a number of machines
  + Used to extend commercial or shareware licenses si it can be used n multiple machines more than once
* Creative commons license
  + Alters how copyrighted materials maybe used without charge. Not recommended for most software products as they do not deal with distribution or source code.
  + Commonly used for artistic works such as photographs
  + Although the conditions of creative commons licenses can be modified to suit specific needs, most permit the work to be freely copied and distributed as long as the original creator is acknowledged

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Licence Type** | **Copyright applicable** | **Source Code provided?**  **Any Restrictions** | **Copies Made and Stored Allowed?** | **Modifications Allowed?** | **Copies May Be distributed?** |
| Commercial | Y | N | Y  Backup only | N | N |
| Shareware | Y | N, Sometime | Y  Backup | N | Y |
| Freeware | Y | Optional | Y | Y | Y |
| Public Domain | N | Optional | Y | Y | Y |
| Open Source | Y, but may be removed through GPL | Y  Credit to the original author in all modified versions | Y | Y | Y |

### 21.02.18

### Social Context of Software Design ergonomics

Ergonomic issues regarding Software Design:

* Effectiveness of screen design
* Ease of use
* Appropriate messages to the user
* Consistency of the user interface

**Inclusivity**

* The need for software to not exclude individuals or groups based on charachteristics as:
  + Cultural Background
  + Economic Background
  + Gender
  + Disability

#### Requirements for a system

* Function Requirements (use and function standards met)
* Non-Function Requirements (Accessibility, reaachability)

#### Privacy

* Need to protect an individual’s data and identity

#### Ergonomics

Ergonomics is the study of the relationship between human workers and their working environment.

* + Poor ergonomics in the workplace reduces workers productivity and can cause health issues. The most common and most debilitating computer – usage health issue is repetitive strain injury (RSI). Other health issues involve general muscle strain and vision problems. Ergonomics for computer users include the design and placement of equipment together with procedures to prevent and minimise injury.

**Ergonomics of software and design:**

* Software design from the user’s perspective is very much about the User Interface. For many users, the user interface is the system. Software developers must create products that meet needs of the users. Products that provide fantastic functionality but have poor user interfaces will not be used. The better the user interface the more people will by the products.

**User interface:** the screen designs and connections between screens that allow the user to communicate with the software.

**User friendly:** software that meets the needs of the users. User-friendly software is intuitive consistent and easily learnt.

### 22.2.18

* A good GUI is important otherwise the program is seen as bad.
* A good user inter face needs to cater for:
  + Disability
  + Gender
  + Economic
  + Cultural

### 23.2.18

### Required Skills SDD

* Communication skills
* Ability to work in teams
* Creativity
* Design Skills
* Technical
* problem solving skills
* Attention to detail

#### Hardware

* Input
* Output
* Process
* storage
* control
* how a variety of input, output, storage devices and Cpu componments achieve their purpose
* the current trends and developments in computer hardare

### Software

* operating systems and utilities
* off-the-shelf applications packages and custom designed software
* generations of programing languages, namely:
  + Machine code: 1st gen
  + Assembly language 2nd gen
* Higher-level languages (imperative/procedural) language: 4th generation
* The need for translation
* Compilation
* Interpretation
* Functions of operating system
* Provide interface to hardware
* Provide interface to software application
* Provide to the user
* Control the concurrent running of software applications
* Manage system resources

### Relationships between hardware and software

* Processing the software instructions by hardware
* The fetch-execute cycle
* The inititiation and running of an application by the operating system
* Locate and load application
* Hand control to application
* Start fetch-decode-execute cycle for the application
* The existence of minimum hardware requirements to run some software

***The Five Fucntiosn performed by hardware devixes***

* Input
  + Puts data into the computer in a form ready for processing
* Output
  + Take the processed data and makes it available to people or other systems
* Processing
  + Works with the data to achieve the purpose of a system
* Control
  + Coordinates the activities of all components of the computer system
* Storage
  + Keeps data for later use by the computers

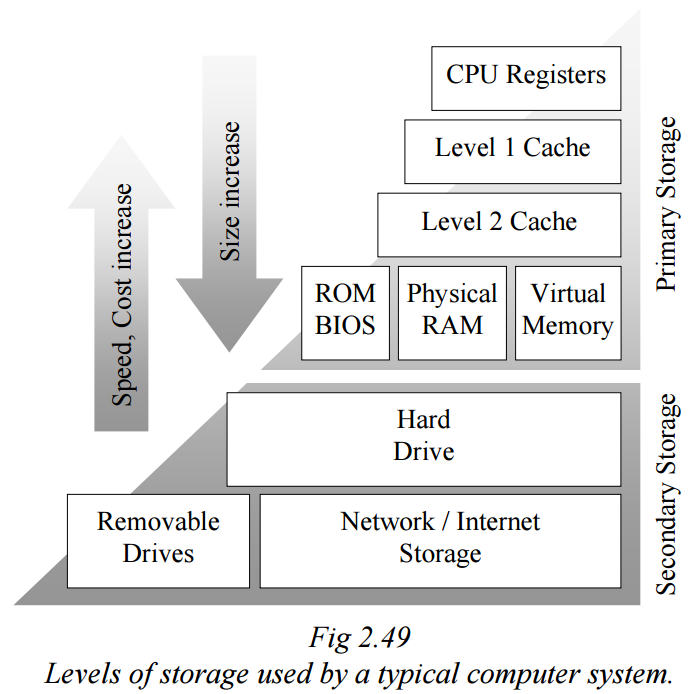
These five basic hardware functions combine to perform all the actions of all computer systems in brief, input is received from the outside of the system and processed into the output

This processing may involve respreading and/or writing to storage. Control is the function the coordinates the sequencing and timing of the other functions

**Input Devices**

|  |  |
| --- | --- |
| **Devices** | **Function & type of Data input / stored** |
| Mic |  |
| Hearing Aid |  |
| Cochlear Implant |  |
| Wii Bowling Ball |  |
| Joystick | Captures movement/ direction |
| Keyboard | Captures textual information – stores as asci digital code |
| Mouse | Captures user Movements- positional data |
| Trackpad | Captures user movements- positional data |
| Graphics tablet | Captures artists user movements- positional data |
| Trackpad | Captures user movements – positional data |
| Light gun camera | Captures reflected light- positional information |
| Trackball | Captures user movement- positional data |
| Tv Remote | Captures button presses |
| MIDI Keyboard | Captures key presses – translates to sound frequency |
| Scanner | Captures images |
| Barcode reader | Captures- image- light/dark – stores as number |
| Game controller | Captures- key presses and joystick data – positional data |
| Yoke (aeroplane Joystick ish) | HW input device for gaming input- positional and rotational info |
| Bionic eye | Captures imagery |
| Infrared touchscreen | Infrared to detect movement on screen |
| Web cam | Captures visual images sores as digital video |
| accelerometer | Captures movements/ acceleration |
| Fingerprint scanner | Captures images and identification/ pattern data |
| Xray Scanner | Captures intensity of X-rays transmitted |
| Magnetic card reader | Captures magnetic strip information |
| DSLR | Captures light/images on a ccd stores as bitmap images |
| Steering wheel- game controller | Captures rational information |
| Gyroscope | Captures directional information |
| OCR | Captures textual info and translates images into ASCII code |
| Sonar | Captures sound waves/ detects reflection of sound off solid object |
| Thermometer | Captures technology |
| Facial recognition camera | Facial contures |
| OMR optical mark reader | On/off data |
| Antenna | RF Capture |
| Thermostat | Capture temp to acute controllers |
| Geiger counter/gamma ray detector | Xrays and gamma rays |

### 6/03/18

* Storage is the function that reads, writes and retains data.
* The ability to store and retieve data is the major reason why computers are able to perform multiple tasks
* Other technologies invented by man are dedicated to performing one particular task; e.g. an oven cannot be easily modified into a refrigerator. Computers can in a matter of seconds change from being a word processor to managing businesses finance.

**Primary storage** works closely with the processor; it is generally very fast and apart from Read Only Memory (ROM), requires power to retain its contents.

* Primary storage is often referred to as main memory or just memory.
* It includes the registers within the CPU, cache, physical RAM, ROM and virtual memory.
* Apart from ROM primary memory is volatile, meaning it only temporarily holds data whilst the power is on

**Secondary storage** generally has a far greater storage capacity and does not require power to retain data. Both these storage types work together to carry out the storage functions.

* Secondary storage is permanent or non-volatile storage.
* The contents of secondary storage remains when the power is turned off.
* Most common secondary storage devices can be classified as either magnetic, optical or solid state.
  + Examples of magnetic devices include: hard disks and magnetic tapes.
  + CD-ROMs and DVDs use optical methods for reading and, where possible, for writing.
  + Network connections, including the Internet, provide further secondary storage on remote computers.
* In this section we consider magnetic storage and hard disk drives, optical storage and CD/DVD reading and writing, and finally flash drives which use solid state storage.

**Ram**

* Random access means that the system can read or write directly to any location
* Contains grid of memory cells is able to store single binary bit.

**CD’s** are a serial storage based on numbers and letters in a spiral on a cd

**Very important**

**Processing is the function that transforms the inputs into outputs.**

**Control is the function that directs the other components within the processor to perform their functions at the correct time and in the correct order.**

Both these functions are integrated within the central processing unit (CPU).

Both processing and control functions are performed by transistors, millions of transitors. Each transistor can be thought of as a switch, just like a light switch.  Either electrons flow through the switch or they do not. By connecting these switches in complex ways microprocessors are able to perform complex and varied tasks.

**The CPU contains**

* the ALU (Arithmetic Logic Unit)
* the control unit and
* registers.

The **registers**are memory areas used to store instructions, addresses and data currently being used by the CPU.

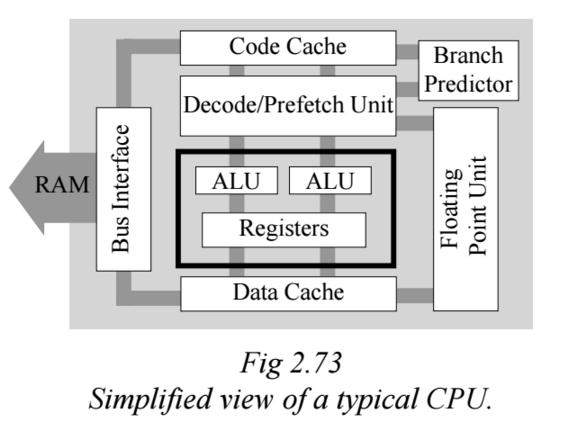
The**control unit** makes sense of each instruction it receives and directs the ALU to perform the appropriate process.

The data and the result from the ALU’s processing are stored in the CPU’s internal **registers**.

The control unit, ALU and the registers are hardwired into the CPU, each design of CPU having its own unique instruction set.

Modern CPU’s also contain one or more **cache**areas to speed up the interface between the CPU and RAM.

Most also have an integrated **floating point unit (FPU)** to perform real number computations.

[[](https://drive.google.com/file/d/0BwzvXeA1LcQuemZ1Rm5PX1BwZ28/view?usp=drive_web)](https://drive.google.com/file/d/0BwzvXeA1LcQuemZ1Rm5PX1BwZ28/view?usp=drive_web)

Let us consider the function of each component shown on this diagram. Remember each of these components are hardware components. They are hardwired to perform their functions.

* **Bus interface** – provides a communication channel between the CPU and primary storage (RAM). Both instructions and data are received and sent via this interface. Information moves faster within the CPU than into and out of the CPU, the bus interface must sort out this inconsistency.
* **Code cache** – a storage area for instructions that are likely to be needed in the near future. Most programs involve repeating the same instructions, these instructions are retained in the code cache where they can be accessed quickly.
* **Branch predictor** – attempts to guess the most likely next instruction and determine its reliance on previous instructions. This is particularly useful when there is a jump within the code to a set of new instructions. The Branch Predictor assists the Decode/Prefetch Unit to get instructions ready in advance.
* **Decode/Prefetch Unit** – performs most of the control functions and handles the execution of some instructions. This unit makes sense of instructions and directs the other components to perform their functions. A number of instructions are being fetched and decoded at the same time in a modern CPU.
* **ALU**– modern CPUs have more than one **arithmetic logic unit**, so more than one instruction can be executing at the same time. The ALUs perform the bulk of the actual processing functions. Essentially the ALU can add binary numbers, shift them left or right, and compare them. Combinations of these tasks allow ALUs to subtract, multiply, divide and perform logical comparisons such as greater than, less than, and equal to.
* **Registers**– temporary storage areas used by the ALUs during execution. Registers hold the operands required to execute instructions and also the results obtained after execution.
* **Data Cache**– very fast memory area for storage of data that may soon be needed or is needed repeatedly.
* **Floating Point Unit (FPU)** – performs all non-integer calculations. The FPU is a processing unit dedicated to performing computations involving floating point numbers.

### Software

|  |
| --- |
| Software is a set of instructions used to direct the operation of hardware causing it to solve some problem.  Software provides the communication link between hardware and users.  There are several different types of software that work together to allow users to interact with the computer hardware and get the computer to actually do what is required |

**Operating System and Utilities**

|  |
| --- |
| The operating system is the first software we see once the computer has booted and the last software seen when we shutdown.  It organises and controls the hardware and other software used by the system.  Operating systems provide a stable and consistent way for applications to use hardware without them having to know the precise nature of the hardware.  **Types of Operating Systems**   * **Real Time OS** - controls machinery and scientific instructments * **Single User, Single Task OS** - e.g. Symbian * **Single User, multi-tasking OS** - eg. Windows, Apple OSX etc * **Multi User OS** - allows access to system resources by many users, protecting each user   from other users - each user thinks they have the machine to themselves..  E.g. Mainframes - IBM Multiple Virtual Storage (MVS)  **Functions of Operating Systems**   * **Processor management** - allocates processes to processors, taking into account delays from   input/output devices   * Memory and storage management - each process is allocated sufficient memory to execute * Device Management - communicating with Drivers - which translate messages from computer   to that which can be understood by device   * Application Interface - allows application software to communicate with OS functions. * User Interface - provides a mechanism for User to interact with the Computer OS functions   - MS DOS prompt, GUI interfaces like Windows.  These sit on top of the main operating system  **Utilities**   * File Compression * Defragmentation * Virus Checking * Embedded Licence installation count * Batch Job Scheduling - e.g. Auto Virus Checking & Defragmentation * Emulation - emulate other OS - e.g. GUI shell on Linux or MS Windows on MAC |