2 Computer Architecture

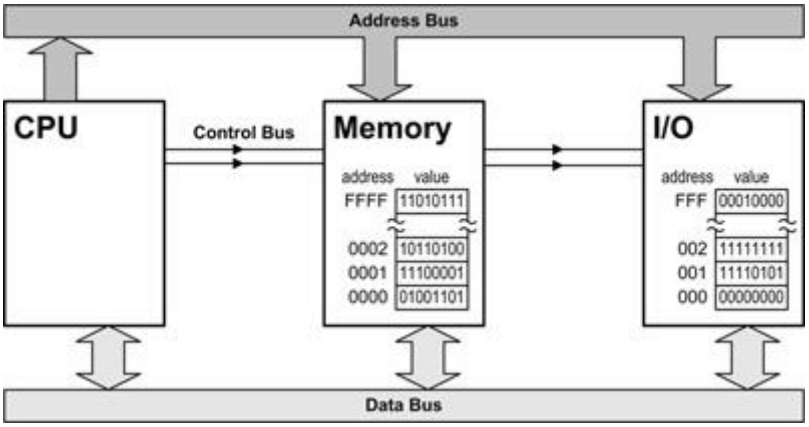
2.1 Overview of the Organization of a Computer System

2.1.1 Introduction

In computer science and engineering, computer architecture is a set of disciplines that describes a computer system by specifying its parts and their relations.

The first documented computer architecture is found in the correspondence between Charles Babbage and Ada Lovelace in mid nineteenth century, describing the analytical engine. Another early example is John Von Neumann's 1945 paper, “First Draft of a Report on the EDVAC”, which described an organization of logical elements.

2.1.2 The Main Components of a Computer



The main components of a computer

The diagram shown above illustrates the main components of a computer. These are:

- CPU
- RAM
- I/O subsystem
- buses

Other important components are the following:

- system clock
- ROM
- secondary storage
- caches

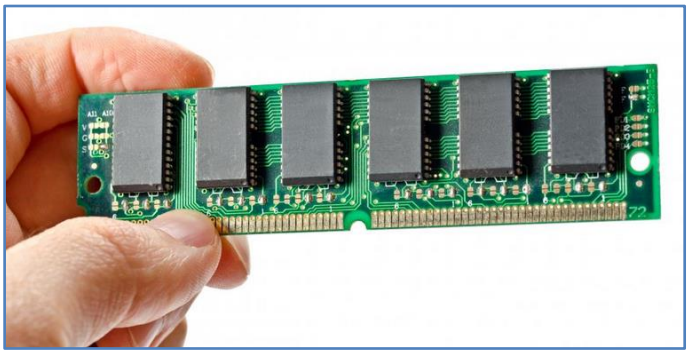
A brief description of the mentioned components is given hereunder.

The CPU

The central processing unit (CPU) is the unit, which executes the instructions. If the computer is given the instruction ‘add 2 and 7’ it is the CPU that has the electronics to perform the addition. The computer will pass on this instruction (from input or from a program) to the CPU and the result will be passed to a program or as output but the actual calculation is done by the CPU.

RAM

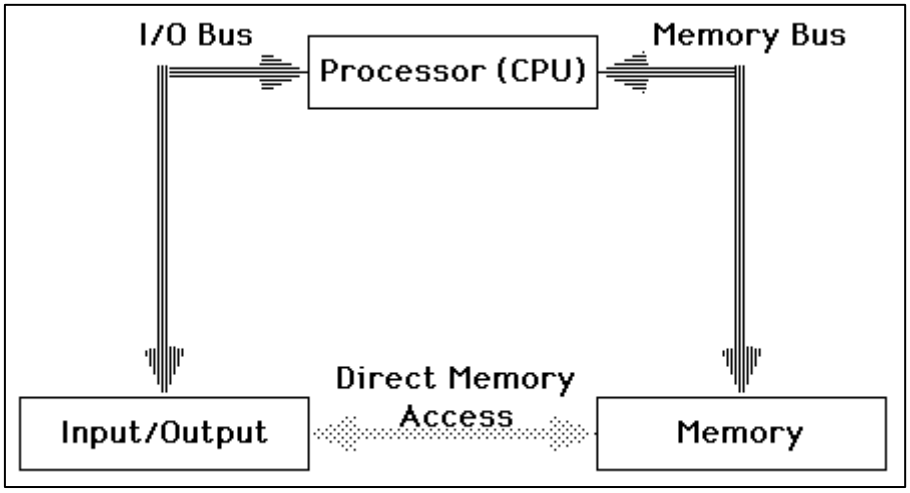
RAM stands for Random Access Memory. It is also known as main memory (or primary memory). It holds the programs currently being executed by the computer. It holds the applications that are 'open'. Once an application is closed the space occupied by it is cleared for use by other applications. This kind of memory is volatile.



RAM chips

I/O Subsystem

The I/O subsystem interfaces the computer with the user.



The I/O subsystem interfaces the computer with the user.

Buses

A bus is a subsystem that is used to connect computer components and transfer data between them. For example, an internal bus connects computer internals to the motherboard.

A bus may be parallel or serial. Parallel buses transmit data across multiple wires. Serial buses transmit data in bit-serial format.

Computer buses are frequently divided into three categories:

- Address bus (to transfer addresses)
- Data bus (to transfer data)
- Control bus (to transfer messages)

The System Clock

The clock is a device that generates periodic, accurately spaced signals used for several purposes such as regulation of the operations of a processor. The clock circuit uses the fixed vibrations generated from a quartz crystal to deliver a steady stream of pulses to the processor. The system clock controls the speed of all the operations within a computer.

The clock speed is the internal speed of a computer. It is expressed in megahertz (MHz) or gigahertz (GHz). 33 MHz means 33 million cycles per second. A computer processor's speed is increased if the clock speed is increased.

The ROM

Read-only memory (ROM) is computer memory which is not volatile. Most personal computers contain a small amount of ROM that stores critical programs such as the program that boots the computer. ROMs are also used in peripheral devices such as laser printers, whose fonts are often stored in ROMs.

A variation of a ROM is a PROM (programmable read-only memory). PROMs are manufactured as blank chips on which data can be written with a special device called a PROM-programmer.

ROM chip

Secondary storage

Secondary storage (also called mass storage) holds data and programs to be accessed later on by the computer. When the computer requires a file or program from secondary storage, this is loaded (copied) in RAM and it is here that the CPU can access it. The following is a list of secondary storage devices:

- Hard disks: Very fast and possessing very large memories. Some hard disk systems are portable but most are not.
- Optical disks: Unlike hard disks, which use electromagnetism to encode data, optical disk systems use a laser to read and write data. Optical disks have very large storage capacity, but they are not as fast as hard disks.
- Tapes: Relatively inexpensive and can have very large storage capacities, but they do not permit random access of data.
- Pen drives: They are fast as they do not have moving parts. They are made of only electronic parts.

Mass storage is measured in kilobytes (1,024 bytes), megabytes (1,024 kilobytes), gigabytes (1,024 megabytes) and terabytes (1,024 gigabytes). Mass storage is sometimes called auxiliary storage.

Caches

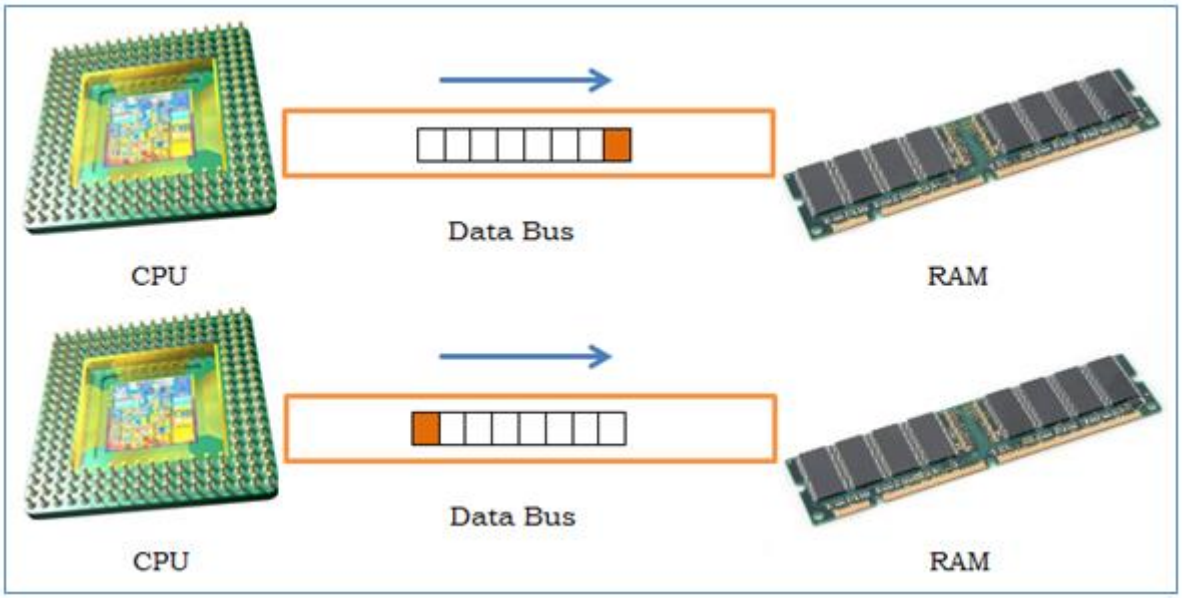
Cache is fast memory that is used to speed up the computer since their retrieval time is very short.

2.2 The System Bus

A bus is a subsystem that is used to connect computer components and transfer data between them. For example, an internal bus connects computer internals to the motherboard.

2.2.1 Protocol

The term 'protocol' indicates the rules by means of which data is carried from one place to another. In the diagram shown below one byte is being carried from the CPU to the RAM. The system at the top puts the parity bit on the 'right' while the system at the bottom puts it on the 'left'. This shows that they are using a different protocol.



Different protocols

2.2.2 Serial and Parallel Buses

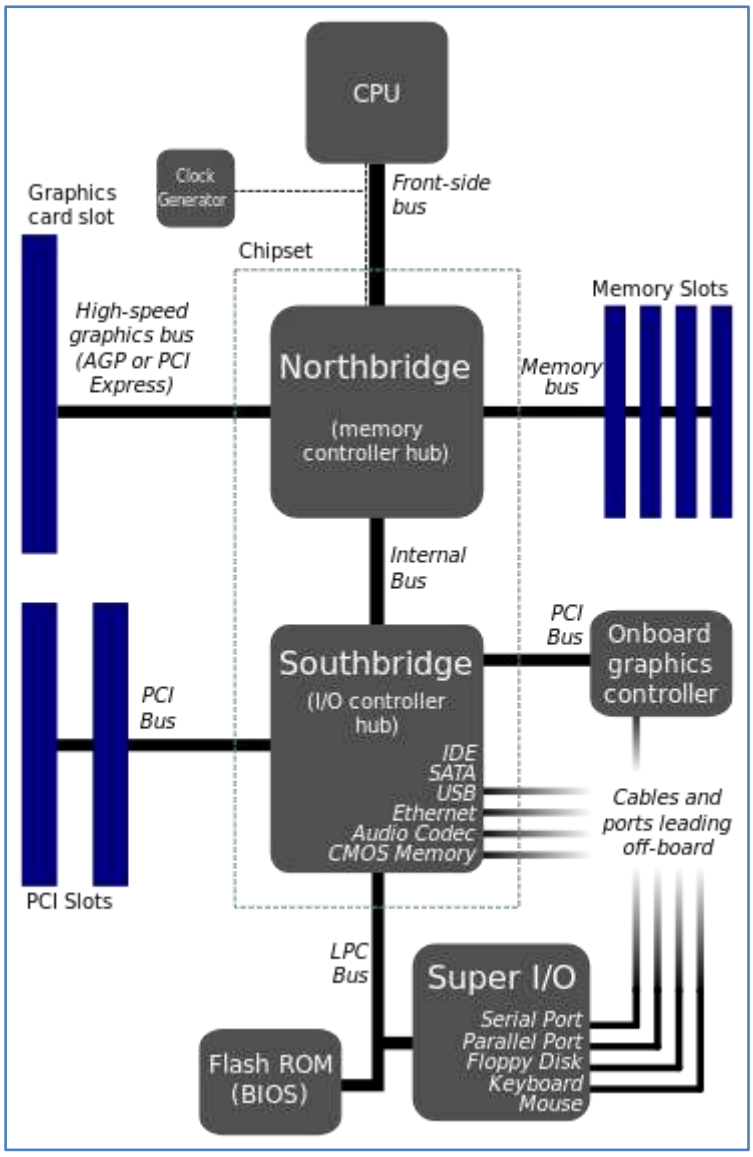
Parallel bus standards include advanced technology attachment (**ATA**) or small computer system interface (**SCSI**) for printer or hard drive devices. Serial bus standards include universal serial bus (**USB**), **FireWire** or **serial ATA**.

2.2.3 The Buses Subsystem

In a computer, there are two major types of buses: the **system bus** and **peripheral bus**. The system bus, also known as the **frontside bus** or **local bus**, is the internal path from the CPU to memory and is split into address bus, data bus and control bus.

System buses transfer data in parallel. In a 32-bit bus, data are sent over 32 wires simultaneously. A 64-bit bus uses 64 wires.

The peripheral bus is the pathway to the peripheral devices such as a disk or printer. PCI and PCI Express are widely used peripheral buses. Devices connect to these parallel buses with cables to controller cards that plug into slots on the motherboard. Another common bus is USB, and devices are cabled to ports on the computer. USB is a serial bus, in which data travels over one wire.



A particular Architecture of PC Buses

2.2.4 Buses Size Consideration

The width of the address bus determines the amount of memory a system can address. For example, a system with a 32-bit address bus can address 2^{32} (4,294,967,296) memory locations. If each memory address holds one byte, the addressable memory space is 4 GB.

If n = width of address bus

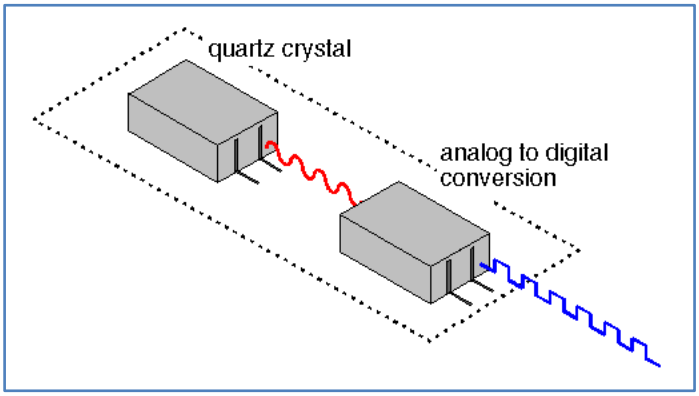
Number of addressable memory locations =

$$2^n$$

2.2.5 System Clock

The **System Clock** (or simply **Clock**) is an internal timing device. Using a quartz crystal, the CPU clock breathes life into the microprocessor by feeding it a constant flow of pulses. For example, a 200 MHz CPU receives 200 million pulses per second from the clock. A 2 GHz CPU gets two billion pulses per second. Similarly, in a communications device, a clock is used to set the transmission speed and may also be used to synchronize the pulses between sender and receiver.

A real-time clock, also called the system clock, keeps track of the time of day and makes this data available to the software.



A System Clock

2.2.6 Decoders

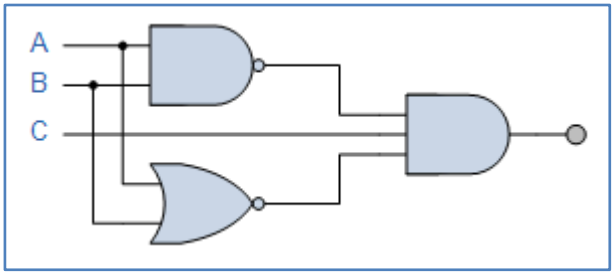
A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

A_2	A_1	A_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Truth Table of two Decoders

A **decoder** circuit is used to recognize the various combinations of an input word and provide an output for each combination. If an input word contains N "bits" then the decoder can have 2^N outputs.

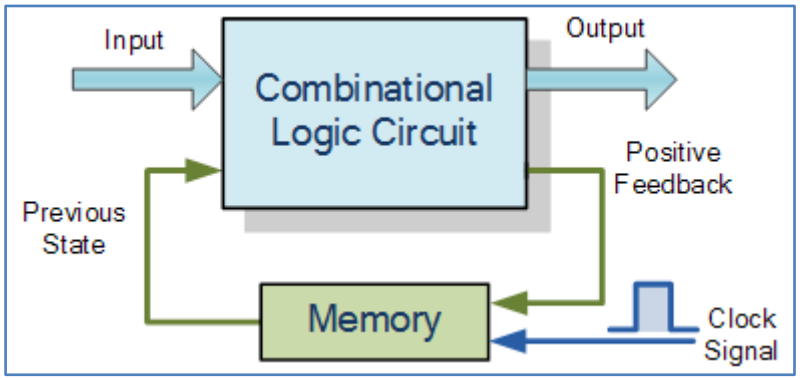
A decoder is the opposite as an **encoder**. An encoder changes data from a user format to computer format. For example the keyboard is an encoder. It changes data from characters to character (ASCII) code. A decoder changes data from machine-form to user-form. For example the data sent to a printer.



A Combinatorial Circuit

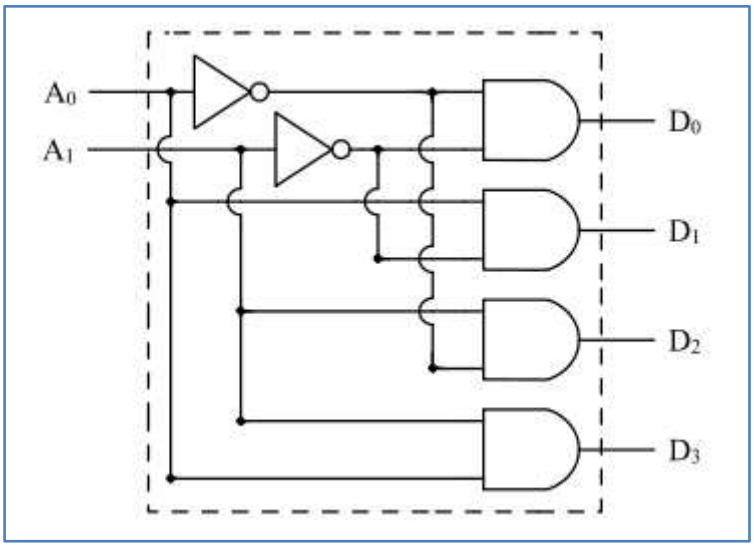
The encoder and decoder are both combinational

circuits. In digital circuit theory, **combinational logic** is a type of digital logic which is implemented by Boolean circuits, where the output is a pure function of the present input only. This is in contrast to **sequential logic**, in which the output depends not only on the present input but also on the history of the input.

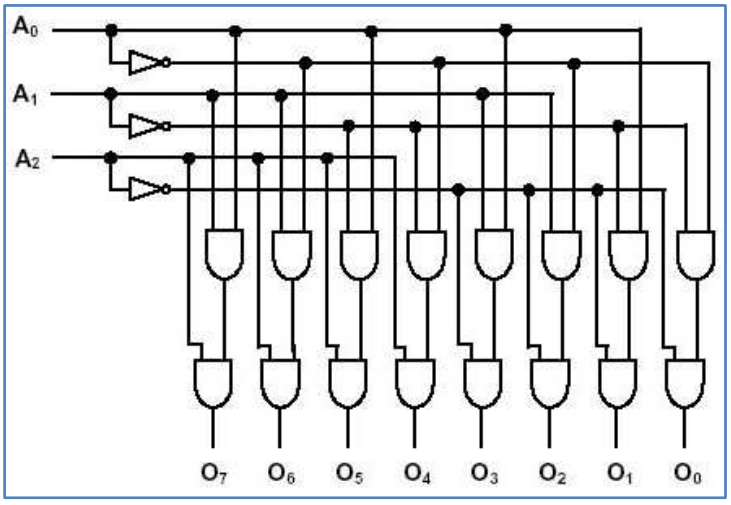


A Sequential Circuit

In other words, sequential logic has memory while combinational logic does not. The circuit diagrams of the two decoder truth tables seen above are given below.



Logic Circuit of a 2-to-4 Decoder

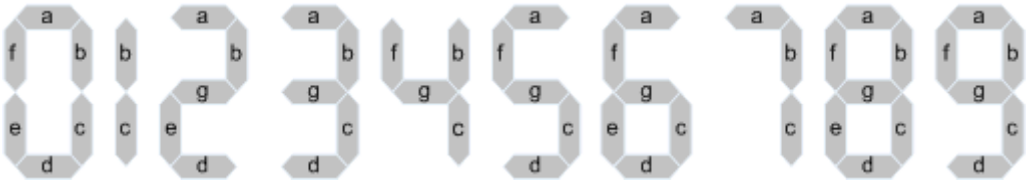


Logic Circuit of a 3-to-8 Decoder

Decoding is necessary in applications such as 7 segment display and memory address decoding.

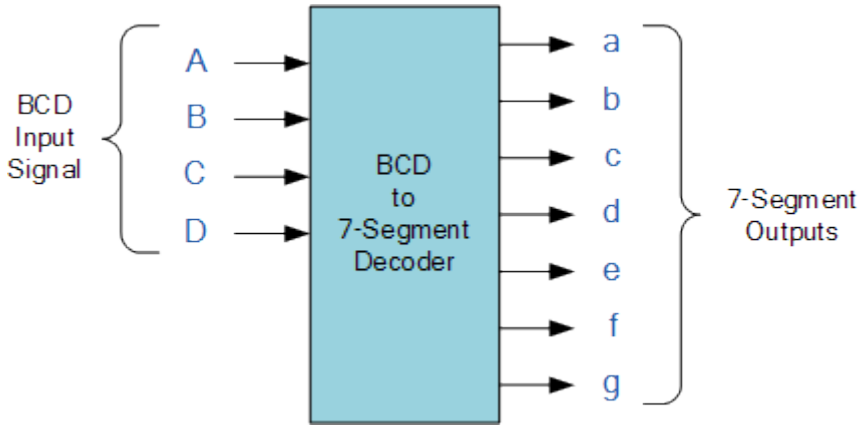
BCD to 7-Segment Display Decoder

The following diagrams show how a decoder can be used to translate a BCD representation to a 7-segment display.



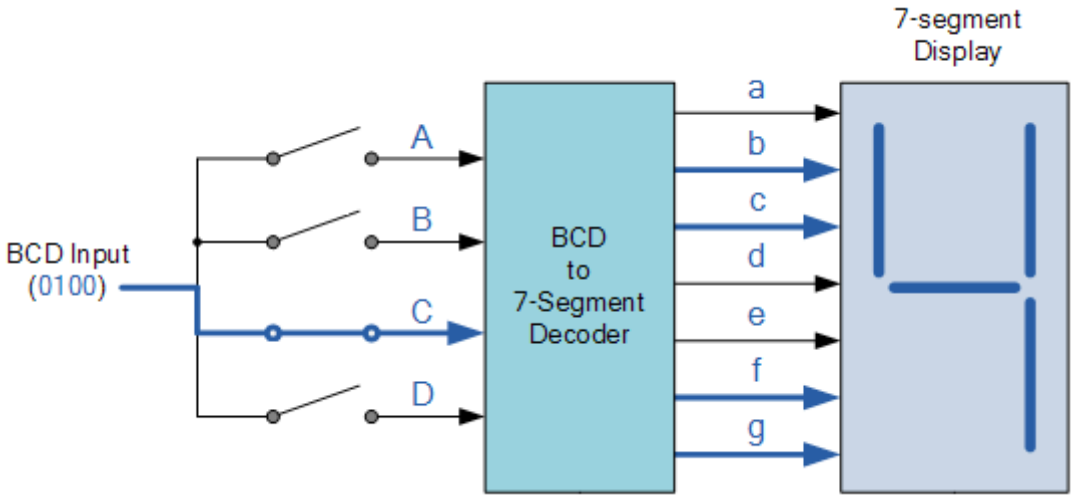
7-Segment Display Elements for all Numbers.

Note in the diagram below that this decoder has 7 outputs and not 16 (2^4).



BCD to 7-Segment Decoder

The following diagram shows an example of how the decoder will translate the number 4 from BCD to the 7-segment display. Note that in this case more than one input can have an output equal to one.

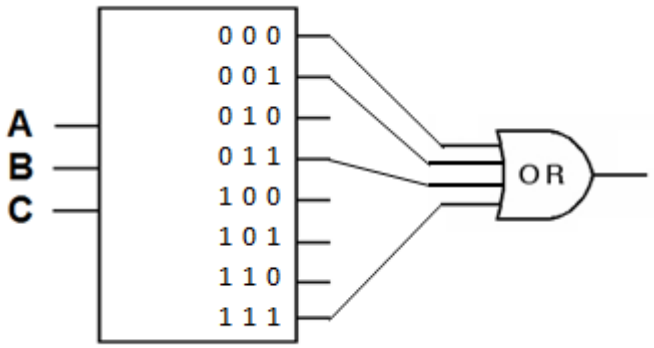


Decoder with input 0100 and output the display of 4

Implementing a Truth Table by means of a Decoder and an OR Gate

The diagram below shows how a truth table can be implemented using a decoder.

A	B	C	Out
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

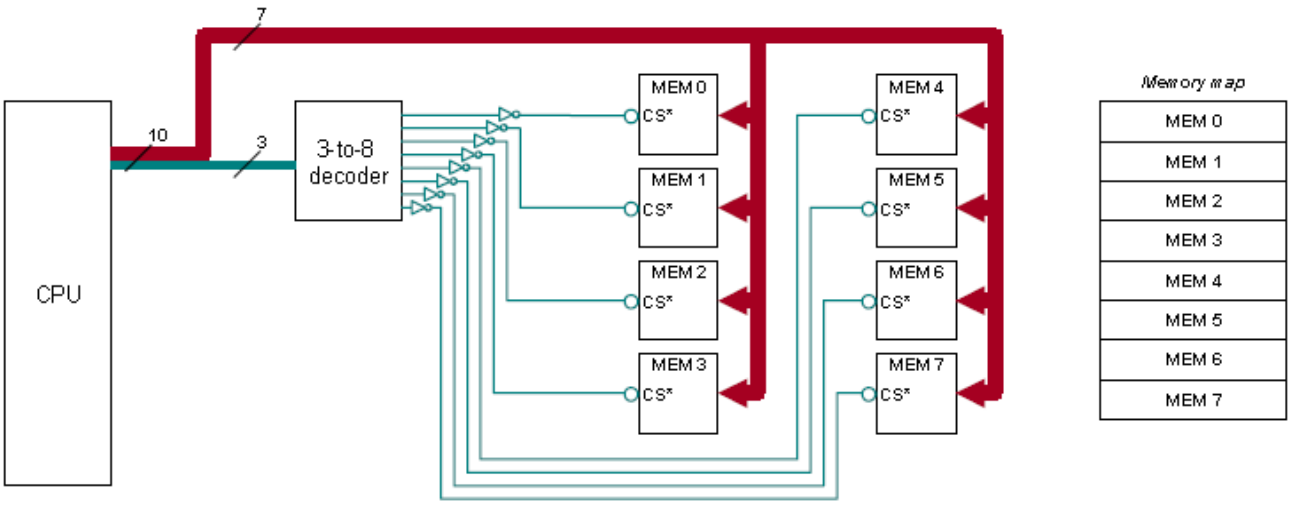


A Decoder can be used to implement a Truth Table

Memory Address Decoder

Most kinds of random-access memory use an n-to-2^n decoder to convert the selected address on the address bus to one of the row address select lines.

Let us consider this simple example. Let's assume a very simple microprocessor with 10 address lines (1KB memory). Let us also assume that we have 8 memory chips of 128B each. So we need 3 address lines to select each one of the 8 chips and each chip will need 7 address lines to address its internal memory cells. The solution is shown in the following diagram.



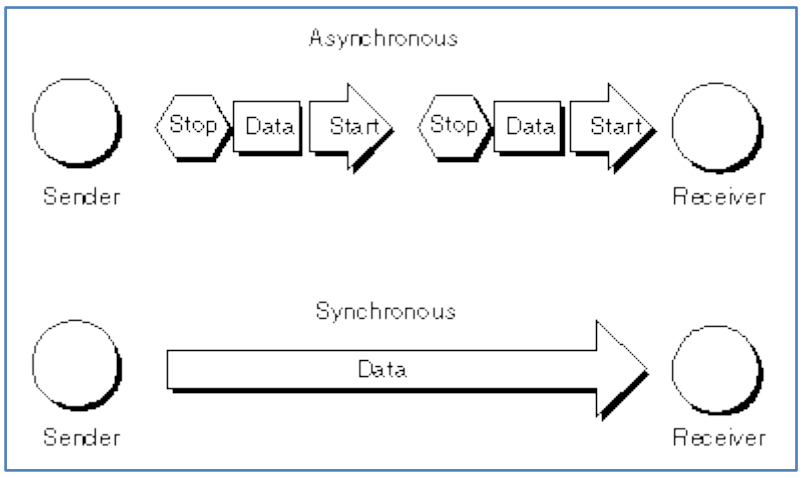
A Decoder can be used select a Memory chip

In the above diagram the decoder inputs a 1 to only one of the CS* (chip select) input on each chip. In this example the decoder can be thought of an interface between the CPU and memory.

2.2.7 Synchronous and Asynchronous Transfer

The term **synchronous** is used to describe a continuous and consistent timed transfer of data blocks. The opposite of synchronous is **asynchronous**. Most communication between computers and devices is asynchronous - it can occur at any time and at irregular

intervals. Communication within a computer, however, is usually synchronous and is governed by the microprocessor clock. Signals along the bus, for example, can occur only at specific points in the clock cycle.

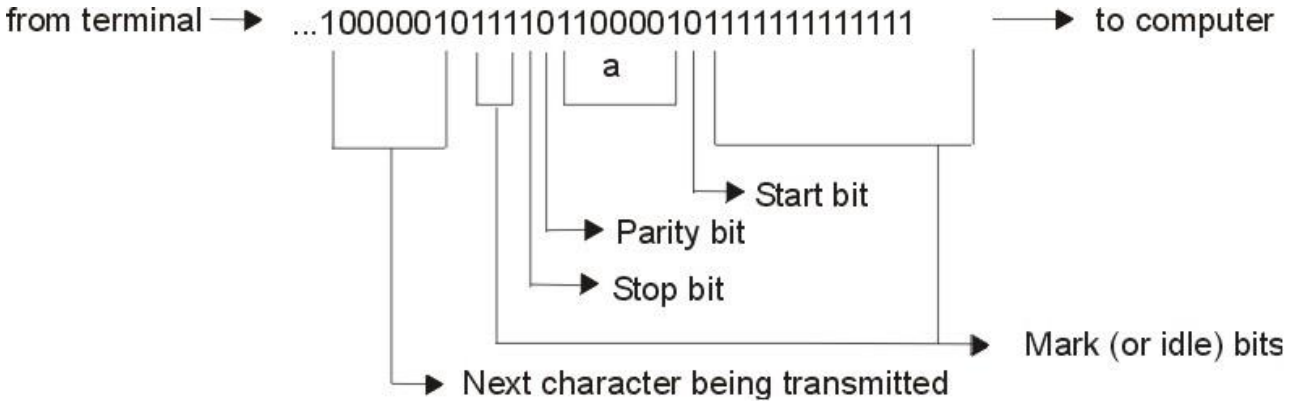


Synchronous and Asynchronous transmission

Asynchronous means 'not synchronized' i.e. not occurring at predetermined or regular intervals. The term asynchronous is usually used to describe communications in which data can be transmitted intermittently rather than in a steady stream.

The difficulty with asynchronous communications is that the receiver must have a way to distinguish between valid data and noise. In computer communications, this is usually accomplished through a special start bit and stop bit at the beginning and end of each piece of data. For this reason, asynchronous communication is sometimes called 'start-stop transmission'.

Most communications between computers and devices are asynchronous.



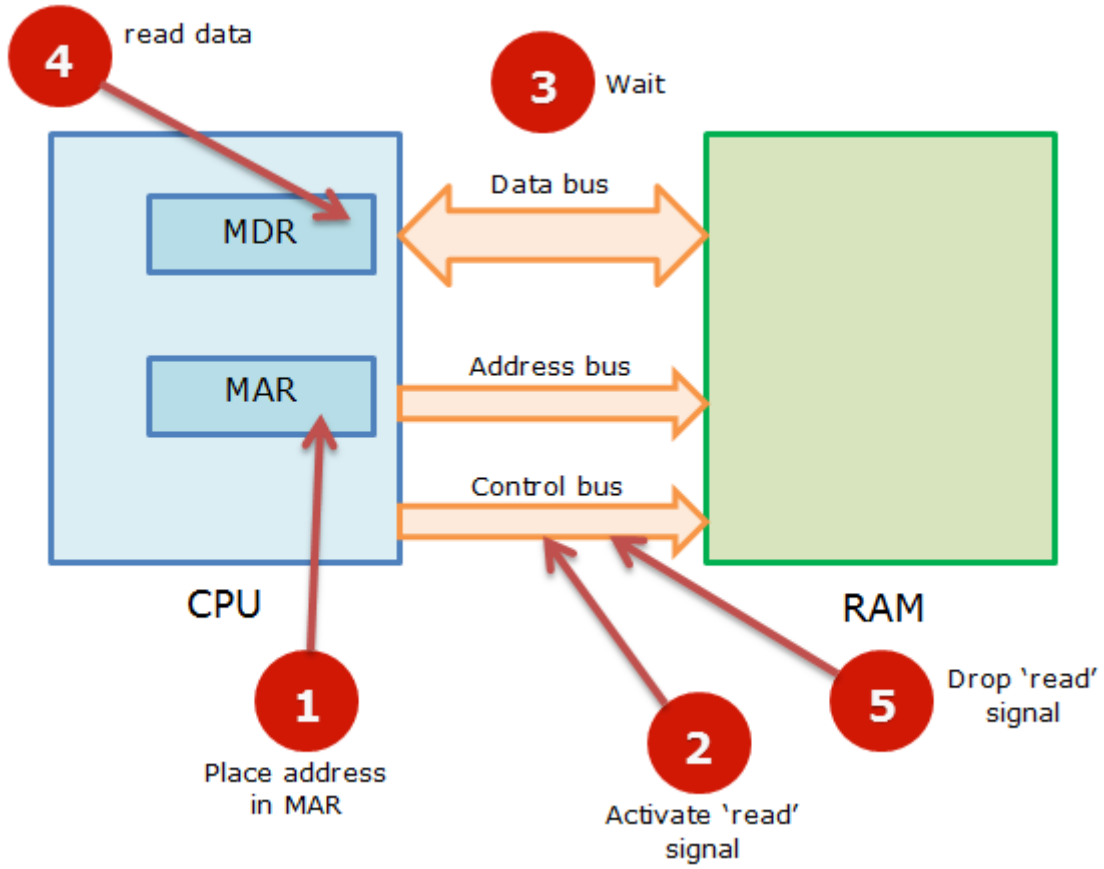
Asynchronous Transmission of Blocks of Data

2.2.8 Read and Write Memory Cycles

These are the steps in a typical Read Cycle:

1. Place the address of the location to be read on the address bus via MAR.
2. Activate the memory read control signal on the control bus.
3. Wait for the memory to retrieve the data from the addressed memory location.
4. Read the data from the data bus into MDR.
5. Drop the memory read control signal to terminate the read cycle.

A simple Pentium memory read cycle takes 3 clock cycles. Steps 1-2 and then 4-5 are done in one clock cycle each. For slower memories, wait cycles will have to be inserted.



Read Cycle

These are the steps in a typical Write Cycle:

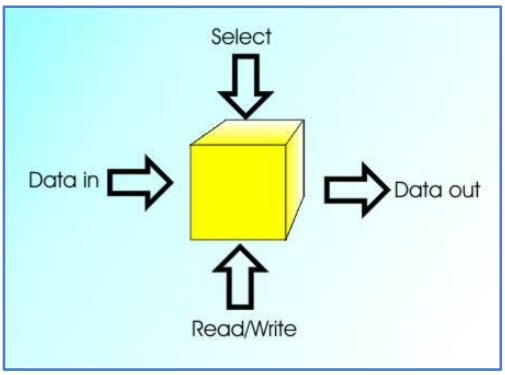
1. Place the address of the location to be written on the address bus via MAR.
2. Place the data to be written on the data bus via MDR.
3. Activate the memory write control signal on the control bus.
4. Wait for the memory to store the data at the addressed location.
5. Drop the memory write control signal to terminate the write cycle.
6. A simple Pentium memory write cycle takes 3 clocks:

Steps 1-2 and 4-5 are done in one clock cycle each. For slower memories, wait cycles will have to be inserted.

2.3 Memory

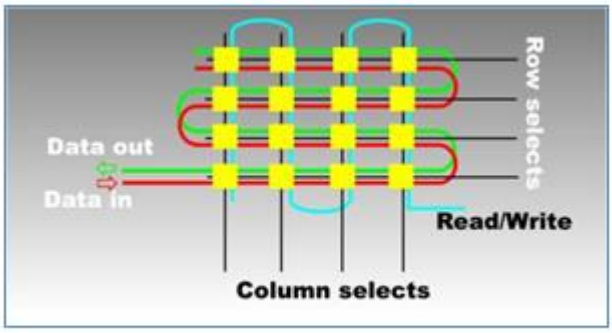
2.3.1 Memory Organisation

Each memory element or cell can store one bit and it has a data input line, a data output line, a read/write line and a select line. The select line activates the cell and read/write line tells it either to output its contents or store what is at its input.

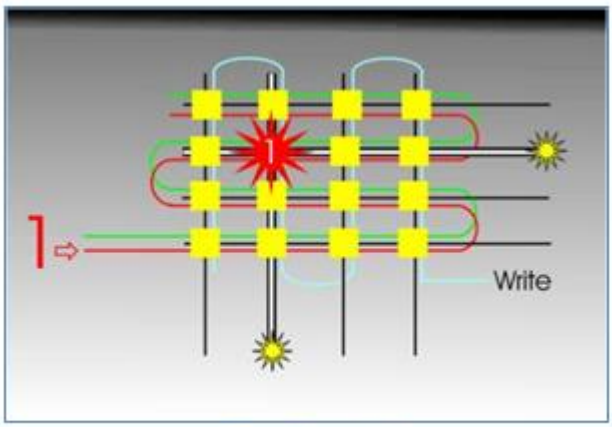


A Memory Element

The cells are organised into a grid with horizontal and vertical selection wires – row and column selects. A cell is



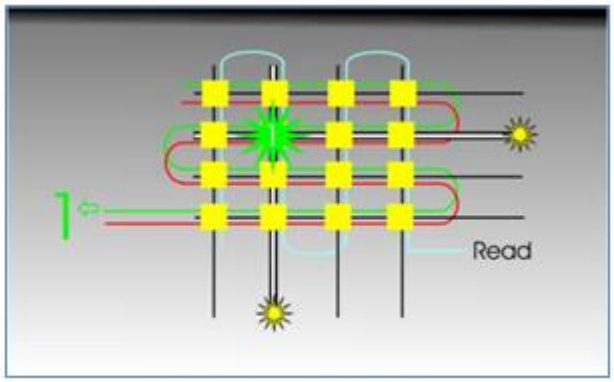
A Memory Array



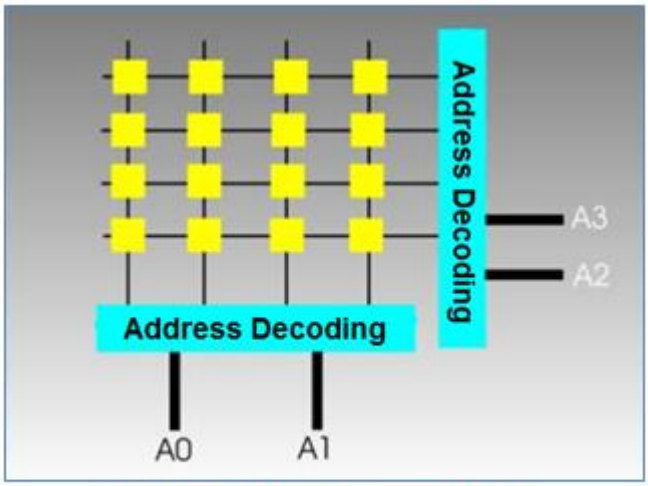
Storing a One

selected when both its row and its column select are high.
Now suppose you want to store a 1 in the array at a particular location. The data is placed on the data input line and the read/write line is set low to indicate that you want to write to the array. Finally the appropriate row and column select is set to high to select the cell in question and it, and only it, stores the data on the data line.

To read the data back from the array you do the same thing only with the read/write line set high to indicate that you want to retrieve the bit in question. Only the selected cell outputs anything to the common data out line.



Reading the Array



Address Lines converted to Row/Column Selects

Notice that at the moment we are looking at a single memory chip and this arrangement can only store a single bit. If you want to store a whole byte you need eight such chips, one for each bit, and eight data input and eight data output lines.

The eight data lines are grouped together into a data input bus and a data output "bus" – a bus is just a group of wires. Early computers really did have separate buses for input and output but today's machines have a single unified data bus that can be used for either input or output.

If you want more storage than a bank of eight chips can provide then you have to add another bank of eight chips and some additional address decoding logic to select the

correct bank. The address lines that come from the processor are generally referred to as an address bus and now we have the fundamental architecture of a simple but modern computer.

The number of address lines indicates the number of different memory locations that can be accessed. The relationship between these two values is shown in the following table.

Address Lines	Number of Locations	
1	2^1	= 2
2	2^2	= 4
3	2^3	= 8
4	2^4	= 16
5	2^5	= 32
6	2^6	= 64
7	2^7	= 128
8	2^8	= 256
9	2^9	= 512
10	2^{10}	= 1,024
11	2^{11}	= 2,048
12	2^{12}	= 4,096
13	2^{13}	= 8,192
14	2^{14}	= 16,384
15	2^{15}	= 32,768
16	2^{16}	= 65,536
17	2^{17}	= 131,072
18	2^{18}	= 262,144
19	2^{19}	= 524,288
20	2^{20}	= 1,048,576

2.3.2 Dynamic and Static RAM

There are two different types of RAM: DRAM (Dynamic Random Access Memory) and SRAM (Static Random Access Memory). The two types differ in the technology they use to hold data, with DRAM being the more common type. In terms of speed, SRAM is faster. DRAM needs to be refreshed thousands of times per second while SRAM does not need to be refreshed, which is what makes it faster than DRAM. DRAM supports access times of about 60 nanoseconds, SRAM can give access times as low as 10 nanoseconds. Despite SRAM being faster, it's not as commonly used as DRAM because it's so much more expensive. Both types of RAM are volatile.

Advantages of SRAM	Advantages of DRAM
• Faster than DRAM • More energy-efficient than DRAM	• Cheaper than SRAM • Smaller in size (for the same amount of memory) than DRAM

A dynamic RAM chip holds millions of memory cells, each comprised of a transistor and a capacitor. Each of these cells is capable of holding 1 bit of information, which is read by the computer as either 1 or 0. To determine the reading of a bit, the transistor checks for

a charge in the capacitor. If a charge is present, then the reading is 1; if not, the reading is 0.

The problem with dynamic RAM chips is that the capacitor leaks energy very quickly and can hold a charge for only a fraction of a second. A refresh circuit is needed to maintain the charge in a capacitor and retain the information.

Static RAM chips, on the other hand, use a different technology. Memory cells flip-flop between 0 and 1 without the use of capacitors, meaning that no refreshing process is needed and access takes place only when the information is required. Without the need to constantly access all information, SRAM is much faster than DRAM. Generally speaking, these chips are far more energy efficient, but this is only because of their limited need to access memory, and the rate of consumption rises with heavy use.

The biggest drawback to SRAM is space. Each transistor on a dynamic RAM chip can store one bit of information, but four to six transistors are required to store a bit using SRAM. This means that a dynamic RAM chip will hold at least four times as much memory as a static RAM chip of the same size, making SRAM much more expensive. DRAM is more commonly used for personal computer memory, and SRAM chips are preferred when energy efficiency is a concern, such as in cars, household appliances, and handheld electronic devices.

There are more than one type of SRAM and DRAM. A few examples are shown hereunder:

Async SRAM is an older type of SRAM. It is asynchronous, meaning that it works independently of the system clock.

Sync SRAM is synchronized with the system clock, and faster than Async SRAM.

Pipeline Burst SRAM is the most common type of SRAM. It is able to operate at bus speeds higher than 66MHz.

FPM DRAM (Fast Page Mode DRAM) is slightly faster than regular DRAM. This used to be the main type of memory used in PCs but was eventually replaced by EDO RAM, due to its slow speed. FPM DRAM, is now considered to be obsolete.

EDO DRAM (Extended Data Out DRAM) provided a better performance increase over FPM DRAM. EDO RAM cannot operate on a bus speed faster than 66MHz. With a need for speed BEDO DRAM was introduced.

BEDO DRAM (Burst EDO DRAM) is a type of EDO DRAM that can process four memory addresses in one burst. BEDO DRAM can only stay synchronized with the CPU clock for short periods (bursts). It is faster than its predecessor, EDO DRAM.

SDRAM (Synchronous DRAM) is a type of DRAM that can run at much higher clock speeds than conventional memory. SDRAM actually synchronizes itself with the CPU's bus. SDRAM is the new memory standard for modern PCs.

Applications of DRAM	Applications of SRAM
• Computer RAM • TVs • Video cameras • GPSs	• Cache (support for DRAM) in computers • Digital cameras • Cell phones

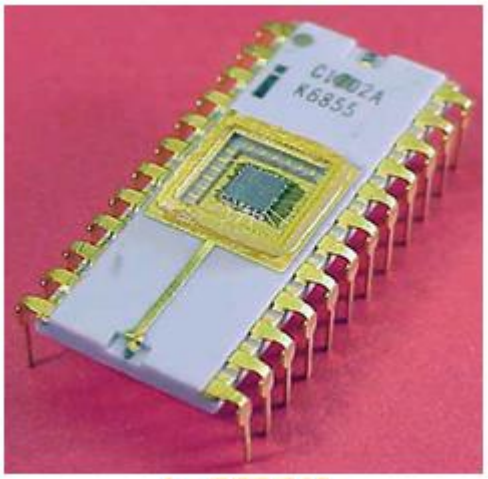
2.3.3 ROM

ROM is an acronym for 'read-only memory'. Unlike main memory (RAM), ROM retains its contents even when the computer is turned off. ROM is referred to as being non-volatile, whereas RAM is volatile.

ROM is also known as firmware. It is an integrated circuit programmed with specific data and programs when it is manufactured.

There are five basic ROM types:

- ROM
- PROM
- EPROM
- EEPROM
- Flash memory



An EPROM

These memories have two things in common:

- They are non-volatile.
- Data stored in these chips is either unchangeable or requires a special operation to change.

ROM chips use very little power, are extremely reliable and, in the case of most small electronic devices, contain all the necessary programming to control the device.

PROM

Creating ROM chips totally from scratch is time-consuming and very expensive in small quantities. For this reason, mainly, developers created a type of ROM known as programmable read-only memory (PROM). Blank PROM chips can be bought inexpensively and coded by anyone with a special tool.

PROM chips have a grid of columns and rows just as ordinary ROMs do. The difference is that every intersection of a column and row in a PROM chip has a fuse connecting them. A charge sent through a column will pass through the fuse in a cell to a grounded row indicating a value of 1. Since all the cells have a fuse, the initial (blank) state of a PROM chip is all 1s. To change the value of a cell to 0, you use a programmer to send a specific amount of current to the cell. The higher voltage breaks the connection between the column and row by burning out the fuse. This process is known as burning the PROM. PROMs can only be programmed once. They are more fragile than ROMs. A jolt of static electricity can easily cause fuses in the PROM to burn out, changing essential bits from 1 to 0. But blank PROMs are inexpensive and are great for prototyping the data for a ROM before committing to the costly ROM fabrication process.

EPROM

Working with ROMs and PROMs can be a wasteful business. Even though they are inexpensive per chip, the cost can add up over time. Erasable programmable read-only memory (EPROM) addresses this issue. EPROM chips can be rewritten many times. Erasing an EPROM requires a special tool that emits a certain frequency of ultraviolet.

EPROMs are configured using an EPROM programmer that provides voltage at specified levels depending on the type of EPROM used.

To rewrite an EPROM, you must erase it first. Each EPROM chip has a quartz window on top of it. The EPROM must be very close to the eraser's light source, within an inch or two, to work properly.

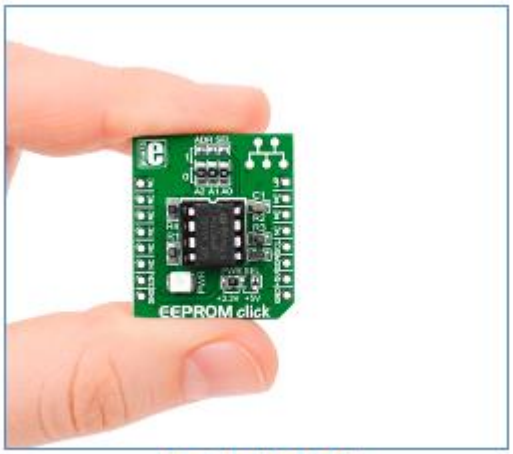
An EPROM eraser is not selective; it will erase the entire EPROM. The EPROM must be removed from the device it is in and placed under the UV light of the EPROM eraser for several minutes. An EPROM that is left under too long can become over-erased. This ruins the EPROM in such a way that it cannot hold any memory any more.

EEPROMs and Flash Memory

Though EPROMs are a big step up from PROMs in terms of reusability, they still require dedicated equipment and a labour-intensive process to remove and reinstall them each time a change is necessary. Also, changes cannot be made incrementally to an EPROM; the whole chip must be erased. Electrically erasable programmable read-only memory (EEPROM) chips remove the biggest drawbacks of EPROMs.

In EEPROMs:

- The chip does not have to be removed to be rewritten.
- The entire chip does not have to be completely erased to change a specific portion of it.
- Changing the contents does not require additional dedicated equipment.



An EEPROM

Instead of using UV light this technology uses electric fields to erase information. EEPROMs are changed 1 byte at a time, which makes them versatile but slow. In fact, EEPROM chips are too slow to use in many products that make quick changes to the data stored on the chip.

Manufacturers responded to this limitation with Flash memory, a type of EEPROM that uses in-circuit wiring to erase by applying an electrical field to the entire chip or to predetermined sections of the chip called blocks. Flash memory works much faster than traditional EEPROMs because it writes data in chunks, usually 512 bytes in size, instead of 1 byte at a time.

Applications of ROMs, PROM etc.

Some applications of the ROMs are the following:

- Stores the BIOS
- Stores fonts in printers
- Used in industry e.g. to hold a program that controls a machine that manufactures parts.
- Used as secondary storage like USB sticks.

BIOS

One of the most common uses of flash memory is for the basic input/output system of your computer, commonly known as the BIOS. The BIOS is special firmware that interfaces the major hardware components of the computer with the operating system

The BIOS software has a number of different roles:

- The most important role is to load the operating system.
- A power-on self-test (POST) for all of the different hardware components in the system to make sure that everything is working properly.
- Activating other BIOS chips on different cards installed in the computer e.g. some bus systems (e.g. SCSI) and graphics cards often have their own BIOS chips.
- Providing a set of low-level routines that the operating system uses to interface to different hardware devices (it is these routines that give the BIOS its name). They manage the keyboard, screen, serial and parallel ports, etc. especially when the computer is booting.
- Managing a collection of settings for the hard disks, clock, etc.

When you turn on your computer, the BIOS does several things. This is its usual sequence:

1. Check the CMOS Setup for custom settings
2. Load the interrupt handlers and device drivers
3. Initialize registers and power management
4. Perform the power-on self-test (POST)
5. Display system settings
6. Determine which devices are bootable
7. Initiate the bootstrap sequence

The first thing the BIOS does is check the information stored in a tiny (64 bytes) amount of RAM located on a complementary metal oxide semiconductor (CMOS) chip. The CMOS Setup provides detailed information particular to your system and can be altered as your system changes. The BIOS uses this information to modify or supplement its default programming as needed.

Interrupt handlers are small pieces of software that act as translators between the hardware components and the operating system. For example, when you press a key on your keyboard, the signal is sent to the keyboard interrupt handler, which tells the CPU what it is and passes it on to the operating system.

A BIOS chip

A device driver (hardware driver) is a group of software files that enable one or more hardware devices to communicate with the computer's operating system. Without drivers, a hardware device such as a computer printer would not be able to work with the computer. If the appropriate driver is not installed, the device may not function properly if at all. If problems or conflicts are encountered with a driver, the manufacturer will release driver updates to fix the problems. Since the BIOS is constantly intercepting signals to and from the hardware, it is usually copied, or shadowed, into RAM to run faster.

If the BIOS finds any errors during the POST, it will notify the user by a series of beeps or a text message displayed on the screen.

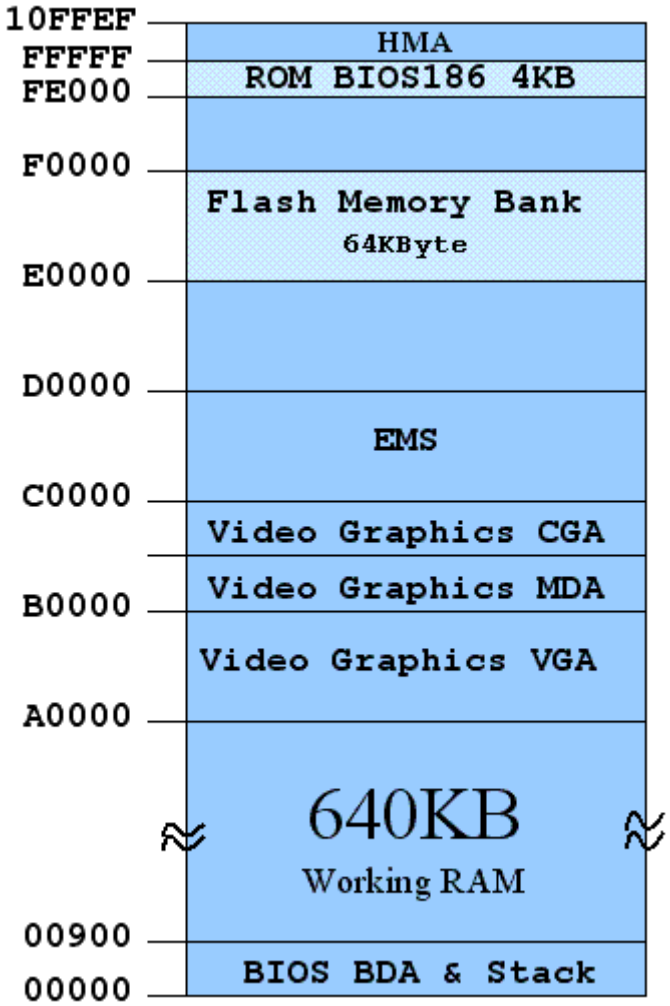
The BIOS uses CMOS technology to save any changes made to the computer's settings. With this technology, a small lithium or Ni-Cad battery can supply enough power to keep the data for years. In fact, some of the newer chips have a 10-year, tiny lithium battery built right into the CMOS chip!

Three advantages of CMOS technology are the following:

1. High operating speed
2. Efficient use of energy
3. High degree of noise immunity

2.3.4 Memory Map

A memory map shows the location of programs and data in memory.



The memory map of a particular system

2.4 I/O Subsystem

The goal of the I/O subsystem is to provide a uniform interface to the wide range of devices. The interface between the operating system and a device is called “device driver” and this is code that controls a device.

2.4.1 I/O Addressing

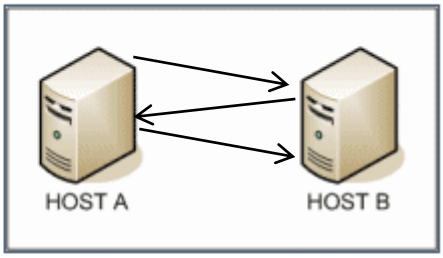
There are two fundamental architectures for mapping Special Function Registers into the memory space.

1. Isolated (separated) I/O i.e. I/O space and memory space are separated. Access to the I/O control registers requires special I/O instructions. (Devices usually have registers where device driver places commands, addresses, and data to write, or read data from registers after command execution. There are the data-in register, data-out register, status register, control register etc.
2. Memory-mapped I/O. The memory-mapped I/O maps the I/O control registers into the CPU's memory address space. Reads and writes to the control registers are done via absolute memory addresses. No special instructions are required.
3. A combination of both is also possible.

Isolated I/O and Memory Mapped I/O

2.4.2 Handshaking

In communication, handshaking is the automated process for negotiation of setting up a communication channel between entities. Handshaking occurs before the transfer of data or any other communication and just after the establishment of the physical channel between the two entities.

EVENT	DIAGRAM
Host A **sends** a TCP **SYN**chronize packet to Host B Host B receives A's **SYN** Host B **sends** a **SYN**chronize-**ACK**nowledgement Host A receives B's **SYN-ACK** Host A **sends** **ACK**nowledge Host B receives **ACK**. ***TCP socket connection is ESTABLISHED.***	 TCP Three Way Handshake (SYN,SYN-ACK,ACK)

TCP three-way handshake

Handshaking is helpful while establishing communication between two devices, as it can help in checking the quality and speed of the transmission and also the necessary authority needed for same.

The handshake can provide the necessary information or protocols for the sender and receiver. It allows the receiving device to know how to receive the input data from the sender and then output the received data in the necessary format applicable to the receiver. It also provides the provisions of how the communication between the devices should continue. This is especially required when the devices are foreign to each other, like a computer to a modem, server, etc.

The parameters involved in handshaking can be hardware protocols, alphabet coding, interrupt procedures or even parity.

2.4.3 Interrupts

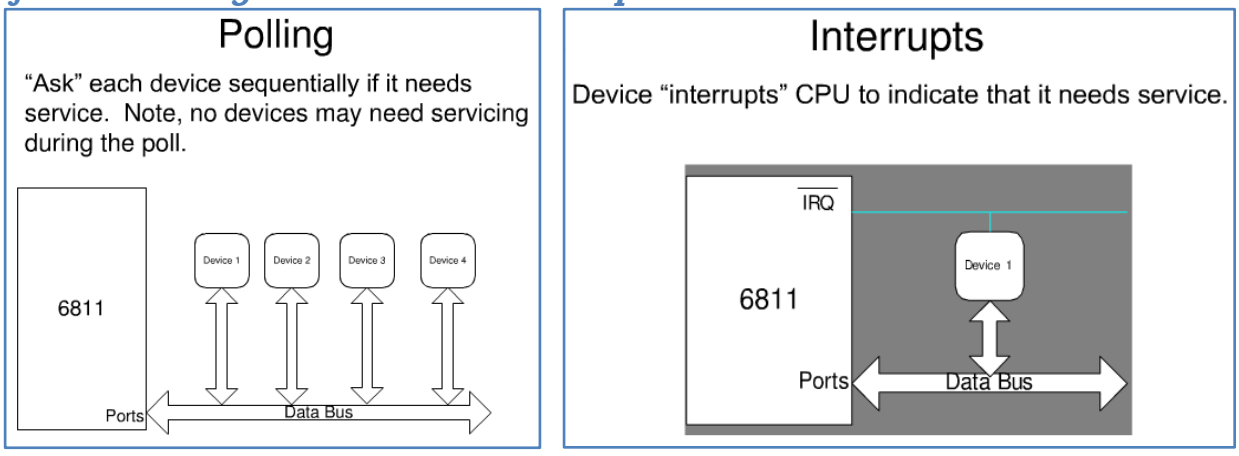
What is an Interrupt?

An **interrupt** is a signal informing a program that an event has occurred. When a program receives an interrupt signal, it takes a specified action (which can be to ignore the signal). Interrupt signals can cause a program to suspend itself temporarily to service the interrupt.

Interrupt signals can come from a variety of sources. For example, every keystroke generates an interrupt signal. Interrupts can also be generated by other devices, such as a printer, to indicate that some event has occurred. These are called **hardware interrupts**. Interrupt signals initiated by programs are called **software interrupts**. A software interrupt is also called a **trap** or an **exception**. An example of an exception is when a program attempts to perform a division by zero.

Each type of interrupt is associated with an **interrupt handler**. This is a routine that takes control when the interrupt occurs. For example, when you press a key on your keyboard, this triggers a specific interrupt handler. The complete list of interrupts and associated interrupt handlers is stored in a table called the **interrupt vector table**.

Software Polling and Vectored Interrupts



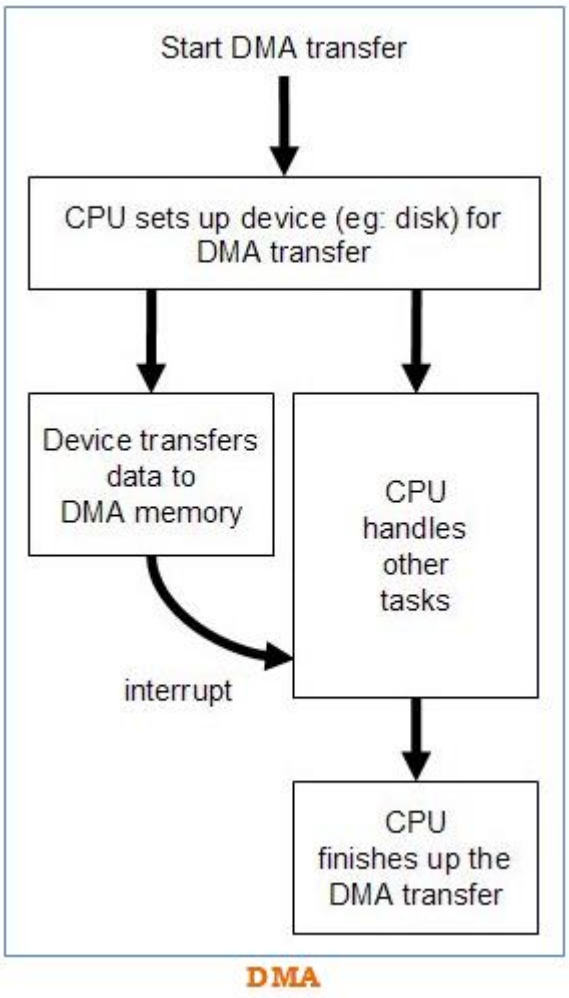
Polling and (Vectored) Interrupts

Polling refers to the situation where the CPU periodically checks each device to see if it needs service. One disadvantage of this method is that it takes CPU time even when no

requests are pending. However the system can be efficient in the case of a high frequency of interrupts. Polling is like picking up your phone every few seconds to see if you have a call.

The system of **vectored interrupts** gives a wire (interrupt line) to each device and by means of this line it can signal interrupts to the processor. Each interrupt is associated with an interrupt handler.

2.4.4 DMA



DMA

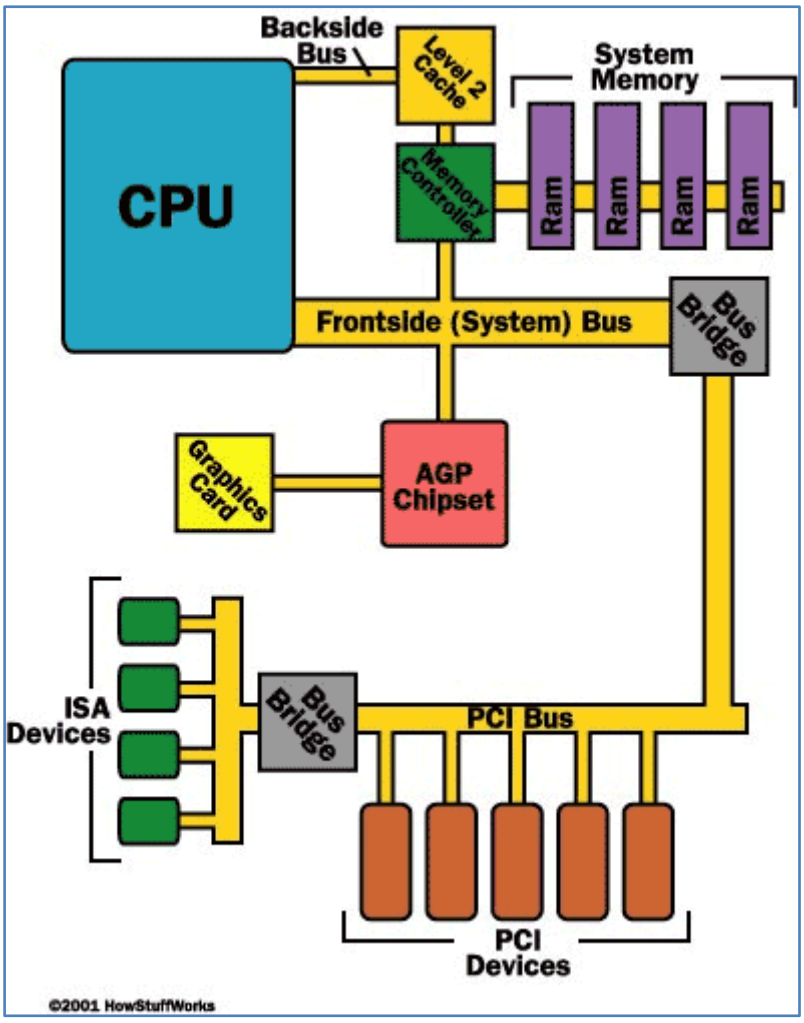
Direct Memory Access (DMA) is a capability provided by some computer bus architectures that allows data to be sent directly from an attached device (such as a disk drive) to the memory on the computer's motherboard. The microprocessor is freed from involvement with the data transfer, thus speeding up overall computer operation.

Usually a specified portion of memory is designated as an area to be used for direct memory access. In the ISA bus standard, up to 16 megabytes of memory can be addressed for DMA. The EISA and Micro Channel Architecture standards allow access to the full range of memory addresses (assuming they're addressable with 32 bits). Peripheral Component Interconnect accomplishes DMA by using a bus master (with the microprocessor "delegating" I/O control to the PCI controller).

An alternative to DMA is the Programmed Input/Output (PIO) interface in which all data transmitted between devices goes through the processor. A newer protocol for the ATA/IDE interface is Ultra DMA, which provides a burst data transfer rate up to 33 MB (megabytes) per second. Hard drives that come with Ultra DMA/33 also support PIO modes 1, 3, and 4, and multiword DMA mode 2 (at 16.6 megabytes per second).

2.4.5 PCI Bus

The idea of a bus is simple - it lets you connect components to the computer's processor. Some of the components that you might want to connect include hard disks, memory, sound systems, video systems and so on. The advantage of a bus is that it makes parts more interchangeable. If you want to get a better graphics card, you simply unplug the old card from the bus and plug in a new one. If you want two monitors on your computer, you plug two graphics cards into the bus. And so on. PCI stands for Peripheral Component Interconnect (PCI).



The illustration above shows how the various buses connect to the CPU.

System Bus vs. PCI Bus

Twenty or 30 years ago, the processors were so slow that the processor and the bus were synchronized - the bus ran at the same speed as the processor, and there was one bus in the machine. Today, the processors run so fast that most computers have two or more buses. Each bus specializes in a certain type of traffic.

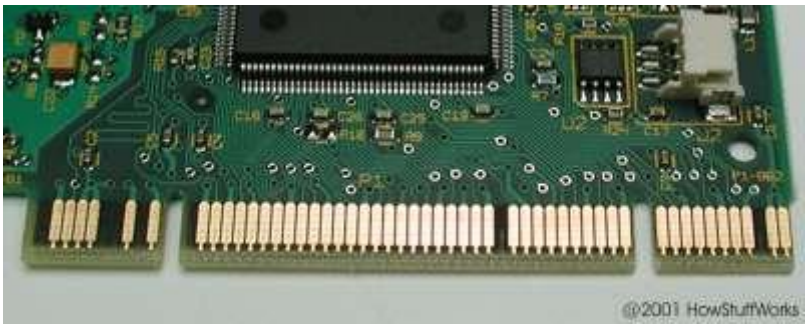
A typical desktop PC today has two main buses:

- The system bus. It is used to connect the major components of a computer system. It is made up of a data bus, an address bus and a control bus.
- A slower bus for communicating with things like hard disks and sound cards. One very common bus of this type is known as the PCI bus. These slower buses connect to the system bus through a bridge, which is a part of the computer's chipset and acts as a traffic cop, integrating the data from the other buses to the system bus.

Technically there are other buses as well. For example, the Universal Serial Bus (USB) is a way of connecting things like cameras, scanners and printers to your computer. It uses a thin wire to connect to the devices, and many devices can share that wire simultaneously. Firewire is another bus, used today mostly for video cameras and external hard drives.

Frontside Bus, Backside Bus and PCI Cards

The frontside bus is a physical connection that actually connects the processor to most of the other components in the computer, including main memory (RAM), hard drives and the PCI slots. These days, the frontside bus usually operates at 400-MHz, with newer systems running at 800-MHz.



A PCI card

The backside bus is a separate connection between the processor and the Level 2 cache. This bus operates at a faster speed than the frontside bus, usually at the same speed as the processor, so all that caching works as efficiently as possible (today the cache is on the processor itself so the backside bus isn't really a bus anymore).

PCI can connect 5 external components. Also, you can have more than one PCI bus on the same computer, although this is rarely done.

This motherboard has four PCI slots.

PCI originally operated at 33 MHz using a 32-bit-wide path. Revisions to the standard include increasing the speed from 33 MHz to 66 MHz and doubling the bit count to 64. Currently, PCI-X provides for 64-bit transfers at a speed of 133 MHz for an amazing 1 GBps (gigabyte per second) transfer rate!

PCI cards use 47 pins to connect (49 pins for a mastering card, which can control the PCI bus without CPU intervention). The PCI bus is able to work with so few pins because of hardware multiplexing, which means that the device sends more than one signal over a single pin. Also, PCI supports devices that use either 5 volts or 3.3 volts.

Plug and Play

Plug and Play (PnP) means that you can connect a device or insert a card into your computer and it is automatically recognized and configured to work in your system. PnP is a simple concept, but it took a concerted effort on the part of the computer industry to make it happen. Intel created the PnP standard and incorporated it into the design for PCI.

2.4.6 PCI-X and PCIe Buses

Both PCI-X and PCIe are 64-bit, high-bandwidth versions of the now decade old PCI interconnect standard. But that is the only commonality between them. They work in completely different ways, have different applications, and have physically incompatible architecture.

Basic Differences

PCI-X and PCIe

PCI-X uses a parallel interconnect along a bus that is shared with other PCI-X devices, just like PCI. In fact, PCI-X is best thought of as "**PCI**-e**X** tended", as it is simply an extension of the legacy PCI 32-bit format, with which it is backward-compatible. It differs mainly in the fact that the bus is now 64-bits wide, and runs at higher frequencies (now up to 533MHz, compared to 66MHz - the fastest PCI frequency).

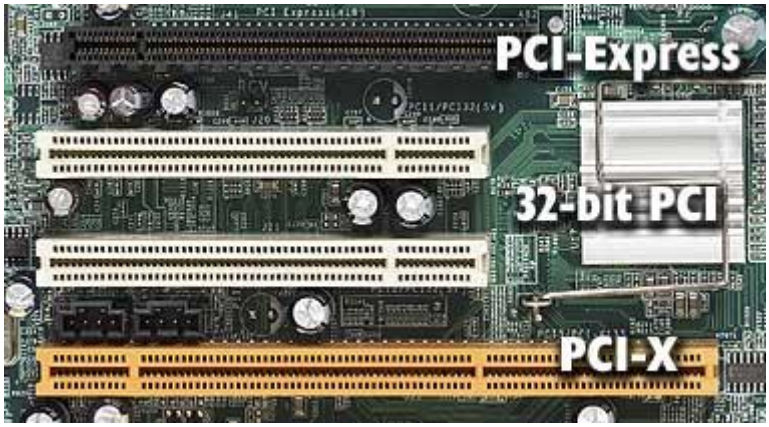
PCI-Express, on the other hand, is **serial and** it uses a radically new architecture, having little to do with old PCI. Furthermore, PCI-Express has the unique capability of **multiplying up** individual data "lanes" that can deliver up to 16 times the bandwidth of a single lane. This is why you will always see PCI-Express slots referred to as "PCI-Express*4" or "PCI-Express*16" etc.

Applications

PCI-X has been with us in the server and workstation arena for some time now, as a bus for high-bandwidth server peripherals such as RAID Controllers and Gigabit Ethernet. PCI-Express, on the other hand, is brand-new, and is intended to replace AGP in the desktop market and ultimately be the de-facto high-bandwidth peripheral bus across all markets.

Hardware that benefits from 64-bit PCI include:

- High-performance graphics cards (PCI-Express only) in the 3D Gaming desktop and graphic intensive workstation markets.
- U320 SCSI Controllers for high-speed hard disk access.
- Multi-port Serial ATA RAID Controllers for terabyte storage arrays.
- Gigabit Ethernet for high-speed networking.
- IEEE1394b ("Firewire 800") for ultra-high bandwidth peripherals, such as external hard drives and DV camcorders.



Comparing the slots used by PCI, PCI-X and PCI-Express

Which is better - PCI-X or PCI-Express?

Well, in a short answer, PCI-Express is the superior technology. Ultimately it has higher bandwidth, because you can just bundle up single PCI-Express lanes into x4 or x16 buses. There are also fewer steps needed to process instructions between the CPU and PCI device compared to PCI-X, thus reducing latency.

A single PCI-Express lane can deliver a maximum of 2.5 gigabits per second in a single direction, with a theoretical maximum of 5Gbps in both directions. PCI-Express slots are available currently in up to 16 lanes i.e. 16x5Gbps = 80Gbps maximum bandwidth in both directions. That's 10 gigabytes per second of data transfer (10 GB/s)!

PCI-X doesn't work in quite the same way, since it uses a shared data bus between slots/devices, which is why most server boards usually have more than one PCI-X bus, to prevent I/O bottlenecks. However, that said, with the PCI-X 2.0 specification, bus frequencies of 533MHz are now possible, providing a theoretical bandwidth of up to 34Gbps (4GB/s).

To get these figures into perspective, consider that the hottest Dual Channel SCSI controller cards can deliver up to 1GB/s of data transfer, and so these speed limits on PCI-X and PCI-Express are not likely to be limiting for any devices in the near future.

2.4.7 AGP

On your computer you point, you click; you drag and you drop. Files open and close in separate windows. Movies play, pop-ups pop, and video games fill the screen, immersing you in a world of 3-D graphics. This is the stuff we're used to seeing on our computers. It all started in 1973, when Xerox completed the Alto, the first computer to use a graphical user interface. This innovation forever changed the way the people work with their computers.

Today, every aspect of computing, from creating animation to simple tasks such as word processing and e-mail, uses lots of graphics to create a more intuitive work environment for the user. The hardware to support these graphics is called a graphics card. The way this card connects to your computer is of key importance in your computer's ability to render graphics. AGP (Accelerated Graphics Port) enables your computer to have a dedicated way to communicate with the graphics card, enhancing both the look and speed of your computer's graphics.

In 1996, Intel introduced AGP as a more efficient way to deliver the streaming video and real-time-rendered 3-D graphics that were becoming more prevalent in all aspects of computing. Previously, the standard method of delivery was the Peripheral Component Interconnect (PCI) bus where graphical information was delivered with other non-graphical information.

Typical example of an AGP-based graphics card

AGP is based on the design of the PCI bus; but unlike a bus, it provides a dedicated point-to-point connection from the graphics card to the CPU. With a clear path to the CPU and system memory, AGP provides a much faster, more efficient way for your computer to get the information it needs to render complex graphics.

AGP Graphics Rendering

AGP uses the following techniques to achieve fast communication of graphical data.

- Dedicated Port - There are no other devices connected to the AGP other than the graphics card. With a dedicated path to the CPU, the graphics card can always operate at the maximum capacity of the connection.
- Pipelining - This method of data organization allows the graphics card to receive and respond to multiple packets of data in a single request. Here's a simplified example of this: With AGP, the graphics card can receive a request for all of the information needed to render a particular image and send it out all at once. With PCI, the graphics card would receive information on the height of the image and wait... then the length of the image, and wait... then the width of the image, and wait... combine the data, and then send it out.
- Sideband addressing - Like a letter, all requests and information sent from one part of your computer to the next must have an address containing "To" and "From." The problem with PCI is that this "To" and "From" information is sent with the working data all together in one packet. This is the equivalent of including an address card inside the envelope when you send a letter to a friend: Now the post office has to open the envelope to see the address in order to know where to send it. This takes up the post office's time. In addition, the address card itself takes up room in the envelope, reducing the total amount of stuff you can send to your friend. With sideband addressing, the AGP issues eight additional lines on the data packet just for addressing. This puts the address on the outside of the envelope, so to speak, freeing up the total bandwidth of the data path used to transfer

information back and forth. In addition, it unclogs system resources that were previously used to open the packet to read the addresses.

Bus and Frequency	Peak 32-Bit Transfer Rate	Peak 64-Bit Transfer Rate
33-MHz PCI	133 MB/sec	266 MB/sec
66-MHz PCI	266 MB/sec	532 MB/sec
100-MHz PCI-X	Not applicable	800 MB/sec
133-MHz PCI-X	Not applicable	1 GB/sec
AGP8X	2.1 GB/sec	Not applicable

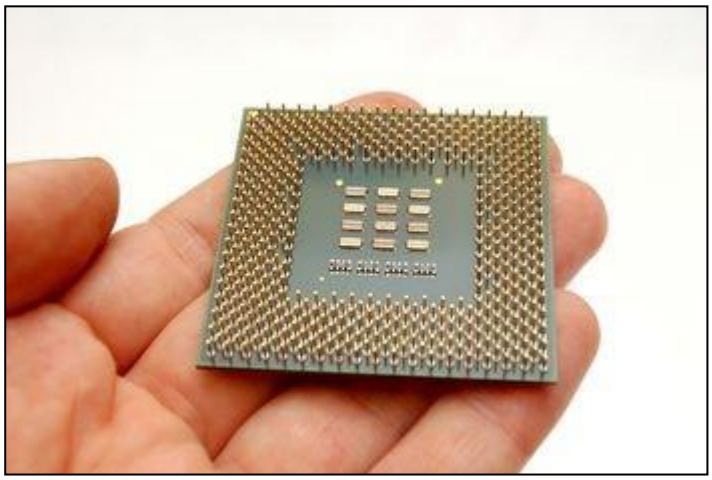
Bandwidth of PCI, PCI-X, and AGP Buses

2.5 CPU

The processor in a computer is the module that executes instructions and programs (a program is a sequence of instructions). Today the terms processor and CPU have the same meaning. In the old days the term CPU used to refer to the combination containing the processor and main memory.

2.5.1 Overview of Main Components

The microprocessor is a silicon chip that contains a whole processor. At the heart of all personal computers and most workstations sits a microprocessor.



CPU

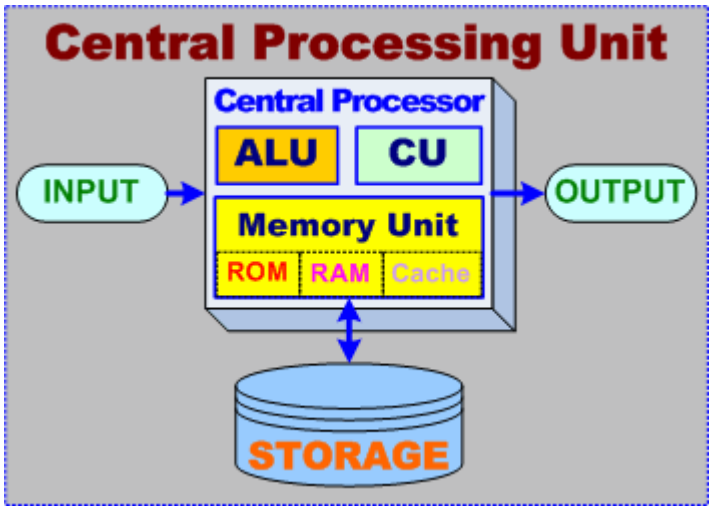
The three basic characteristics that differentiate microprocessors are:

- Instruction set: The set of instructions that the microprocessor can execute.
- Bandwidth: The number of bits processed in a single instruction.
- Clock speed: Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In addition to bandwidth and clock speed, microprocessors are classified as being either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

The most important parts of a processor are:

- Control Unit (CU)
- Arithmetic-Logic Unit (ALU)
- Registers
- Cache

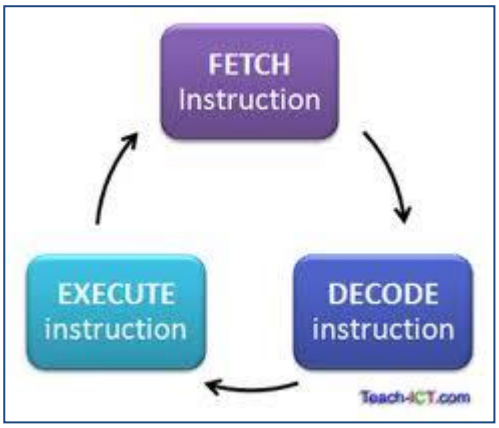


Central Processing Unit

2.5.2 The Fetch-Decode-Execute Cycle

The role of the processor in a computer is to execute instructions. These instructions are given in the form of a program. The processor follows the Fetch-Execute cycle (this is also called the Fetch-Decode-Execute cycle). In very simple terms the Fetch-Execute cycle performs the following sequence of commands.

1. Bring the next command from main memory.
2. Interpret this command.
3. Execute this command.
4. Return to step 1.



The Fetch-Execute Cycle

2.5.3 Control Unit

The control unit extracts instructions from memory and decodes and executes them, and sends the necessary signals to the ALU to perform the operation needed. Control Units are either hardwired or micro-programmed. The control unit communicates with the arithmetic logic unit and the system memory.

2.5.4 Arithmetic-Logic Unit

The ALU is where all the arithmetic and logical operations are carried out. Apart from the basic arithmetic operations (addition, subtraction, multiplication, division) the ALU performs operations involving logic (AND, OR, NOT, comparison between two values to see if they are equal or which one is greater than the other).

2.5.5 Registers

The registers are what the CPU uses for temporary storage of data. They are not part of any of the system memory, but are instead additional storage locations that are on the CPU itself. This makes registers very fast for the CPU to use. The registers are controlled by the control unit and are used to hold and transfer instructions, and perform the logical and arithmetic operations.

Registers with different functions

Registers are assigned specific functions. Some are listed below:

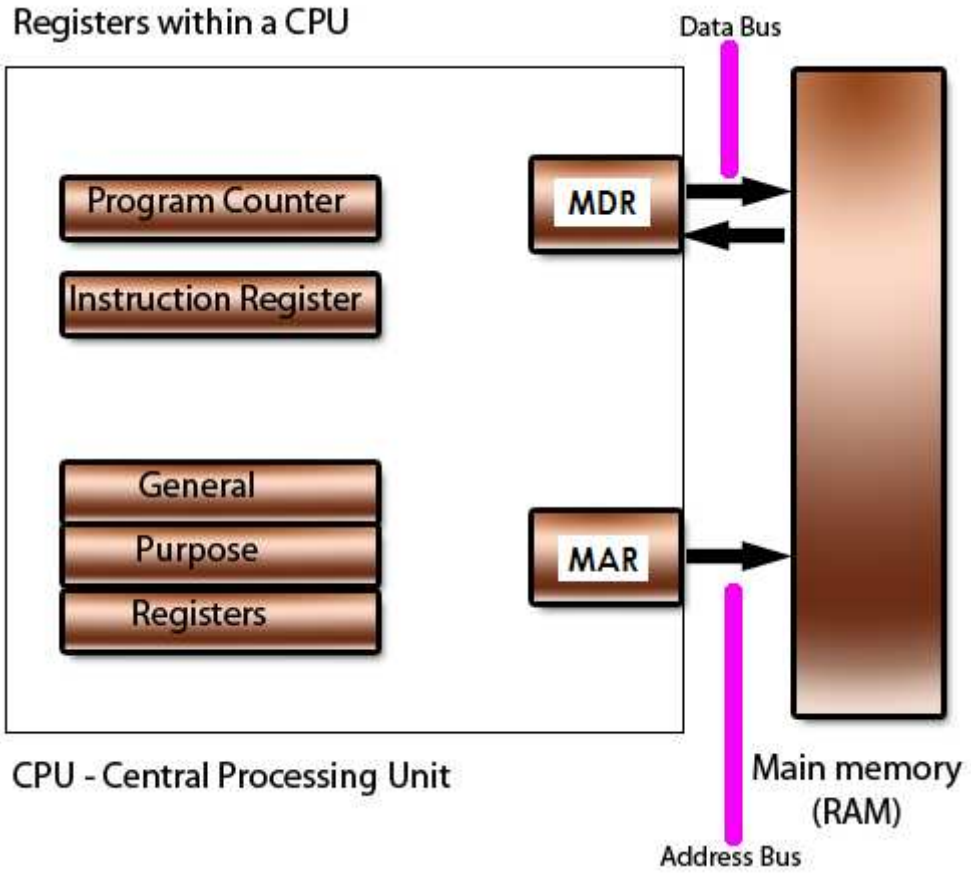
- Accumulators: they hold values are can perform operations with these values
- Address Registers: they can store memory addresses.
- Data Registers: they can temporarily store data.
- Miscellaneous: general purpose used for several functions.

Particular Registers

Particularly important registers are the following:

- CIR: Current Instruction Register (CIR): This register (also called IR) is the part of a CPU's control unit that stores the instruction currently being executed or decoded.

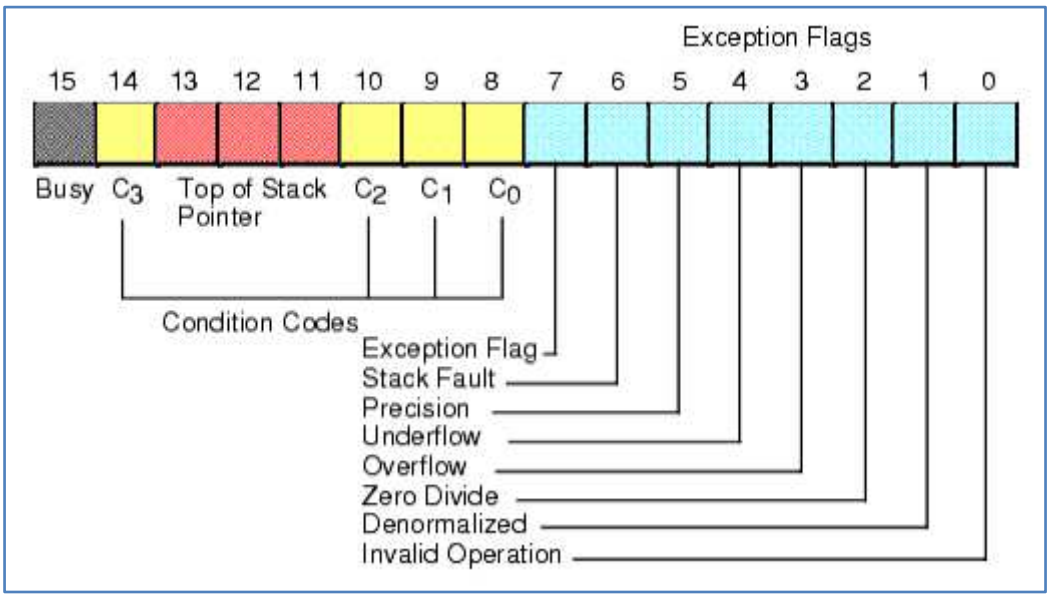
- PC: Program Counter (PC): The PC (also called the 'instruction pointer' or 'instruction address register') holds the address of the next instruction to be executed. In most processors, the instruction pointer is incremented automatically after fetching a program instruction.



Registers

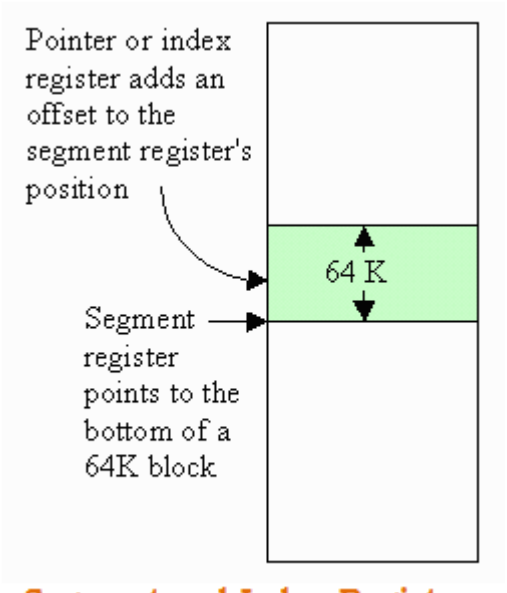
- MAR: Memory Address Register (MAR): This register either stores the memory address from which data will be fetched to the CPU or the address to which data will be sent and stored.
- MDR: Memory Data Register (MDR): This register contains the data to be stored in the computer storage (e.g. RAM), or the data after a fetch from the computer storage. It acts like a buffer.
- Status Register: The status register (flag register) is a collection of 1-bit values which reflect the current state of the processor and the results of recent operations. Here are some examples:
    - Carry bit: set if the last arithmetic operation ended with a leftover carry bit coming off the left end of the result. This signals an overflow on unsigned numbers.
    - Parity bit: set if the low-order byte of the last data operation contained an even number of 1 bits (that is, it signals an even parity condition).
    - Zero bit: set if the last computation had a zero result. After a comparison this indicates that the values compared were equal (since their difference was zero).
    - Sign bit: set if the last computation had a negative result (a 1 in the leftmost bit).

- Interrupt bit: when set, interrupts are enabled.
- Overflow bit: set if the last arithmetic operation caused an overflow



Status (Flag) Register

- Segment register: A segment-register points to the base of the current segment being addressed. A segmented address space is a kind of memory addressing where each byte is referenced by a base number (the segment) plus an offset. An x86-based PC running in 16-bit mode uses 64KB segments, and a segment register always points to the base of the segment that is currently being addressed.
- Index Registers: An index register holds the current, relative position of an item. The address held in the index register is added to that held in the segment register to produce the physical address. An index register can also hold the index of an element in a table (array).
- Stack Register: It holds the current address of the top of a region of separate computer memory known as the stack.



Segment and Index Registers

  - The stack register is important because, without it, a computer would need to implement a slower, more error-prone method of tracing the flow of execution of a program.
  - In most system architectures, the stack register is a dedicated register so it is not accidentally accessed when working with other memory registers.
- Control Register: A control register is a register which changes or controls the general behaviour of a CPU or other digital device. Common tasks performed by control registers include interrupt control, switching the addressing mode, paging

control, and coprocessor control. Some examples of control registers are the following:

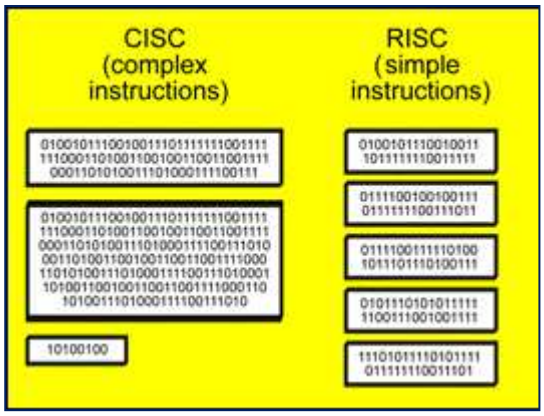
- Enables/disable the memory cache
- Determines whether the CPU can write to pages marked read-only

2.5.6 CISC and RISC

CISC stands for Complex Instruction Set Computer. Most personal computers use a CISC architecture, in which the CPU supports as many as two hundred instructions. An alternative architecture, used by many workstations and also some personal computers, is RISC (reduced instruction set computer), which supports fewer instructions.

CISC computers were designed with a full set of computer instructions that were intended to provide needed capabilities in the most efficient way. Later, it was discovered that, by reducing the full set to only the most frequently-used instructions, the computer would get more work done in a shorter amount of time for most applications.

Macintosh computers use a RISC microprocessor. Intel's Pentium microprocessors are CISC.



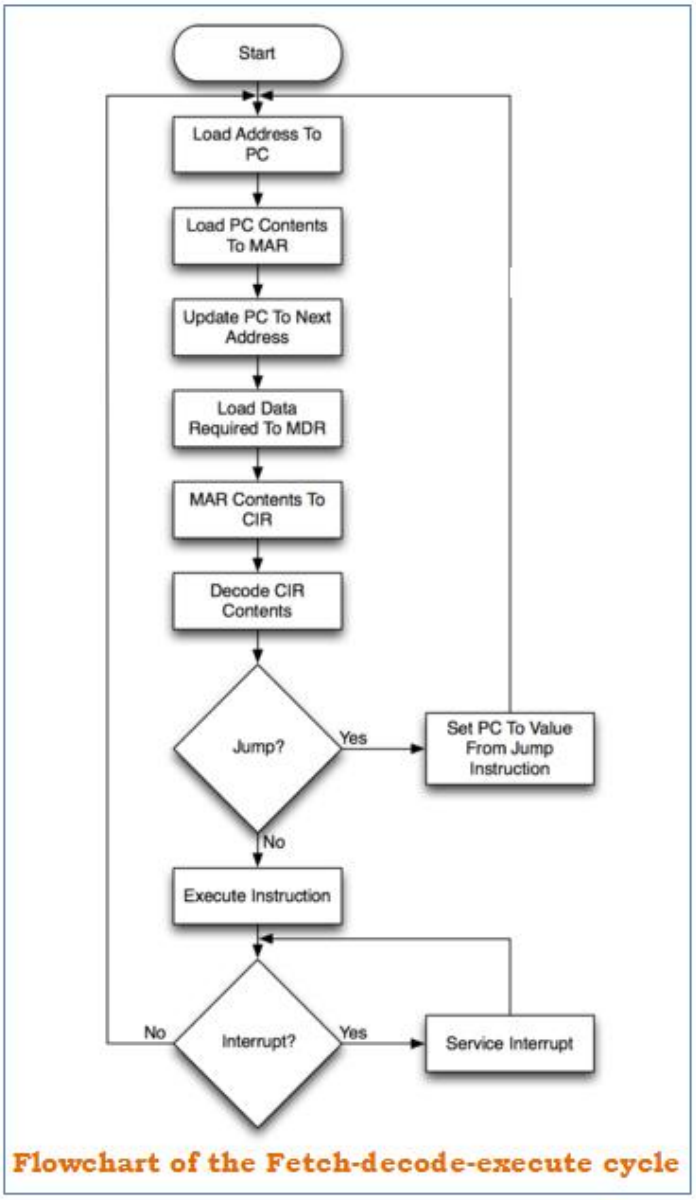
CISC and RISC

One advantage of RISC computers is that they can execute their instructions very fast because the instructions are so simple. Another, perhaps more important advantage, is that RISC chips require fewer transistors, which makes them cheaper to design and produce.

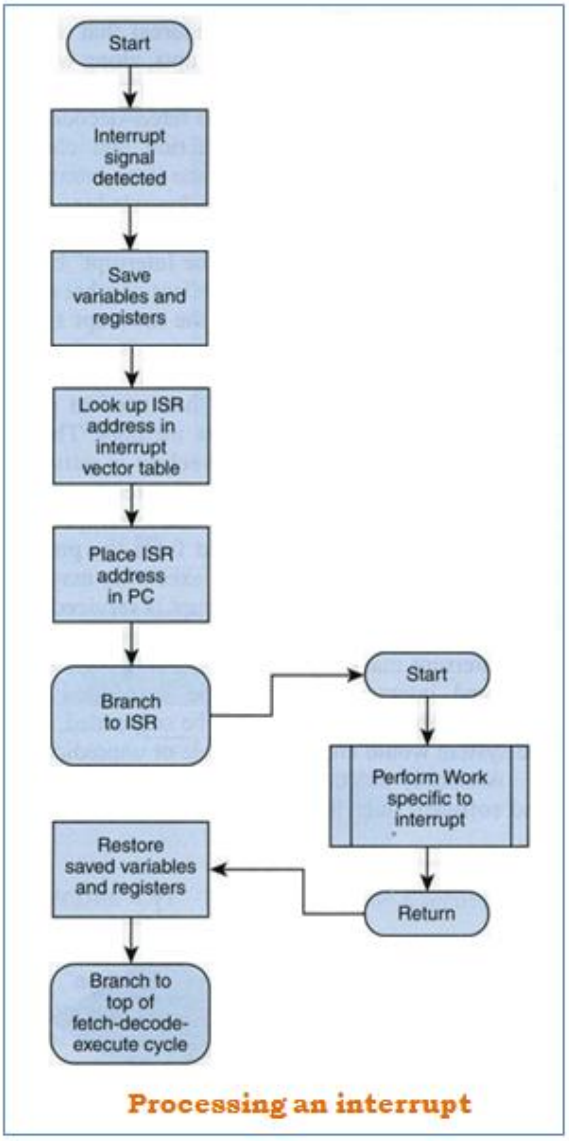
2.5.7 Fetch-decode-execute Cycle

A more detailed 'fetch-decode-execute cycle' is the following:

1. The Program Counter (PC) contains the address of the next instruction to be fetched. The address contained in the PC is copied to the Memory Address Register (MAR).
2. The instruction is copied from the memory location contained in the MAR and placed in the Memory Data Register (MDR).
3. The entire instruction is copied from the MDR and placed in the Current Instruction Register (CIR)
4. The PC is incremented so that it points to the next instruction to be fetched
5. The address part of the instruction is placed in the MAR.
6. The instruction is decoded and executed.
7. The processor checks for interrupts (signals from devices or other sources seeking the attention of the processor) and then it either branches to the relevant interrupt service routine or starts the cycle again.



Flowchart of the Fetch-decode-execute cycle



Processing an interrupt

2.6 Caches

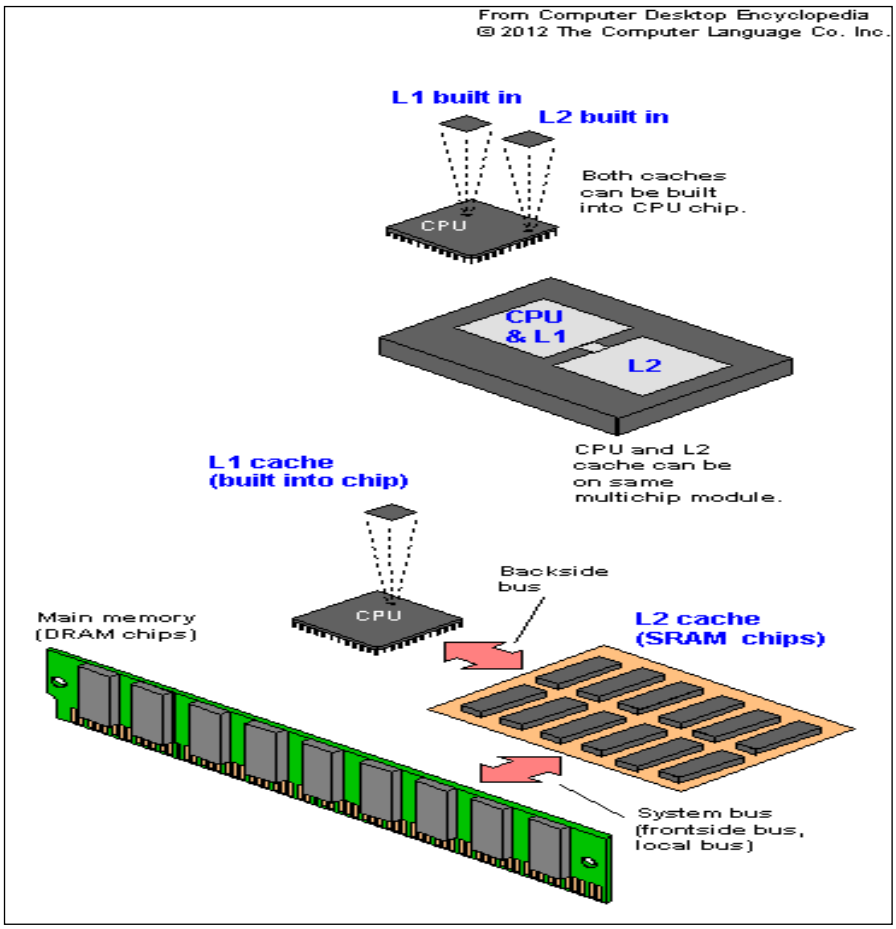
Cache is a special high-speed storage mechanism. Two types of caching are commonly used in personal computers:

- memory caching
- disk caching

A memory cache, sometimes called a 'cache store' or 'RAM cache', is a portion of memory made of high-speed static RAM (SRAM) instead of the slower and cheaper dynamic RAM (DRAM) used for main memory. Memory caching is effective because most programs access the same data or instructions over and over. By keeping as much of this information as possible in SRAM, the computer avoids accessing the slower DRAM.

Some memory caches are built into the architecture of microprocessors. The Intel 80486 microprocessor, for example, contains an 8K memory cache, and the Pentium has a 16K cache. Such internal caches are often called Level 1 (L1) caches. Most modern PCs also

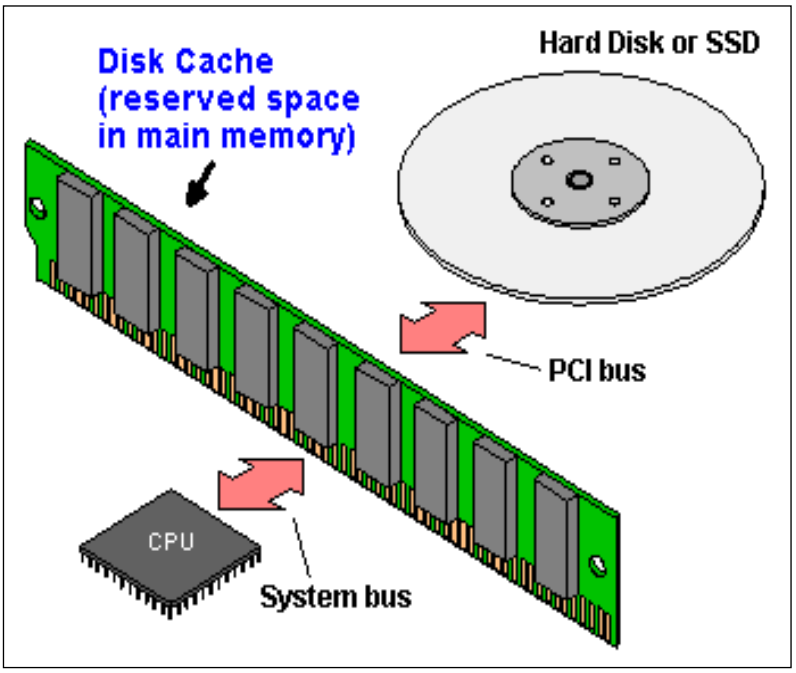
come with external cache memory, called Level 2 (L2) caches. These caches sit between the CPU and the DRAM. Like L1 caches, L2 caches are composed of SRAM but they are much larger.



Caches

Disk caching works under the same principle as memory caching, but instead of using high-speed SRAM, a disk cache uses conventional main memory. The most recently accessed data from the disk (as well as adjacent sectors) is stored in a memory buffer. When a program needs to access data from the disk, it first checks the disk cache to see if the data is there. Disk caching can dramatically improve the performance of applications, because accessing a byte of data in RAM can be thousands of times faster than accessing a byte on a hard disk.

When data is found in the cache, it is called a cache hit, and the effectiveness of a cache is judged by its hit rate. Many cache systems use a technique known as 'smart caching', in which the system can recognize certain types of frequently used data. The strategies for determining which information should be kept in the cache constitute interesting problems in computer science.



Disk Cache

Therefore as a summary we can say:

- L1
  - Stands for Level 1
  - Built inside a microprocessor
  - Composed of SRAM
- L2
  - Stands for Level 2
  - Also called
    - Memory cache
    - Cache store
    - RAM cache

    - Made of SRAM
    - Much larger than L1
    - Memory caching is effective because most programs access the same data or instructions over and over. By keeping as much of this information as possible in SRAM, the computer avoids accessing the slower DRAM.
- L3
    - As more and more processors begin to include L2 cache into their architectures, Level 3 cache is now the name for the extra cache built into motherboards between the microprocessor and the main memory. Quite simply, what was once L2 cache on motherboards now becomes L3 cache when used with microprocessors containing built-in L2 caches.
- Disk caching
    - Made of DRAM
    - It holds the most recently accessed data from the disk
- Cache hit
    - This occurs when data is found in the cache

2.7 I/O Peripherals

2.7.1 Serial Ports

Two serial ports on the back of a PC

The serial port has been an integral part of most computers for more than 20 years. Although many of the newer systems have done away with the serial port completely in favour of USB connections, most modems still use the serial port, as do some printers, PDAs and digital cameras. Few computers have more than two serial ports.

UART

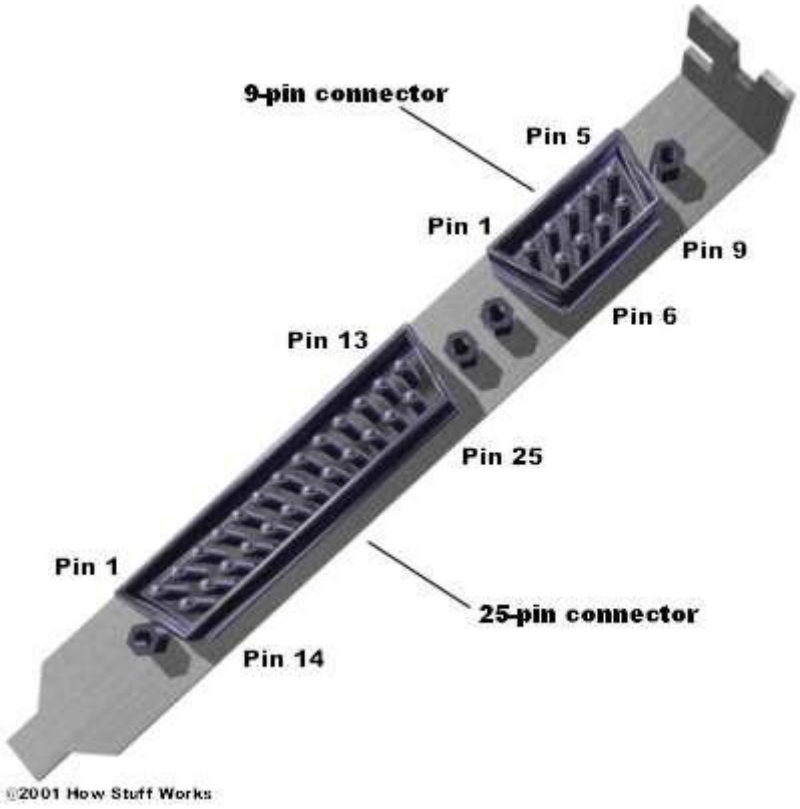
All computer operating systems in use today support serial ports, because serial ports have been around for decades. Parallel ports are a more recent invention and are much faster than serial ports. USB ports are only a few years old, and will likely replace both serial and parallel ports completely over the next several years.

The name "serial" comes from the fact that a serial port "serializes" data. That is, it takes a byte of data and transmits the 8 bits in the byte one at a time. The advantage is that a serial port needs only one wire to transmit the 8 bits (while a parallel port needs 8). The disadvantage is that it takes 8 times longer to transmit the data than it would if there were 8 wires. Serial ports lower cable costs and make cables smaller.

Before each byte of data, a serial port sends a start bit, which is a single bit with a value of 0. After each byte of data, it sends a stop bit to signal that the byte is complete. It may also send a parity bit.

Serial ports, also called communication (COM) ports, are bi-directional. Bi-directional communication allows each device to receive data as well as transmit it. Serial devices use different pins to receive and transmit data - using the same pins would limit communication to half-duplex, meaning that information could only travel in one direction at a time. Using different pins allows for full-duplex communication, in which information can travel in both directions at once.

Serial ports rely on a special controller chip, the Universal Asynchronous Receiver/Transmitter (UART), to function properly. The UART chip takes the parallel output of the computer's system bus and transforms it into serial form for transmission through the serial port. In order to function faster, most UART chips have a built-in buffer of anywhere from 16 to 64 kilobytes. This buffer allows the chip to cache data coming in from the system bus while it is processing data going out to the serial port. While most standard serial ports have a maximum transfer rate of 115 Kbps (kilobits per second), high speed serial ports, such as Enhanced Serial Port (ESP) and Super Enhanced Serial Port (Super ESP), can reach data transfer rates of 460 Kbps.



Close-up of 9-pin and 25-pin serial connectors

The external connector for a serial port can be either 9 pins or 25 pins. Originally, the primary use of a serial port was to connect a modem to your computer. The pin assignments reflect that. Each pin is assigned a particular function for example:

- Carrier Detect - Determines if the modem is connected to a working phone line.
- Receive Data - Computer receives information sent from the modem.
- Transmit Data - Computer sends information to the modem.
- Data Terminal Ready - Computer tells the modem that it is ready to talk.

Voltage sent over the pins can be in one of two states, 'on' or 'off'. 'On' (binary value "1") means that the pin is transmitting a signal between -3 and -25 volts, while 'off' (binary value "0") means that it is transmitting a signal between +3 and +25 volts.

2.7.2 Parallel Ports

If you have a printer connected to your computer, there is a good chance that it uses the parallel port. While USB is becoming increasingly popular, the parallel port is still a commonly used interface for printers. Parallel ports can be used to connect a host of popular computer peripherals:

- Printers
- Scanners
- CD burners
- External hard drives
- Iomega Zip removable drives
- Network adapters
- Tape backup drives

Female side and male side of a parallel port

Parallel ports were originally developed by IBM as a way to connect a printer to your PC. When IBM was in the process of designing the PC, the company wanted the computer to work with printers offered by Centronics, a top printer manufacturer at the time.

When a PC sends data to a printer or other device using a parallel port, it sends 1 byte at a time. These 8 bits are transmitted parallel to each other, as opposed to the same eight bits being transmitted serially (all in a single row) through a serial port. The standard parallel port is capable of sending 50 to 100 kilobytes of data per second.

Let us look at the function of some of the pins:

- Pins 2 through 9 are used to carry data. To indicate that a bit has a value of 1, a charge of 5 volts is sent through the correct pin. No charge on a pin indicates a value of 0. This is a simple but highly effective way to transmit digital information over an analog cable in real-time.
- Pin 10 sends the acknowledge signal from the printer to the computer.
- The printer lets the computer know if it is out of paper by sending a charge on Pin 12.
- If the printer has any problems, it drops the voltage to less than 0.5 volts on Pin 15 to let the computer know that there is an error.

SPP/EPP/ECP

The original specification for parallel ports was unidirectional, meaning that data only traveled in one direction for each pin. With the introduction of the PS/2 in 1987, IBM offered a new bidirectional parallel port design. This mode is commonly known as Standard Parallel Port (SPP) and has completely replaced the original design.

Enhanced Parallel Port (EPP) was created by Intel, Xircom and Zenith in 1991. EPP allows for much more data, 500 kilobytes to 2 megabytes, to be transferred each second. It was targeted specifically for non-printer devices that would attach to the parallel port, particularly storage devices that needed the highest possible transfer rate.

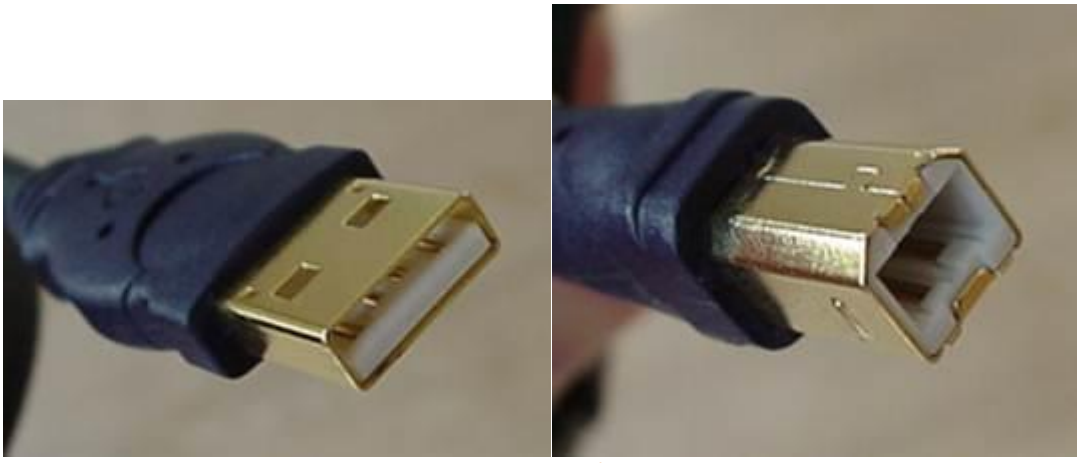
Close on the heels of the introduction of EPP, Microsoft and Hewlett Packard jointly announced a specification called Extended Capabilities Port (ECP) in 1992. While EPP was geared toward other devices, ECP was designed to provide improved speed and functionality for printers.

In 1994, the IEEE 1284 standard was released. It included the two specifications for parallel port devices, EPP and ECP. In order for them to work, both the operating system and the device must support the required specification. This is seldom a problem today since most computers support SPP, ECP and EPP and will detect which mode needs to be used, depending on the attached device. If you need to manually select a mode, you can do so through the BIOS on most computers.

2.7.3 USB Ports

Just about any computer that you buy today comes with one or more Universal Serial Bus connectors. These USB connectors let you attach mice, printers and other accessories to your computer quickly and easily. The operating system supports USB as well, so the installation of the device drivers is quick and easy, too. Compared to other ways of connecting devices to your computer (including parallel ports, serial ports and special cards that you install inside the computer's case), USB devices are incredibly simple.

The Universal Serial Bus gives you a single, standardized, easy-to-use way to connect up to 127 devices to a computer. Just about every peripheral made now comes in a USB version.



USB A connector and B connector

The USB standard uses "A" and "B" connectors to avoid confusion:

- "**A**" connectors head "upstream" toward the computer.
- "**B**" connectors head "downstream" and connect to individual devices.

Most computers that you buy today come with at least one or two USB sockets. But with so many USB devices on the market, you easily run out of sockets very quickly. For

example, you could have a keyboard, mouse, printer, microphone and webcam all running on USB technology. The easy solution to the problem is to buy a USB hub.

A USB four-port hub

The USB standard allows for devices to draw their power from their USB connection. A high-power device like a printer or scanner will have its own power supply, but low-power devices like mice and digital cameras get their power from the bus. The power comes from the computer.

USB Features

The Universal Serial Bus has the following features:

- The computer acts as the host.
- Up to 127 devices can connect to the host, either directly or by way of USB hubs.
- Individual USB cables can run as long as 5 meters; with hubs, devices can be up to 30 meters (six cables' worth) away from the host.
- With USB 2.0, the bus has a maximum data rate of 480 megabits per second (10 times the speed of USB 1.0).
- A USB 2.0 cable has two wires for power (+5 volts and ground) and a twisted pair of wires to carry the data. The USB 3.0 standard adds four more wires for data transmission. While USB 2.0 can only send data in one direction at a time (downstream or upstream), USB 3.0 can transmit data in both directions simultaneously.
- On the power wires, the computer can supply up to 500 milliamps of power at 5 volts. A USB 3.0 cable can supply up to 900 milliamps of power.
- Low-power devices (such as mice) can draw their power directly from the bus. High-power devices (such as printers) have their own power supplies and draw minimal power from the bus. Hubs can have their own power supplies to provide power to devices connected to the hub.

- USB devices are hot-swappable, meaning you can plug them into the bus and unplug them any time.
- A USB 3.0 cable is compatible with USB 2.0 ports - you won't get the same data transfer speed as with a USB 3.0 port but data and power will still transfer through the cable.

Many USB devices can be put to sleep by the host computer when the computer enters a power-saving mode.

2.7.4 Flash RAM

Flash memory is used for easy and fast information storage in computers, digital cameras and home video game consoles. It is used more like a hard drive than as RAM. Flash memory is known as a solid state storage device, meaning there are no moving parts and everything is electronic instead of mechanical.

Here are a few examples of flash memory:

- Your computer's BIOS chip
- CompactFlash (most often found in digital cameras)
- SmartMedia (most often found in digital cameras)
- Memory Stick (most often found in digital cameras)
- PCMCIA Type I and Type II memory cards (used as solid-state disks in laptops)
- Memory cards for video game consoles

Flash memory is a type of EEPROM chip.

There are a few reasons to use flash memory instead of a hard disk:

- It has no moving parts, so it's noiseless.
- It allows faster access.
- It's smaller in size and lighter.

So why don't we just use flash memory for everything? Because the cost per megabyte for a hard disk is drastically cheaper and the capacity is substantially more.