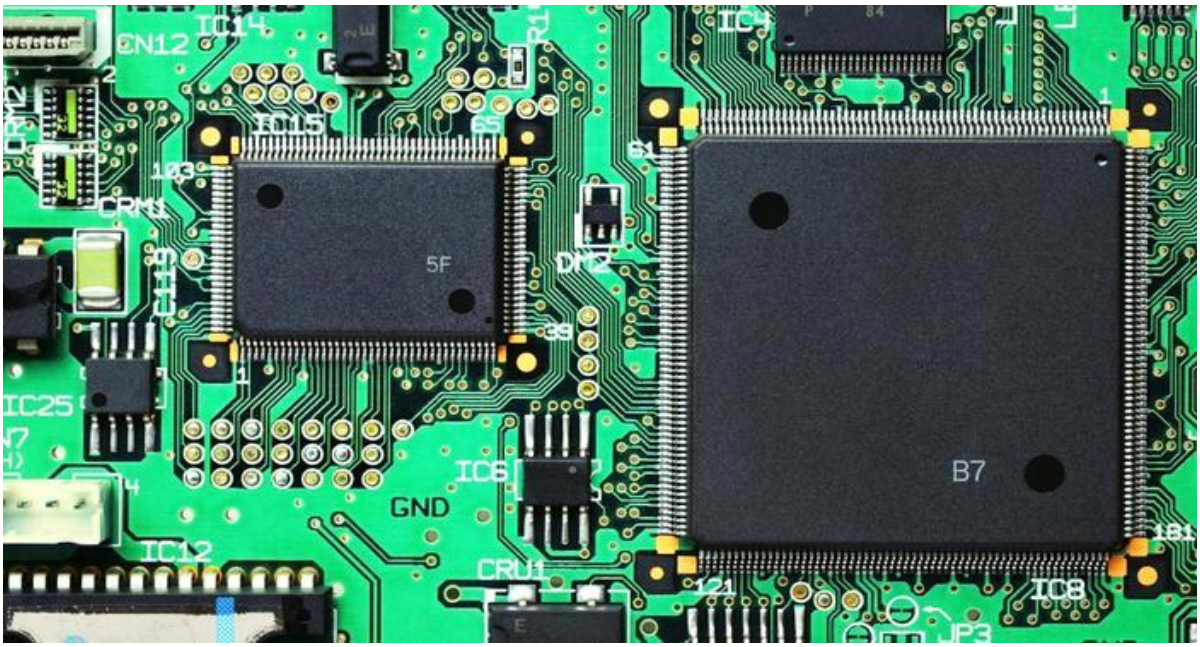
Processor



What is processor:

A processor is the logic circuitry that responds to and processes the basic instructions that drive a computer.

The term processor has generally replaced the term central processing unit (CPU). The processor in a personal computer or embedded in small devices is often called a microprocessor.

Processor is the heart of an embedded system. It is the basic unit that takes inputs and produces an output after processing the data. For an embedded system designer, it is necessary to have the knowledge of both microprocessors and microcontrollers.

A processor has two essential units −

Program Flow Control Unit (CU)

Execution Unit (EU)

The CU includes a fetch unit for fetching instructions from the memory. The EU has circuits that implement the instructions pertaining to data transfer operation and data conversion from one form to another.

The EU includes the Arithmetic and Logical Unit (ALU) and also the circuits that execute instructions for a program control task such as interrupt, or jump to another set of instructions.

A processor runs the cycles of fetch and executes the instructions in the same sequence as they are fetched from memory.

Types of Processors

Processors can be of the following categories −

General Purpose Processor (GPP)

Microprocessor

Microcontroller

Embedded Processor

Digital Signal Processor

Media Processor

Application Specific System Processor (ASSP)

Application Specific Instruction Processors (ASIPs)

GPP core(s) or ASIP core(s) on either an Application Specific Integrated Circuit (ASIC) or a Very Large Scale Integration (VLSI) circuit.

The Central Processing Unit (Normally called a processor or CPU) is the brain of the PC. It executes instructions, allowing a computer to perform all kinds of tasks. From burning CDs or DVDs to something as simple as a mouse click, the CPU is always at work. Processors consist of two parts: The Arithmetic Unit, which performs math and logical operations, & the Control Unit, which decodes instructions. Over the years, processors have become extremely fast. AMD and Intel are the two primary manufacturers.

CPU technology constantly changes, probably faster than any other type of hardware. On this page I highlight what I consider are the main specifications.

When looking at a CPU, you don't really see the processor itself. The little piece of silicon that contains the circuitry is very small. What you actually see is the package that it's in.

Processors are designed to fit into a certain type of socket on the motherboard.

History:

The advent of low-cost computers on integrated circuits has transformed modern society. General-purpose microprocessors in personal computers are used for computation, text editing, multimedia display, and communication over the Internet.

Many more microprocessors are part of embedded systems, providing digital control over myriad objects from appliances to automobiles to cellular phones and industrial process control.

The first use of the term "microprocessor" is attributed to Viatron Computer Systems describing the custom integrated circuit used in their System 21 small computer system announced in 1968.

Today, computers are a part of our lifestyle, but the first computer that was used was developed at the University of Pennsylvania in the year 1946! It had an ENIAC (Electronic Numerical Integrator And Computer) processor.

From the development of the first microprocessor - Intel's 4004 to the latest ones - the microprocessors have come a long way. Here, we look into the story so far.

Types of processors:

Processors can be identified by two main parameters: how wide they are and how fast they are. The speed of a processor is a fairly simple concept. Speed is counted in megahertz (MHz), which means millions of cycles per secondv

The width of a processor is a little more complicated to discuss because there are three main specifications in a processor that are expressed in width. They are

Internal registers

Data input and output bus

Memory address bus

Systems below 16MHz usually had no cache memory at all. Starting with 16MHz systems, high-speed cache memory appeared on the motherboard because the main memory at the time could not run at 16MHz. Prior to the 486 processor, the cache on the motherboard was the only cache used in the system.

Starting with the 486 series, processors began including what was called L1 (Level 1) cache directly on the processor die. This meant that the L1 cache always ran at the full speed of the chip, especially important when the later 486 chips began to run at speeds higher than the motherboards they were plugged into. During this time the cache on the motherboard was called the second level or L2 cache, which ran at the slower motherboard speed.

Starting with the Pentium Pro and Pentium II, Intel began including L2 cache memory chips directly within the same package as the main processor. Originally this built-in L2 cache was implemented as physically separate chips contained within the processor package but not a part of the processor die. Since the speed of commercially available cache memory chips could not keep pace with the main processor, most of the L2 cache in these processors ran at one-half speed (Pentium II/III and AMD Athlon), while some ran the cache even slower, at two-fifths or even one-third the processor speed (AMD Athlon).

The original Pentium II, III, Celeron, and Athlon (Model 1 and 2) processors use 512KB of either one-half, two-fifths, or one-third speed L2 cache

Table 3.1 L2 Cache Speeds

Processor	Speed	L2 Size	L2 Type	L2 Speed
Pentium III	450–600MHz	512KB	External	1/2 core (225–300MHz)
Athlon	550–700MHz	512KB	External	1/2 core (275–350MHz)
Athlon	750–850MHz	512KB	External	2/5 core (300–340MHz)
Athlon	900–1000MHz	512KB	External	1/3 core (300–333MHz)

The Pentium Pro, Pentium II/III Xeon, newer Pentium III, Celeron, K6-3, Athlon (Model 4), and Duron processors include full-core speed L2 as shown in Table

Table 3.2 Full-Core Speed Cache

Processor	Speed	L2 Size	L2 type	L2 Speed

Pentium Pro	150–200MHz	256KB–1MB	External	Full core
K6-3	350–450MHz	256KB	On-die	Full core
Duron	550–700+MHz	64KB	On-die	Full core
Celeron	300–600+MHz	128KB	On-die	Full core
Pentium II Xeon	400–450MHz	512KB–2MB	External	Full core
Athlon	650–1000+MHz	256KB	On-die	Full core
Pentium III	500–1000+MHz	256KB	On-die	Full core
Pentium III Xeon	500–1000+MHz	256KB–2MB	On-die	Full core

Evaluating CPU performance can be tricky. CPUs with different internal architectures do things differently and may be relatively faster at certain things and slower at others. To fairly compare different CPUs at different clock speeds, Intel has devised a specific series of benchmarks called the iCOMP (Intel Comparative Microprocessor Performance) index that can be run against processors to produce a relative gauge of performance. The iCOMP index benchmark has been updated twice and released in original iCOMP, iCOMP 2.0, and now iCOMP 3.0 versions.

Processor	iCOMP 2.0 Index	Processor	iCOMP 2.0 Index
Pentium 75	67	Pentium Pro 200	220

Pentium 100	90	Celeron 300	226
Pentium 120	100	Pentium II 233	267
Pentium 133	111	Celeron 300A	296
Pentium 150	114	Pentium II 266	303
Pentium 166	127	Celeron 333	318
Pentium 200	142	Pentium II 300	332
Pentium-MMX 166	160	Pentium II Overdrive 300	351
Pentium Pro 150	168	Pentium II 333	366
Pentium-MMX 200	182	Pentium II 350	386
Pentium Pro 180	197	Pentium II Overdrive 333	387
Pentium-MMX 233	203	Pentium II 400	440
Celeron 266	213	Pentium II 450	483

Table :Intel iCOMP 2.0 Index Ratings

Processor	iCOMP3.0 Index	Processor	iCOMP 3.0 Index
Pentium II 350	1000	Pentium III 650	2270
Pentium II 450	1240	Pentium III 700	2420
Pentium III 450	1500	Pentium III 750	2540

Pentium III 500	1650	Pentium III 800	2690
Pentium III 550	1780	Pentium III 866	2890
Pentium III 600	1930	Pentium III 1000	3280
Pentium III 600E	2110		

Table: Intel iComp 3.0 Ratings

Normally, you can set the motherboard speed and multiplier setting via jumpers or other configuration mechanism (such as BIOS setup) on the motherboard. Modern systems use a variable- frequency synthesizer circuit usually found in the main motherboard chipset to control the motherboard and CPU speed. Most Pentium motherboards will have three or four speed settings. The processors used today are available in a variety of versions that run at different frequencies based on a given motherboard speed. For example, most of the Pentium chips run at a speed that is some multiple of the true motherboard speed.

CPU Type	CPU Speed (MHz)	CPU Clock Multiplier	Motherboard Speed (MHz)
Pentium	60	1x	60
Pentium	66	1x	66
Pentium	75	1.5x	50
Pentium	90	1.5x	60
Pentium	100	1.5x	66
Pentium	120	2x	60
Pentium	133	2x	66
Pentium	150	2.5x	60
Pentium/Pentium Pro/MMX	166	2.5x	66

Pentium/Pentium Pro	180	3x	60
Pentium/Pentium Pro/MMX	200	3x	66
Pentium-MMX/Pentium II	233	3.5x	66
Pentium-MMX(Mobile)/ Pentium II/Celeron	266	4x	66
Pentium II/Celeron	300	4.5x	66
Pentium II/Celeron	333	5x	66
Pentium II/Celeron	366	5.5x	66
Celeron	400	6x	66
Celeron	433	6.5x	66
Celeron	466	7x	66
Celeron	500	7.5x	66
Celeron	533	8x	66
Celeron	566	8.5x	66
Celeron	600	9x	66
Celeron	633	9.5x	66
Celeron	667	10x	66
Pentium II	350	3.5x	100
Pentium II/Xeon	400	4x	100
Pentium II/III/Xeon	450	4.5x	100
Pentium III/Xeon	500	5x	100
Pentium III/Xeon	550	5.5x	100
Pentium III/Xeon	600	6x	100

Pentium III/Xeon	650	6.5x	100
Pentium III/Xeon	700	7x	100
Pentium III/Xeon	750	7.5x	100
Pentium III/Xeon	800	8x	100
Pentium III/Xeon	850	8.5x	100
Pentium III/Xeon	533	4x	133
Pentium III/Xeon	600	4.5x	133
Pentium III/Xeon	667	5x	133
Pentium III/Xeon	733	5.5x	133
Pentium III/Xeon	800	6x	133
Pentium III/Xeon	866	6.5x	133
Pentium III/Xeon	933	7x	133
Pentium III/Xeon	1000	7.5x	133
Pentium III/Xeon	1066	8x	133
Pentium III/Xeon	1133	8.5x	133
Pentium III/Xeon	1200	9x	133
Pentium III/Xeon	1266	9.5x	133
Pentium III/Xeon	1333	10x	133

Table: Intel Processor and Motherboard Speeds

Address Bus:

The address bus is the set of wires that carries the addressing information used to describe the memory location to which the data is being sent or from which the data is being retrieved. As with the data bus, each wire in an address bus carries a single bit of information. This single bit is a single digit in the address. The more wires (digits) used in calculating these addresses, the greater

the total number of address locations. The size (or width) of the address bus indicates the maximum amount of RAM that a chip can address.

Computers use the binary (base 2) numbering system, so a two-digit number provides only four unique addresses (00, 01, 10, and 11) calculated as 2^2. A three-digit number provides only eight addresses (000–111), which is 2^3. For example, the 8086 and 8088 processors use a 20-bit address bus that calculates as a maximum of 2^{20} or 1,048,576 bytes (1MB) of address locations. Table 3.10 describes the memory-addressing capabilities of processors.

Processor Family	Address Bus	Bytes	KB	MB	GB
8088/8086	20-bit	1,048,576	1,024	1	—
286/386SX	24-bit	16,777,216	16,384	16	—
386DX/486/P5 Class	32-bit	4,294,967,296	4,194,304	4,096	4
P6 Class	36-bit	68,719,476,736	67,108,864	65,536	64

Table: Processor Memory-Addressing Capabilities

Processor Modes:

All Intel 32-bit and later processors, from the 386 on up, can run in several modes. Processor modes refer to the various operating environments and affect the instructions and capabilities of the chip. The processor mode controls how the processor sees and manages the system memory and the tasks that use it.

Three different modes of operation possible are

Real mode (16-bit software)

Protected mode (32-bit software)

Virtual Real mode (16-bit programs within a 32-bit environment)

Real Mode:

The original IBM PC included an 8088 processor that could execute 16-bit instructions using 16-bit internal registers and could address only 1MB of memory using 20 address lines. All original PC software was created to work with this chip and was designed around the 16-bit instruction set and 1MB memory model. For example, DOS and all DOS software, Windows 1.x through 3.x.

Protected (32-bit) Mode:

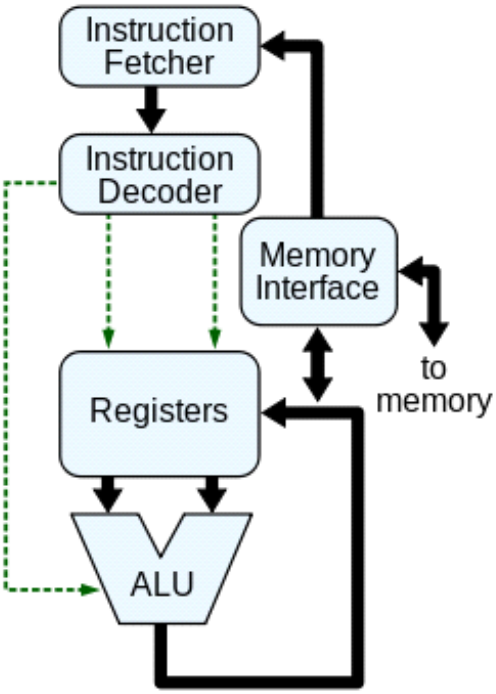
Then came the 386, which was the PC industry's first 32-bit processor. This chip could run an entirely new 32-bit instruction set. To take full advantage of the 32-bit instruction set, you needed a 32-bit operating system and a 32-bit application. This new 32-bit mode was referred to as protected mode, which alludes to the fact that software programs running in that mode are protected from overwriting one another in memory.

Virtual Real Mode:

The key to the backward compatibility of the Windows 32-bit environment is the third mode in the processor: virtual real mode. Virtual real is essentially a virtual real mode 16-bit environment that runs inside 32-bit protected mode. When you run a DOS prompt window inside Windows, you have created a virtual real mode session. Because protected mode allows true multitasking, you can actually have several real mode sessions running, each with its own software running on a virtual PC. This can all run simultaneously, even while other 32-bit applications are running.

Processors can be of size 4 bit,8 bit,16 bit,32 bit. Processors can be socket based or without socket. Generally we use socket based processors.

Components used in processors:



A typical CPU has a number of components. The first is the arithmetic logic unit (ALU), which performs simple arithmetic and logical operations. Second is the control unit (CU), which manages the various components of the computer. It reads and interprets instructions from memory and transforms them into a series of signals to activate other parts of the computer. The control unit calls upon the arithmetic logic unit to perform the necessary calculations.

Third is the cache, which serves as high-speed memory where instructions can be copied to and retrieved. Early CPUs consisted of many separate components, but since the 1970s, they have been constructed as a single integrated unit called a microprocessor. As such, a CPU is a specific type of microprocessor. The individual components of a CPU have become so integrated that you can't even recognize them from the outside. This CPU is about two inches by two inches in size.

The ALU is where the calculations occur, but how do these calculations actually get carried out? To a computer, the world consists of zeros and ones. Inside a processor, we can store zeros and ones using transistors. Transistors are located on a very thin slice of silicon. A single silicon chip can contain thousands of transistors.

Hardware components are often categorized as being either input, output, storage or processing components. Devices which are not an integral part of the CPU are referred to as being peripherals. Peripherals are usually used for either input, storage or output (such as a hard disk, keyboard or

printer). A device does not necessarily have to be outside the same physical box as the CPU. The best example of this is the hard disk, which is a peripheral even though it is not usually housed within the main case.

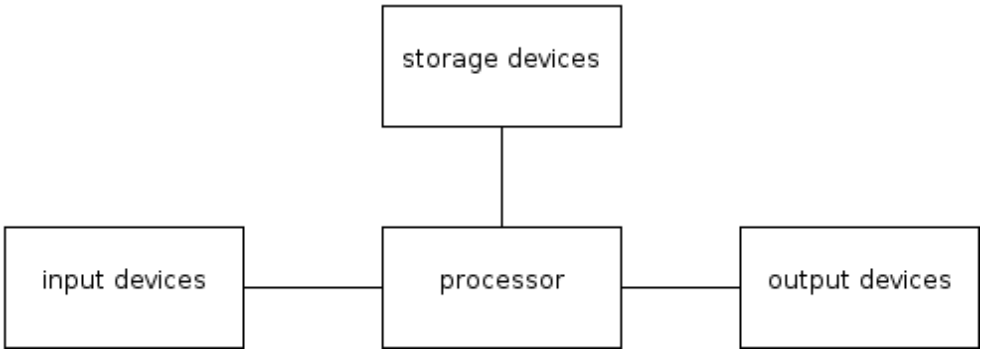


Fig 1: The main hardware components of a computer system

Input devices are hardware devices which take information from the user of the computer system,

Factors Affecting CPU Performance:

There are several factors that affect processor performance. Understanding these factors will help you make the proper choices when designing your homebuilt computer.

The most important factors affecting processor performance are:

Instruction Set

This is the processor's built-in code that tells it how to execute its duties.

You really have no control over the instruction set. It is built-in to the CPU and is not something you can change or update. But together with processor architecture, it does affect performance across a given line of CPUs. The processor's architecture determines how many cycles, or ticks, are needed to execute a given instruction.

In other words, some instruction sets are more efficient than others, enabling the processor to do more useful work at a given speed. This is one reason why when choosing a processor, benchmark tests that measure the chips' abilities to do actual work can be very useful.

Clock Speed:

The clock speed (or clock rate) is stated in megahertz (MHz) or gigahertz (GHz), and refers to the speed at which the processor can execute instructions. The faster the clock, the more instructions the processor can complete per second.

All else being equal, processors with faster clock speeds process data faster than those with slower clock speeds. It's also the first number you will see in advertisements for CPUs. But as mentioned previously, the efficiency of the processor's architecture determines how much actual work a processor can do with the same number of cycles.

So don't select a CPU based on clock speed alone. It's only one of the factors (albeit an important one) that determines how well a CPU will perform in real-world situations. Again, benchmarking tests are your friend.

Bandwidth

Measured in bits, the bandwidth determines how much information the processor can process in one instruction. If you were to compare data flow to the flow of traffic on a highway, then clock speed would be the speed limit, and bandwidth would be the number of lanes on the highway.

The current bandwidth standard for desktop and laptop PCs is 64 bit. 32-bit is officially a thing of the past.

Front Side Bus (FSB) Speed

The FSB is the interface between the processor and the system memory. As such, the FSB speed limits the rate at which data can get to the CPU, which in turn limits the rate at which the CPU can process that data. The CPU's FSB speed determines the maximum speed at which it can transfer data to the rest of the system.

Other factors affecting data transfer rates include the system clock speed, the motherboard chipset, and the RAM speed.

On-Board Level-2 (L2) Cache

The on-board (or "on-die") cache is a little bit of high-performance RAM built directly into the processor. It enables the CPU to access repeatedly used data directly from its own on-board memory, rather than repeatedly requesting it from the system RAM.

L2 Cache is very critical to applications such as games, video editing, and 3-D applications such as CAD/CAM programs. It's less important for activities such as web surfing, email, and word processing.

Some low-cost CPU's have as little as 128K of L2 cache. Higher-end CPU's have up to 4 MB.

Heat and Heat Dissipation

When processors run too hot, they can start doing funky things like cause errors, lock, freeze, or even burn up. Installing an inadequate cooling system can cause your homebuilt computer project to go sour in a big (and possibly expensive) way. So don't skimp on the cooling.

Processor frequency:

Microprocessor frequency specifies the operating (internal) frequency of CPU's core. The higher the frequency is for a given CPU family, the faster the processor is. Processor frequency is not the

only parameter that affects system performance. Another parameter than greatly affects the performance is CPU efficiency, that is how many Instructions Per Clock (IPC) the CPU can process. Knowing these two parameters it's easy to calculate total number of instructions per second that can be processed by CPU: Frequency * IPC. All modern AMD x86 microprocessors and all Intel microprocessors based P6, mobile and Core micro-architectures tried to improve their performance by improving the IPC, and, whenever possible, by increasing processor frequency. Intel Netburst microarchitecture used quite different approach - it tried to increase processor frequency at the expense of IPC. This didn't work well for this micro-architecture.

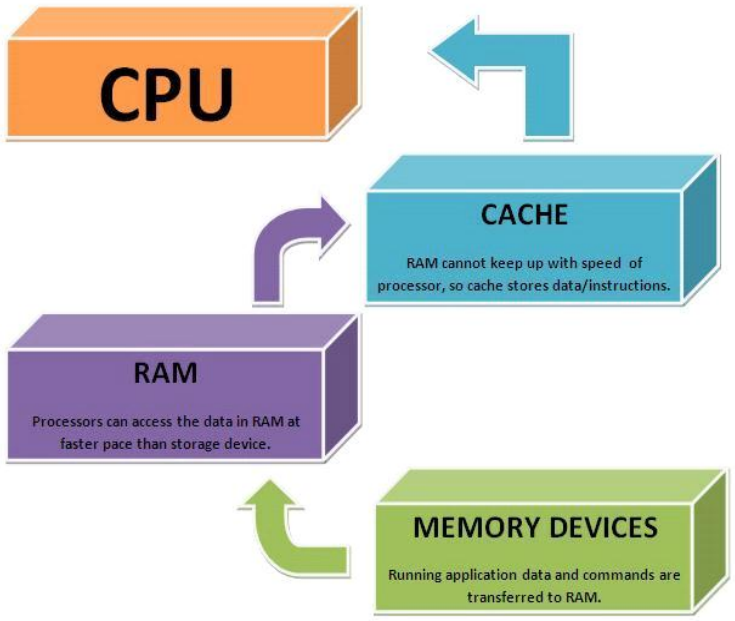
Modern microprocessors do not always operate at the same frequency. To save power, all processors with Power Now! or SpeedStep technology may temporarily reduce their operating frequency. Some mobile Intel Core 2 Duo processors may temporarily increase frequency of one of their cores when another core is idle.

The CPU frequency is measured in Hertz. The frequency can also be expressed in:

- Kilohertz, or kHz, equals to 1,000 Herts
- Megahertz, or MHz, equals to 1,000,000 Herts or 1,000 kHz
- Gigahertz, or GHz, equals to 1,000,000,000 Herts, or 1,000,000 kHz, or 1,000 MHz.

First microprocessors ran at frequencies close to 1 MHz. Modern microprocessors run at frequencies exceeding 3 GHz, or 3,000,000,000 Hertz.

Cache memory:



Cache memory is a small-sized type of volatile computer memory that provides high-speed data access to a processor and stores frequently used computer programs, applications and data. It stores and retains data only until a computer is powered up.

Cache memory is the fastest memory in a computer. It is typically integrated on the motherboard and directly embedded on the processor or main random access memory (RAM).

Cache memory provides faster data storage and access by storing an instance of programs and data routinely accessed by the processor. Thus, when a processor requests data that already has an instance in the cache memory, it does not need to go to the main memory or the hard disk to fetch the data.

Cache memory can be primary or secondary cache memory, where primary cache memory is directly integrated or closest to the processor. In addition to hardware-based cache, cache memory also can be a disk cache, where a reserved portion on a disk stores and provide access to frequently accessed data/applications from the disk.

Caches are a critical part of the memory hierarchy in any modern computer architecture. The two previous answers hit on this. A famous paper (*Cache Memories* by Alan Jay Smith) tells us that the following things are typically optimized in cache design. There are more constraints now than in the 1980s, namely power consumption, but the four principles here are still important.

Increase hit rate:
Hit rate is the rate at which memory references are found in the cache memory. We want to make use the performance benefit that we get out of caching in the first place and that is most easily achieved by maximizing this.

Decrease access time:
Accessing a particular word in a cache should take as short as possible. We can decrease access time in a number of ways. We can decrease the latency of caches by decreasing their size or their associativity. Way prediction can also be employed in an n-way set associative cache but may not need to be depending on its design. We also can increase their bandwidth, or the number of outstanding memory references that can be handled at any point in time.

Decrease miss penalty:
Misses are inevitable. However if we can decrease the amount of time that it takes to handle a miss, we get better processor performance. The miss penalty can be decreased by increasing the hit rate and also by applying different optimizations like the critical word first and early restart cache optimizations.

Some queries regarding cache:

Q.1 Where all do you want caches i.e. at what all level?
So it can be L1$, L2$, L3$ etc. Then along with this you need to decide which all will be shared and among which processor. So LLC acn be shared among all CPU cores, can be shared among CPU and GPU sitting on same die area.

Q.2 What will be cache design parameters?
So if you are talking about L1 cache, then they need to fast, so they should be direct mapped or very low set associative. As you move to lower level caches then they are around 16 way set associative. Along with this the size of L1 is small because then decoding time i.e. searching time will be drastically reduced.

Q.3 What will be cache replacement policy?
This was not asked but this remains an important parameter. Generally if you are not doing R&D and production based then it is very unlikely that you will change from you previous implementations.

Q.4 Caches will be inclusive or exclusive?
This is a very important design choice because many other design choices will be highly dependent on this. Your cache coherence protocol highly depends upon this.

Q.5 Cache will be write through or write back?
This seems to be pretty straightforward. You will chose write back as one will be ready to add more control logic rather than having very high level of traffic on shared bus.

Q.6 How much non-blocking capability will it have?
Modern day caches are non blocking. You need to find the sweet spot that how many miss under misses should it accommodate.

Q.7 What will be cache coherence protocol?
In present CMPs it becomes very important to have an super awesome and efficient protocol which is effective and is not bandwidth hungry. I think Intel uses MESIF protocol.