Introduction to Computer Terminology

In this section I will define terminology which will be used throughout the book. It is very difficult to define technical terms in computer science without referencing other terms which must also be defined. For example, in defining the Central Processing Unit (CPU), we might introduce new terms like Machine Code, Caches, Arithmetic Logical Units (ALU) and Registers. If these terms are not defined, then the attempt to define the CPU falls short. Sometimes it will be necessary to use a term first and then define it later. Hopefully by the time we get to the end of this chapter the majority of key terms will be defined.

I need to make assumptions about the background of my target audience. I assume that most readers understand the C++ language and the concept of Machine Code. I also assume a knowledge of the internals of various CPUs. As we get further into the book it will get quite technical and detailed.

I cannot personally teach all the background needed to understand and evaluate the RISC++ Architecture while assuming the reader is an absolute beginner. But I would like to make this book to be accessible to those beginners who understand the basics of programming, as well as experts in CPU design. I will do the best job I can to explain things simply in the text and then provide links for those who want to study further. Wikipedia is a major source of information through out this document along with other sites.

For terminology, I recommend the website TechTerms.com. Some of the links below will take you to the full definition in TechTerms. I have also included "cut and paste" selections of some online definitions, and they are indented as block quotes. I will also include my own comments which will not be indented as a block quote.

Computer Architecture

The term "architecture" refers to a wide variety of industry standards which allow manufacturers of computer components to produce devices that work together. This includes both hardware and software standards. The ubiquitous USB ports are an example of an industry standard "architecture". The HDMI cable that connects to your TV or monitor is also an industry standard architecture. If manufacturers of USB and HDMI cables and ports did not follow very exact rules, the products would not be able to work in such a wide variety of devices.

Even though computers are much smaller and faster than the old mainframes I worked on at IBM, much of the basic design, the "architecture", has remained the same. Without the rigor of clearly defined specifications (and people like me who tested them), there could not have been an entire industry, with thousands of manufacturing companies and millions of engineers and programmers, all working from the same blueprints to build and program computers and their interconnected devices.

There are many kinds of "architectures" in computer science. As with builders of physical buildings, Computer Scientists must rely on "blueprints" which allow various hardware and software products to work together.

This link provides a partial list of various industry standards for computer hardware and software:

https://en.wikipedia.org/wiki/List_of_computer_standards

I will be discussing two kinds of architectures extensively in this book:

1). Instruction Set Architecture (ISA)

Another name for the ISA is "Machine Code".

The Machine Code is an industry standardized binary programming language which tells the CPU what to do.

The most common Machine Code in the world today is called the "X86". While it was designed by the Intel corporation, many other manufacturers have produced "X86 compatible" CPUs over the years.

> Quoted from: https://en.wikipedia.org/wiki/X86

> At various times, companies such as IBM, VIA, NEC, AMD, TI, STM, Fujitsu, OKI, Siemens, Cyrix, Intersil, C&T, NexGen, UMC, and DM&P started to design or manufacture x86 processors (CPUs) intended for personal computers and embedded systems. Other companies that designed or manufactured x86 or x87 processors include ITT Corporation, National Semiconductor, ULSI System Technology, and Weitek.

> Such x86 implementations were seldom simple copies but often employed different internal micro architectures and different solutions at the electronic and physical levels. Quite naturally, early compatible microprocessors were 16-bit, while 32-bit designs were developed much later. For the personal computer market, real quantities started to appear around 1990 with i386 and i486 compatible processors, often named similarly to Intel's original chips.

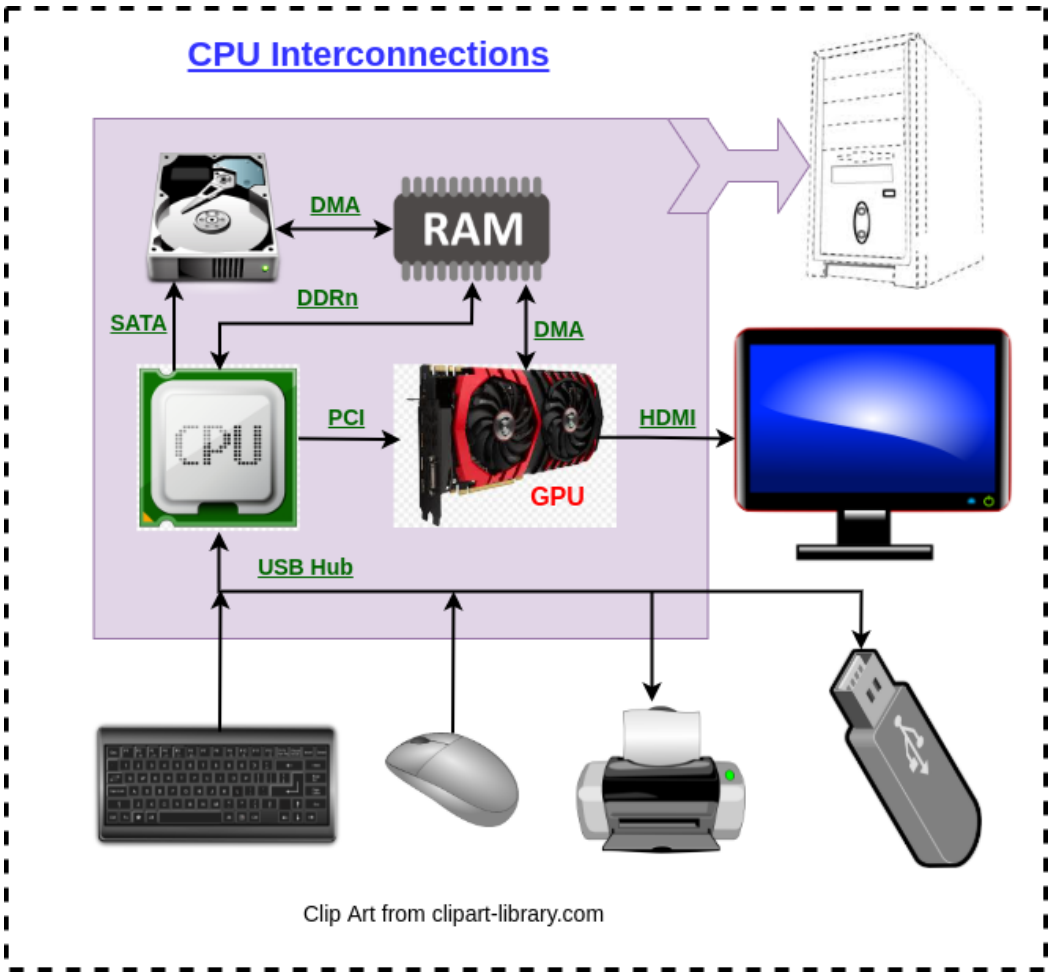
The X86 Instruction Set Architecture has provided a way for programs that were written decades ago to still run on the most advanced computer systems. But the internal designs of CPUs have changed dramatically while still maintaining compatibility with old software preventing obsolescence.

2). Micro-architecture

The second kind of architecture I will be discussing in detail in the book is the "micro-architecture". This is the internal hardware inside a CPU chip. When discussing the Machine Code (ISA) it is very important to also discuss the micro-architecture as well. The Machine Code tells the micro-architecture what to do. Over the years the Intel Corporation has designed many different micro-architectures, for which the X86 Machine Code is the binary programming language. We will examine some of these in detail.

Components of a Computer System

Sometimes people erroneously refer to the familiar "tower" as the CPU. This is not accurate. The CPU is a computer chip inside the tower. This chip is where the operating system and the various apps run. Inside the "tower" the CPU connects to Random Access Memory (RAM), to one or more Hard Drives, DVD drives, USB hubs, Ethernet ports, and a Graphics Processor Unit (GPU). The GPU connects to the Display through an HDMI port. Connected to the USB ports are the Mouse, Keyboard, Printer, and Flash Drive Memory.



Motherboard

> Quoted from: https://techterms.com/definition/motherboard
>
> The motherboard is the main circuit board of your computer and is also known as the mainboard or logic board. If you ever open your computer, the biggest piece of silicon you see is the motherboard. Attached to the motherboard, you'll find the CPU, ROM, memory RAM expansion slots, PCI slots, and USB ports. It also includes controllers for devices like the hard drive, DVD drive, keyboard, and mouse. Basically, the motherboard is what makes everything in your computer work together.

In the illustration above, a facsimile of the motherboard is shown in purple.

Central Processing Unit (CPU)

> Quoted from: https://techterms.com/definition/cpu
>
> Stands for "Central Processing Unit." The CPU is the primary component of a computer that processes instructions. It runs the operating system and applications, constantly receiving input from the user or active software programs. It processes the data and produces output, which may be stored by an application or displayed on the screen.
>
> While processor architectures differ between models, each processor within a CPU typically has its own ALU, FPU, register, and L1 cache. In some cases, individual processing cores may have their own L2 cache, though they can also share the same L2 cache. A single frontside bus routes data between the CPU and the system memory.

Almost this whole book will be about the CPU, so we will delve into the details later.

Random Access Memory (RAM)

> Quoted from: https://techterms.com/definition/ram

> Stands for "Random Access Memory" and is pronounced "ram." RAM is a common hardware component found in electronic devices, including desktop computers, laptops, tablets, and smartphones. In computers, RAM may be installed as memory modules, such as DIMMs or (SO-DIMMs sodimm). In tablets and smartphones, RAM is typically integrated into the device and cannot be removed.

> The amount of RAM in a device determines how much memory the operating system and open applications can use. When a device has sufficient RAM, several programs can run simultaneously without any slowdown. When a device uses close to 100% of the available RAM, memory must be swapped between applications, which may cause a noticeable slowdown. Therefore, adding RAM or buying a device with more RAM is one of the best ways to improve performance.

> "RAM" and "memory" may be used interchangeably. For example, a computer with 16 GB of RAM has 16 gigabytes of memory. This is different than storage capacity, which refers to how much disk space the device's HDD or SSD provides for storing files.

> System memory is considered "volatile" memory since it only stores data while a device is turned on. When the device is powered down, data stored in the RAM is erased. When the device is restarted, the operating system and applications load fresh data into the system memory. This is why restarting a computer often fixes problems.

There are many kinds of RAM both inside and outside the CPU chip. Inside the chip are "registers", and various kinds of memory "caches". There are also many kinds of RAM external to the chip.

Double Data Rate (DDR) Access to External RAM

> Quoted from: https://techterms.com/definition/ddr

> Stands for "Double Data Rate." It is an advanced version of SDRAM, a type of computer memory. DDR-SDRAM, sometimes called "SDRAM II," can transfer data twice as fast as regular SDRAM chips. This is because DDR memory can send and receive signals twice per clock cycle. The efficient operation of DDR-SDRAM makes the memory great for notebook computers since it uses up less power.

There are several enhancements to DDR, each of which provides faster access to RAM:

> DDR2 https://techterms.com/definition/ddr2

> DDR3 https://techterms.com/definition/ddr3

> DDR4 https://techterms.com/definition/ddr4

This is why I label the access from the CPU to RAM as "DDRn".

Serial Advanced Technology Attachment (SATA)

> Quoted from: https://techterms.com/definition/sata

> Stands for "Serial Advanced Technology Attachment," or "Serial ATA." It is an interface used to connect ATA hard drives to a computer's motherboard. SATA transfer rates start at 150MBps, which is significantly faster than even the fastest 100MBps ATA/100 drives. For this and other reasons, Serial ATA is likely to replace the previous standard, Parallel ATA (PATA), which has been around since the 1980s.

The SATA interface is widely used in towers to communicate with many different Hard Drives of different physical sizes and storage capacity made by many different manufacturing companies. This is what standardized "architectures" are all about.

Peripheral Component Interconnect (PCI)

> Quoted from: https://techterms.com/definition/pci

> Stands for "Peripheral Component Interconnect." PCI is a hardware bus used for adding internal components to a desktop computer. For example, a PCI card can be inserted into a PCI slot on a motherboard, providing additional I/O ports on the back of a computer.

> The PCI architecture, also known as "conventional PCI," was designed by Intel and introduced in 1992. Many desktop PCs from the early 1990s to the mid 2000s had room for two to five PCI cards. Each card required an open slot on the motherboard and a removable panel on the back of the system unit. Adding PCI cards was an easy way to upgrade a computer, since you could add a better video card, faster wired or wireless networking, or add new ports, like USB 2.0.

If you click on the link above you can see a picture of a PCI card as well as the slots on the motherboard. PCI cards used to be the main way of providing interfaces to devices outside the tower. External devices such as printers, displays, external hard drives, speakers, microphones, etc. were all connected via PCI cards. PCI cards also used to provide Ethernet, WiFi, and USB ports.

Today many of the external interfaces are part of the mother board. Even GPUs are incorporated onto the same chip as the CPU in many phones and laptops. This is why laptops can have USB, HDMI and Ethernet ports without needing PCI cards.

The miracle of miniaturization has allowed devices such as phones and tablets to have large RAMs, high resolution displays, cameras, microphones, speakers, WiFi transmitters, solid state storage, etc. all inside the phone with the only connection to the outside is a USB port which doubles as a power source!

PCI cards are still found in computer "towers" but not in Laptops, phones, or "All-In-One" desktop computers. The main use for PCI slots in larger gaming computers today is to provide a high speed link to high resolution Graphic Processor units (GPUs). The GPUs have large amounts of high speed RAM. To transfer data between the GPU card and the CPU's external RAM, "Direct Memory Access DMA" is used.

Direct Memory Access (DMA)

> Quoted from: https://techterms.com/definition/dma

> Stands for "Direct Memory Access." DMA is a method of transferring data from the computer's RAM to another part of the computer without processing it using the CPU. While most data that is input or output from your computer is processed by the CPU, some data does not require processing, or can be processed by another device. In these situations, DMA can save processing time and is a more efficient way to move data from the computer's memory to other devices.

> In order for devices to use direct memory access, they must be assigned to a DMA channel. Each type of port on a computer has a set of DMA channels that can be assigned to each connected device. For example, a PCI controller and a hard drive controller each have their own set of DMA channels.

For example: if the Operating System or an App needs move data between the Hard Drive and external RAM, the Machine Code tells the CPU to start the data transfer. Then a large amount of data (a minimum of several kilo bytes) is transferred directly from the Hard Drive to or from memory while the CPU continues other tasks. The CPU meanwhile has access the RAM through the DDRn at the same time. When the transfer is complete the DMA hardware send a signal called an "interrupt" back to the CPU to tell it that the task is complete.

Similarly, if a game programmer wants to use the GPU to provide high resolution animation, the program sends a signal to move a pre-programmed "game world" from the Hard Drive into RAM using DMA. Then the "world" is transferred from the External RAM via the DMA into the high speed Graphics RAM (GRAM) on the GPU card. The programmer then uses a programming language designed for GPUs called CUDA to calculate high speed changes to the game world and produce high resolution animations, with light and shadows, smooth movements etc. All this is done without the CPU needing to be involved.

During game play there is constant interaction between the CPU and GPU over the PCI interface. The CPU handles all the mouse and keyboard input (or joysticks, pistols, Wii devices, steering wheels etc.) very quickly while also processing the logic that governs how the game proceeds. This is why very fast CPUs are needed for games as well as fast GPUs.

Meanwhile there are large amounts of data being transferred from the Hard Disk to RAM, and between RAM and the GPU RAM, both over their own DMA channels. There are calculations of how the animations take place on the Graphics Card. Meanwhile, data is sent to Display through the HDMI interface producing high resolution animations at many frames per second.

So next time you get an adrenaline rush and a dopamine fix as you defeat Darth Vader in a light saber battle; give a word of thanks to the thousands of computer architects, engineers, programmers and manufacturers who made it all possible. While you are at it, next time you complain about government regulations, understand that without strictly enforced industry standards, none of this would be possible.

End soap box rave!

Graphics Processing Unit (GPU)

> Quoted from: https://techterms.com/definition/gpu

> Stands for "Graphics Processing Unit." A GPU is a processor designed to handle graphics operations. This includes both 2D and 3D calculations, though GPUs primarily excel at rendering 3D graphics.

> The primary purpose of a GPU is to render 3D graphics, which are comprised of polygons. Since most polygonal transformations involve decimal numbers, GPUs are designed to perform floating point operations (as opposed to integer calculations). This specialized design enables GPUs to render graphics more efficiently than even the fastest CPUs. Offloading graphics processing to high-powered GPUs is what makes modern gaming possible.

> While GPUs excel at rendering graphics, the raw power of a GPU can also be used for other purposes. Many operating systems and software programs now support GPGPU, or general-purpose computation on graphics processing units. Technologies like OpenCL and CUDA allow developers to utilize the GPU to assist the CPU in non-graphics computations. This can improve the overall performance of a computer or other electronic device.

Because GPUs are used for very high speed "floating point" arithmetic on very large data sets, they are used extensively for various scientific and Artificial Intelligence (AI) applications. CPUs and GPUs each have their own advantages and disadvantages. The CPU is where the main program runs while it sends tasks to the GPU. The CPU excels at processing complex programs with many "If Then Else" clauses and operations on scalar data. The GPU excels in performing the same operations in parallel on huge 1, 2, or 3 dimension data sets.

The following Wikipedia link has a comprehensive overview of GPGPU:

https://en.wikipedia.org/wiki/General-purpose_computing_on_graphics_processing_units

For a comprehensive list of GPU Accelerated Applications see:

https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/gpu-applications-catalog.pdf

The RISC++ Architecture is a hybrid which incorporates features of both CPUs and GPUs. This is not intended to replace either. Instead the RISC++ is a CPU which can process relatively small arrays of data without needing to use some of the parallel processing techniques that are required for GPU programming. More on that later.

The GPU connects to a display through an HDMI interface.

High-Definition Multimedia Interface (HDMI)

> Quote from: https://techterms.com/definition/hdmi

> Stands for "High-Definition Multimedia Interface." HDMI is a trademark and brand name for a digital interface used to transmit audio and video data in a single cable. It is supported by modern audio/video equipment, such as 4K televisions, HDTVs, audio receivers, DVD and Blu-ray players, cable boxes, and video game consoles.

> While other types of A/V connections require separate cables for audio and video data, a single HDMI cable carries the audio and video streams together, eliminating cable clutter. For example, an analog component cable connection requires three cables for video and two for audio, totaling five cables in all. The same information can be transmitted digitally using one HDMI cable.

Universal Serial Bus (USB)

> Quote from: https://techterms.com/definition/usb

> Stands for "Universal Serial Bus." USB is the most common type of port found on modern computers. It is used to connect various peripherals, such as keyboards, mice, game controllers, printers, scanners, and external storage devices.

> USB provides both data transmission and low voltage (5V) power over a single cable. Devices that require five volts or less can operate over USB without an external power source. A single USB bus supports up to 127 devices, using multiple USB hubs.

The link above shows all the different types of USB connectors and their speeds.

Ethernet

> Quote from: https://techterms.com/definition/ethernet

> Ethernet, pronounced "E-thernet" (with a long "e"), is the standard way to connect computers on a network over a wired connection. It provides a simple interface and for connecting multiple devices, such computers, routers, and switches. With a single router and a few Ethernet cables, you can create a LAN, which allows all connected devices to communicate with each other.

> A standard Ethernet cable is slightly thicker than a phone cable and has an RJ45 connector on each end. Ethernet ports look similar to telephone jacks, but are slightly wider. You can plug or unplug devices on an Ethernet network while they are powered on without harming them.

> While Ethernet is still the standard for wired networking, it has been replaced in many areas by wireless networks. Wi-Fi allows you to connect your laptop or smartphone to a network without being tethered to the wall by a cable. The 802.11ac Wi-Fi standard even provides faster maximum data transfer rates than Gigabit Ethernet. Still, wired connections are less prone to interference and are more secure than wireless ones, which is why many businesses and organizations still use Ethernet.

The Ethernet cable is used in home computer configurations to connect to a cable modem and/or WiFii router.

In large data centers there are thousands of Ethernet cables interconnecting computers in a "server farm".

Machine Code

The "Machine Code" is a set of binary ones and zeros which have a specific meaning to the CPU. The Machine Code is the software program that tells the CPU what to do.

RISC vs CISC

Whenever Machine Code (or Instruction Set Architectures), the terms RISC and CISC come up often, so we will define them now. Back in the 60's and 70's RAM was very expensive. In order to use as little RAM as possible to store machine code instructions, each instruction was designed to perform many functions. The IBM 360 and 370 mainframes which I worked on had a machine code which was translated into "micro-code" and the micro-code ran on the hardware. But because each instruction needed to perform many complex functions it put unnecessary complexity into the hardware.

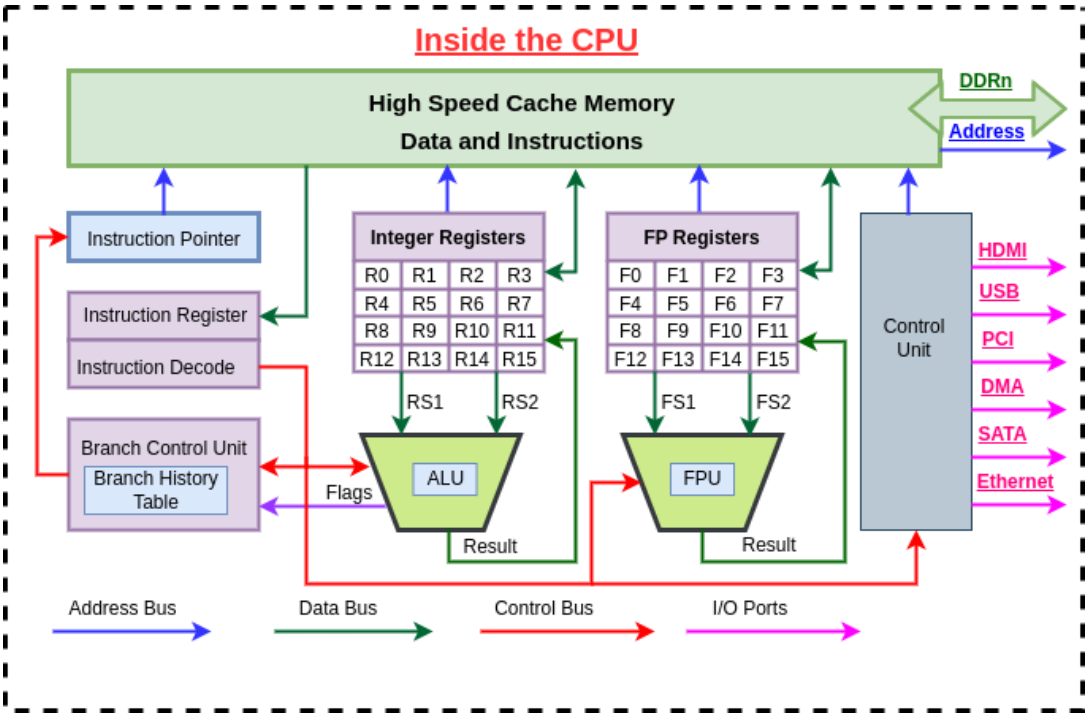
The 801 RISC was invented by the research division at IBM to produce a simpler CPU that could take advantage of the fact that that the price of RAM was falling. The idea of a RISC computer is that more of the complex function is performed by software freeing up the hardware to be more efficient and streamlined. The trade-off is that programs take up more memory.

The table below compares RISC vs CISC computers:

RISC vs CISC	
RISC	**CISC**
Reduced Instruction Set Computer	**Complex Instruction Set Computer**
Fixed length instructions (usually 32 bits)	Variable length instructions.
Programs use more RAM	Programs use RAM more efficiently.
Single clock cycle per instruction	Many clock cycles per instruction
Most instructions are register to register.	Many memory to memory instructions
Small number of instructions.	Large number of instructions
Few addressing modes	Many addressing modes
Software performs complex operations	Complex operations performed by hardware via micro-code.
More registers	Fewer registers
Instructions run directly on hardware	Instructions translated into micro-code
Pipelining is easy	Pipelining is difficult

Inside the CPU

The diagram below shows a some of the key components of a typical CPU:



Cache Memory

> Quoted from: https://en.wikipedia.org/wiki/CPU_cache
>
> A CPU cache is a hardware cache used by the central processing unit (CPU) of a computer to reduce the average cost (time or energy) to access data from the main memory. A cache is a smaller, faster memory, located closer to a processor core, which stores copies of the data from frequently used main memory locations.

For the moment, as we refer to the illustration above, we will think of the "cache memory" as a single unit which holds instructions and data which can be accessed very quickly. This high speed memory sends and receives data to and from external RAM through the DDRn interface we discussed earlier.

Modern CPUs have a hierarchy of caches named L1, L2 and L3 which will be discussed shortly.

Instruction Pointer

The instruction pointer tells the CPU where to find the next machine code instruction to execute. For the moment, we will assume that a single instruction is sent from the unified cache to the instruction decoder at a time. In reality, modern CPUs read many instructions at a time and decode them in parallel.

The instruction pointer accesses the next instruction in sequence unless a "Branch" or "Jump" instruction directs instruction flow to another part of the cache. If the desired instruction is not in cache, an "instruction cache miss" occurs. At this time the CPU "stalls" while it waits for the cache to be replenished from RAM over the DDRn interface. Not to worry though, if there are multiple cores, or multi-threading, program execution continues on the other cores or threads.

Instruction Decoder

The once the instruction is fetched from RAM or Cache it is sent to the decoder. The decoder is where the Machine Code is translated into control signals to the various parts of the CPU. Let us assume that the CPU in the diagram is a RISC CPU with fixed length instructions of 32 bits.

Let us further assume that arithmetic/logical (ALU) instructions are broken into the following fields:

- 8 bit Operation Code: This allows 256 unique instructions.
- 4 bit Destination Register: This selects one of 16 registers to hold the result of the operation.
- 4 bit Source Register 1: one of 16.
- 4 bit Source Register 2: one of 16.
- 12 bits not used with ALU instructions.

Integer Data Registers

Floating Point Data Registers

Arithmetic Logical Unit

ARM vs X86

More about Cache Memory

The Wikipedia article linked about is quite comprehensive and provides more links for further study. We will be getting more specific about caches as we discuss various micro-architectures and the RISC++ Architecture in future chapters. As mentioned above most modern CPUs have a hierarchy of caches. In L1 Caches are about 64 Kiblobytes, L2 are 1 or 2 Megabytes and External RAM is about 8 or 16 Gigabytes. I'm sure that as time goes on those numbers will continue to increase.

As the number of transistors that can be put on a chip increases exponentially, the data can be moved from one part of the chip to another exponentially faster. For this reason there needs to be very fast memory on the chip which can be accessed in a few clock cycles. But to build this memory requires many transistors and data paths to be dedicated to this task. This makes high speed on-chip memory very expensive both in terms of chip "real estate" and the financial cost of manufacturing. These cost must be passed on to the consumer. The faster the memory, the smaller amount of data capacity.

The basic operation of a memory cache is that it keeps copies of the most recently used data. When the Cache is full it sends the least recently used data back to the next level of Cache in the hierarchy and invalidates the "cache line". If the CPU needs data that is not in cache, a "cache miss" occurs. When this happens the CPU stalls until the data can be fetched. If the data is not in either the Level 1 or Level 2 cache then it must be fetched from external RAM. The CPU has to wait until the data is available. This can be a very long time in computer terms. One of the things that mitigates this is multi-core, and multi-threaded CPUs. More on that in future chapters.

Computer Programming Overview

High Level Languages

> Quoted from: https://techterms.com/definition/high-level_language

> A high-level language is a programming language designed to simplify computer programming. It is "high-level" since it is several steps removed from the actual code run on a computer's processor. High-level source code contains easy-to-read syntax that is later converted into a low-level language, which can be recognized and run by a specific CPU.

Examples of High Level Languages include:

- C++
- C#
- Cobol
- Fortran
- Java
  JavaScript

- Objective C
- Pascal
- Perl
- PHP
- Python
- Swift

Each of these languages use different syntax. Some are designed for writing desktop software programs, while others are best-suited for web development. But they all are considered high-level since they must be processed by a compiler or interpreter before the code is executed.

Source code written in languages like C++ and C# must be compiled into machine code in order to run. The compilation process converts the human-readble syntax of the high-level language into low-level code for a specific processor. Source code written in scripting languages like Perl and PHP can be run through an interpreter, which converts the high-level code into a low-level language on-the-fly.

Assembler Languages

Machine Code Languages