Virtual Memory

Virtual memory is a valuable concept in computer architecture that allows to run large, sophisticated programs on a computer even if it has a relatively small amount of RAM. A computer with virtual memory artfully manipulates the conflicting demands of multiple programs within a fixed amount of physical memory. A PC that's low on memory can run the same programs as one with abundant RAM, although more slowly.

A computer accesses the contents of its RAM through a system of addresses, which are essentially numbers that locate each byte. Because the amount of memory varies from PC to PC, determining which software will work on a given computer becomes complicated. Virtual memory solves this problem by treating each computer as if it has a large amount of RAM and each program as if it uses the PC exclusively. The operating system, such as Microsoft Windows or Apple's OS X, creates a set of virtual addresses for each program. The OS translates virtual addresses into physical ones, dynamically fitting programs into RAM as it becomes available.

Paging

Virtual memory breaks programs into fixed-size blocks called pages. If a computer has abundant physical memory, the operating system loads all of a program's pages into RAM. If not, the OS fits as much as it can and runs the instructions in those pages. When the computer is done with those pages, it loads the rest of the program into RAM, possibly overwriting earlier pages. Because the operating system automatically manages these details, this frees the software developer to concentrate on program features and not worry about memory issues.

Multiprogramming

Virtual memory with paging lets a computer run many programs at the same time, almost regardless of available RAM. This benefit, called multiprogramming, is a key feature of modern PC operating systems, as they accommodate many utility programs such as printer drivers, network managers and virus scanners at the same time as your applications -- Web browsers, word processors, email and media players.

Paging File

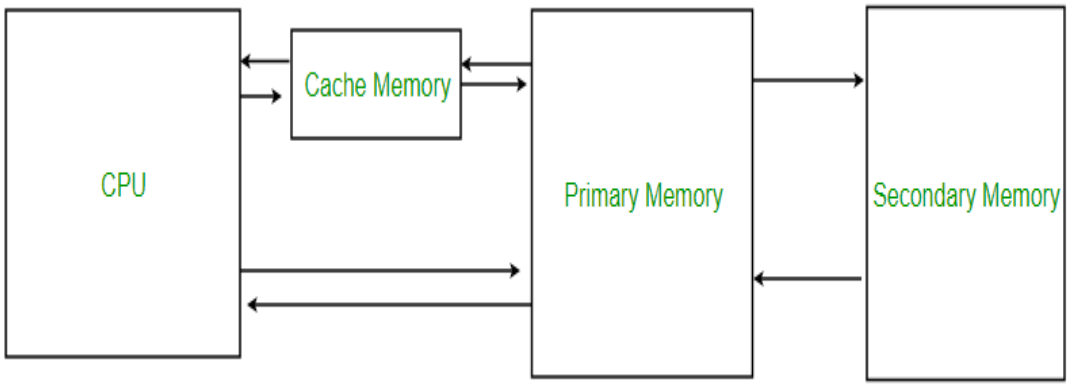
With virtual memory, the computer writes program pages that have not been recently used to an area on the hard drive called a paging file. The file saves the data contained in the pages; if the program needs it again, the operating system reloads it when RAM becomes available. When many programs compete for RAM, the act of swapping pages to the file can slow a computer's processing speed, as it spends more time doing memory management chores and less time getting useful work done. Ideally, a computer will have enough RAM to handle the

demands of many programs, minimizing the time the computer spends managing its pages.

Memory Protection

A computer without virtual memory can still run many programs at the same time, although one program might change, accidentally or deliberately, the data in another if its addresses point to the wrong program. Virtual memory prevents this situation because a program never "sees" its physical addresses. The virtual memory manager protects the data in one program from changes by another.

Cache Memory is a special very high-speed memory. It is used to speed up and synchronize with high-speed CPU. Cache memory is costlier than main memory or disk memory but more economical than CPU registers. Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed. Cache memory is used to reduce the average time to access data from the Main memory. The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations. There are various different independent caches in a CPU, which store instructions and data.



Cache Performance: When the processor needs to read or write a location in main memory, it first checks for a corresponding entry in the cache.

- If the processor finds that the memory location is in the cache, a **cache hit** has occurred and data is read from the cache.
- If the processor **does not** find the memory location in the cache, a **cache miss** has occurred. For a cache miss, the cache allocates a new entry and copies in data from main memory, then the request is fulfilled from the contents of the cache.

The performance of cache memory is frequently measured in terms of a quantity called **Hit ratio.**

Hit ratio = hit / (hit + miss) = no. of hits/total accesses

Cache performance can be improved by using higher cache block size, and higher associativity, reduce miss rate, reduce miss penalty, and reduce the time to hit in the cache.

Architecture of 80286 Microprocessor:

The Architecture of 80286 Microprocessor is an advanced, high-performance microprocessor with specially optimized capabilities for multi-user and multitasking systems. The 80286 has built-in memory protection that supports operating system and task isolation as well as program and data privacy. A 12 MHz 80286 provides about six times more than the 5 MHz 8086. The 80286 includes memory management capabilities that map 2^{30} (one gigabyte) of virtual address space per task into 2^{24} bytes (16 megabytes) of physical memory.

The 80286 is compatible with 8086 and 8088 operating software. The Architecture of 80286 Microprocessor has two operating modes: **real address mode** and **protected virtual address mode**. In real address mode, the 80286 is object code compatible with existing 8086, and 8088 software. In protected virtual address mode, the 80286 is source code compatible with 8086, 8088 software and sometimes it may require upgrading to use virtual addresses supported by the 80286's integrated memory management and protection scheme. Both modes operate at full 80286 performances and execute all instructions of the 8086 and 8088 processors.

The internal block diagram of the Architecture of 80286 Microprocessor is depicted in Fig. 11.15. The CPU of the 80286 processor consists of four functional units such as

1. Address Unit (AU)
2. Bus Unit (BU)
3. Instruction Unit (IU)
4. Execution Unit (EU)

Address Unit (AU) The address unit (AU) is used to determine the physical addresses of instructions and operands which are stored in memory. The AU computes the 20-bit physical address based on the contents of the segment register and 16-bit offset just like 8086. The address lines derived by AU can be used to address different peripheral devices such as memory and I/O devices. This physical address computed by the address unit is sent to the Bus Unit (BU) of the CPU.

Bus Unit (BU) The bus unit interfaces the 80286 with memory and I/O devices. This processor has a 16-bit data bus, a 24-bit address bus, and a control bus. The bus unit is responsible for performing all external bus operations. This unit consists of latches and drivers for the address bus, which transmit the physical address A_{19}-A_0. The A_{19}-A_0 facilitates all memory and I/O devices for read and write operations.

The bus unit is used to fetch instruction bytes from the memory. Generally, the instructions are fetched in advance and stored in a queue for faster execution of the instructions. This concept is known as **instruction pipelining**.

Hence, to fetch instruction, the CPU will not wait till the completion of execution of the previous instruction. While one instruction is being executed, the subsequent instruction is to be prefetched, decoded and kept ready for execution. The prefetcher module in the bus unit performs this task of prefetching. The bus unit has a bus control module which controls the prefetcher module. The fetched instructions are arranged in a 6-byte prefetch queue. In this way, the CPU prefetches the instructions, to enhance the speed of execution.

Instruction Unit (IU) The 6-byte prefetch queue forwards the instructions sequentially to the Instruction Unit (IU). The instruction unit receives instructions from the prefetch queue and an instruction decoder decodes them one by one. The decoded instructions are latched onto a decoded instruction queue. The IU decodes maximum 3 prefetched instructions and loads them into decoded instruction queue for execution by execution unit.

Execution Unit (EU) The output of the decoded instruction queue is fed to a control circuit of the execution unit. This unit is responsible for executing the instructions received from the decoded instruction queue. The execution unit consists of the register bank, arithmetic and logic unit (ALU) and control block. The register bank is used for storing the data as a scratch pad. The register hank can also be used as special-purpose registers. The ALU is the core of the EU. and perform all the arithmetic and logical operations and sends the results either over the data bus or back to the register bank. The control block controls the overall operation of the execution unit.

The 80286 CPU family contains all the basic set of registers, instructions, and addressing modes of 8086. The 80286 processor is upward compatible with the 8086, 8088, and 80186 CPU's. In this section, operations of registers are explained elaborately.

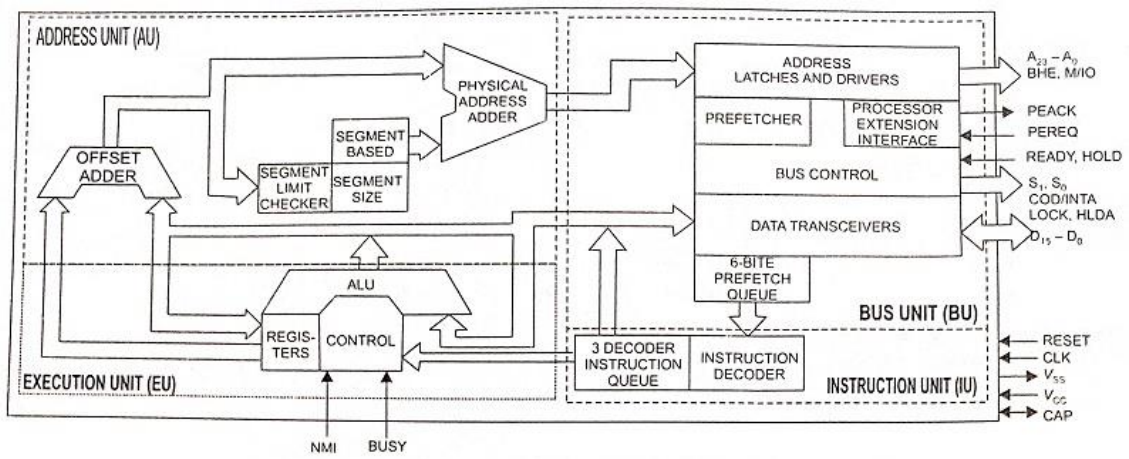


Fig. 11.15 *Internal block diagram of the 80286 processor*

Architecture of 80486 Microprocessor:

The 80486DX is a 32-bit processor. Figure 11.46 shows the simplified block diagram of 80486 and the internal architecture of 80486 Microprocessor is depicted in Fig. 11.47.

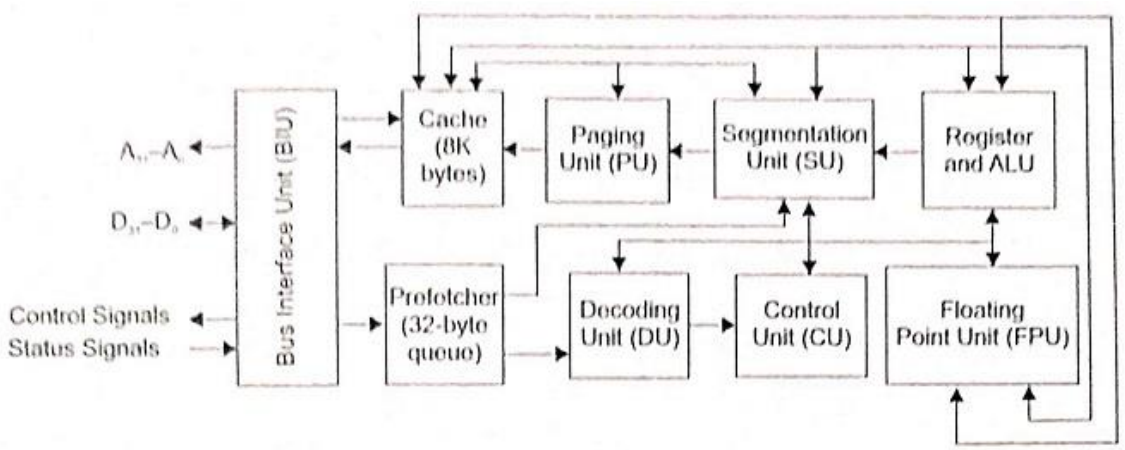


Fig. 11.46 *The simplified block diagram of 80486*

The architecture of Intel's 80486 can be divided into three different sections such as

- Bus interface unit (BIU),
- Execution and control unit (EU), and
- Floating-point unit (FU).

- **Bus Interface Unit (BIU)** The bus interface unit is used to organize all the bus activities of the processor. The address driver is connected with the internal 32-bit address output of the cache and the system bus. The data bus transreceivers are interconnected between the internal 32-bit data bus and system bus. The write data buffer is a queue of four 80-bit registers and is able to hold the 80-bit data which will he written to the memory. Due to pipelined execution of the write operation, data must be available in advance. To control the bus access and operations, the following bus control and request sequencer signals $\overline{ADS}$, W/$\overline{R}$, D/$\overline{C}$, M/$\overline{IO}$, PCD, PWT, $\overline{RDY}$, $\overline{LOCK}$, $\overline{PLOCK}$, $\overline{BOFF}$, $\overline{A20M}$, BREQ, HOLD, HLDA, RESET, INTR, NMI, $\overline{FERR}$ and $\overline{IGNNE}$ are used.
- **Execution Unit (EU) and Control Unit (CU)** The burst control signal updates the processor that the burst is ready. This signal works as a ready signal in the burst cycle. The $\overline{BLAST}$ output shows that the previous burst cycle is over. The bus size control signals $\overline{BS16}$ and $\overline{BS8}$ indicates dynamic bus sizing. The cache control signals $\overline{KEN}$, FLUSH, AHOLD and $\overline{EADS}$ are used to control the cache control unit.

- **Floating–point Unit (FPU)** The floating-point unit and register banks or FPU communicate with the bus interface unit (BIU) under the control of memory management unit (MMU), through a 64-bit internal data bus. Generally, the FPU is used for mathematical data processing at very high speed as compared to the ALU.

Pentium Processor:

Pentium Microprocessor is one of the powerful family members of Intel's X86 microprocessor. It is an advanced superscalar 32-bit microprocessor, introduced in the year 1993 that contains around 3.1 million transistors.

It has a 64-bit data bus and a 32-bit address bus that offers 4 Gb of physical memory space. While the maximum clock rating offered is around 60 to 233 MHz.

The architectural representation of the Pentium processor is considered to be an advancement of 80386 and 80486 microprocessors. Basically, Pentium has included modifications related to cache structure, the width of the data bus, numeric coprocessor with faster speed along with providing dual integer processor.

In the case of a Pentium processor, there are two caches, one for caching data while another for caching information and each one is of 8K size. By using a dual integer processor, two instructions can be executed in each clock cycle. The data bus width in Pentium is 64-bit which was 32-bit in 80386 and the numeric coprocessor exhibits quite a faster speed than that of 80486.

Following are the features that Pentium processor offers:

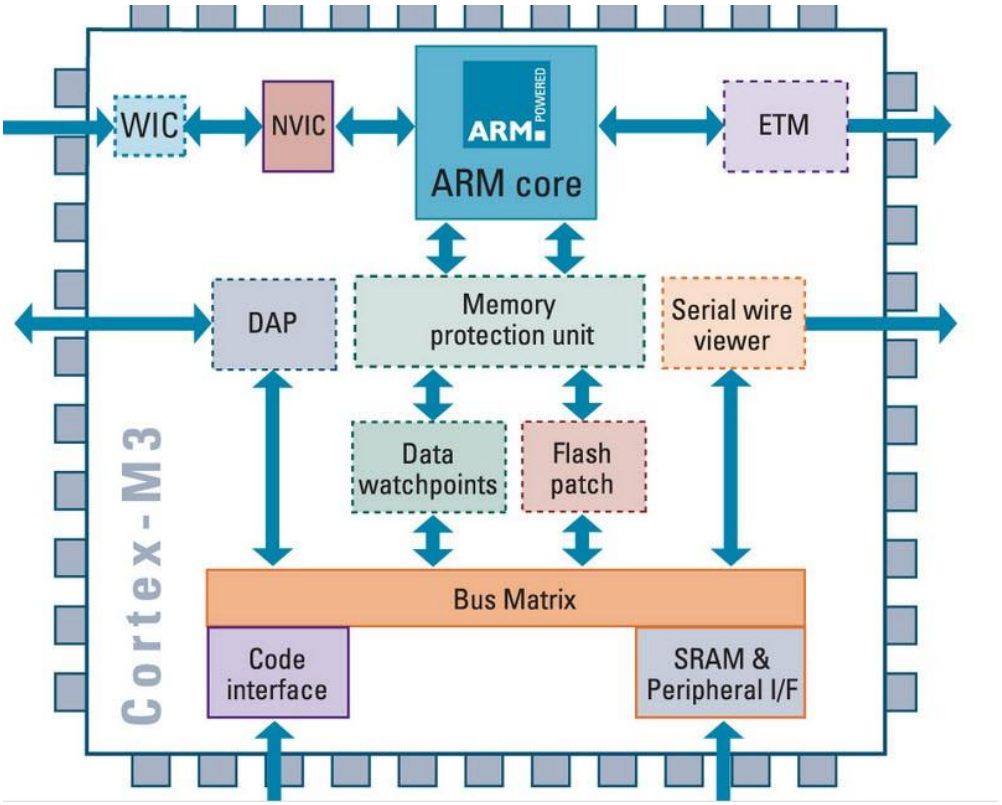
- Superscalar architecture
- Separate data and instruction caches
- Bus cycle pipelining
- Execution tracing
- 64-bit data bus
- Internal parity checking
- Dynamic branch prediction
- Dual processing support
- Performance monitoring

ARM microcontroller

The ARM microcontroller stands for Advance RISC Machine; it is one of the extensive and most licensed processor cores in the world. The first ARM processor was developed in the year 1978 by Cambridge University, and the first ARM RISC processor was produced by the Acorn Group of Computers in the year 1985. These processors are specifically used in portable devices like digital cameras, mobile phones, home networking modules and wireless communication technologies and other embedded systems due to the benefits, such as low power consumption, reasonable performance, etc.

ARM Architecture

The ARM architecture processor is an advanced reduced instruction set computing [RISC] machine and it's a 32bit reduced instruction set computer (RISC) microcontroller. It was introduced by the Acron computer organization in 1987. This ARM is a family of microcontroller developed by makers like ST Microelectronics, Motorola, and so on. The ARM architecture comes with totally different versions like ARMv1, ARMv2, etc., and, each one has its own advantage and disadvantages.



RISC Processor

RISC stands for **Reduced Instruction Set Computer Processor**, a microprocessor architecture with a simple collection and highly customized set of instructions. It is built to minimize the instruction execution time by optimizing and limiting the number of instructions. It means each instruction cycle requires only one clock cycle, and each cycle contains three parameters: fetch, decode and execute. The RISC processor is also used to perform various complex instructions by combining them into simpler ones. RISC chips require several transistors, making it cheaper to design and reduce the execution time for instruction.

Examples of RISC processors are SUN's SPARC, PowerPC, Microchip PIC processors, RISC-V.

Advantages of RISC Processor

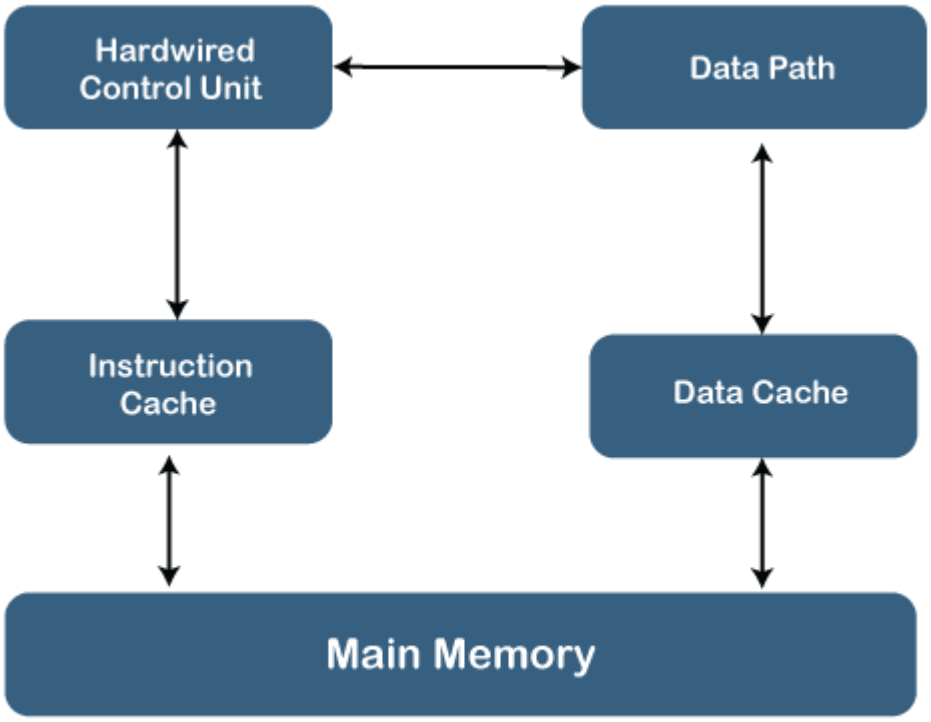
1. The RISC processor's performance is better due to the simple and limited number of the instruction set.
2. It requires several transistors that make it cheaper to design.
3. RISC allows the instruction to use free space on a microprocessor because of its simplicity.
4. RISC processor is simpler than a CISC processor because of its simple and quick design, and it can complete its work in one clock cycle.

Disadvantages of RISC Processor

1. The RISC processor's performance may vary according to the code executed because subsequent instructions may depend on the previous instruction for their execution in a cycle.
2. Programmers and compilers often use complex instructions.
3. RISC processors require very fast memory to save various instructions that require a large collection of cache memory to respond to the instruction in a short time.

RISC Architecture

It is a highly customized set of instructions used in portable devices due to system reliability such as Apple iPod, mobiles/smartphones, Nintendo DS,



RISC Architecture

Features of RISC Processor

Some important features of RISC processors are:

1. **One cycle execution time:** For executing each instruction in a computer, the RISC processors require one CPI (Clock per cycle). And each CPI includes the fetch, decode and execute method applied in computer instruction.
2. **Pipelining technique:** The pipelining technique is used in the RISC processors to execute multiple parts or stages of instructions to perform more efficiently.
3. **A large number of registers:** RISC processors are optimized with multiple registers that can be used to store instruction and quickly respond to the computer and minimize interaction with computer memory.
4. It supports a simple addressing mode and fixed length of instruction for executing the pipeline.
5. It uses LOAD and STORE instruction to access the memory location.
6. Simple and limited instruction reduces the execution time of a process in a RISC.

CISC Processor

The CISC Stands for **Complex Instruction Set Computer**, developed by the Intel. It has a large collection of complex instructions that range from simple to very complex and specialized in the assembly language level, which takes a long time to execute the instructions. So, CISC approaches reducing the number of instruction on each program and ignoring the number of cycles per instruction. It emphasizes to build complex instructions directly in the hardware because the hardware is always faster than software. However, CISC chips are relatively slower as compared to RISC chips but use little instruction than RISC. Examples of CISC processors are VAX, AMD, Intel x86 and the System/360.

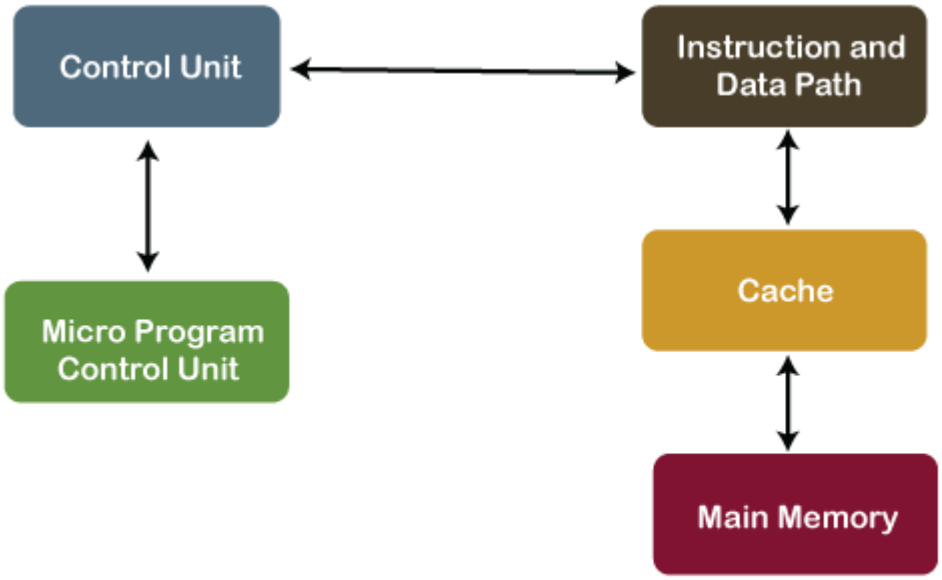
Characteristics of CISC Processor

Following are the main characteristics of the RISC processor:

1. The length of the code is shorts, so it requires very little RAM.
2. CISC or complex instructions may take longer than a single clock cycle to execute the code.
3. Less instruction is needed to write an application.
4. It provides easier programming in assembly language.
5. Support for complex data structure and easy compilation of high-level languages.
6. It is composed of fewer registers and more addressing nodes, typically 5 to 20.
7. Instructions can be larger than a single word.
8. It emphasizes the building of instruction on hardware because it is faster to create than the software.

CISC Processors Architecture

The CISC architecture helps reduce program code by embedding multiple operations on each program instruction, which makes the CISC processor more complex. The CISC architecture-based computer is designed to decrease memory costs because large programs or instruction required large memory space to store the data, thus increasing the memory requirement, and a large collection of memory increases the memory cost, which makes them more expensive.



CISC Architecture

Advantages of CISC Processors

1. The compiler requires little effort to translate high-level programs or statement languages into assembly or machine language in CISC processors.
2. The code length is quite short, which minimizes the memory requirement.
3. To store the instruction on each CISC, it requires very less RAM.
4. Execution of a single instruction requires several low-level tasks.
5. CISC creates a process to manage power usage that adjusts clock speed and voltage.
6. It uses fewer instructions set to perform the same instruction as the RISC.

Disadvantages of CISC Processors

1. CISC chips are slower than RSIC chips to execute per instruction cycle on each program.
2. The performance of the machine decreases due to the slowness of the clock speed.
3. Executing the pipeline in the CISC processor makes it complicated to use.
4. The CISC chips require more transistors as compared to RISC design.
5. In CISC it uses only 20% of existing instructions in a programming event.

Difference between the RISC and CISC Processors

RISC	CISC
It is a Reduced Instruction Set Computer.	It is a Complex Instruction Set Computer.
It emphasizes on software to optimize the instruction set.	It emphasizes on hardware to optimize the instruction set.
It is a hard wired unit of programming in the RISC Processor.	Microprogramming unit in CISC Processor.
It requires multiple register sets to store the instruction.	It requires a single register set to store the instruction.
RISC has simple decoding of instruction.	CISC has complex decoding of instruction.
Uses of the pipeline are simple in RISC.	Uses of the pipeline are difficult in CISC.
It uses a limited number of instruction that requires less time to execute the instructions.	It uses a large number of instruction that requires more time to execute the instructions.
It uses LOAD and STORE that are independent instructions in the register-to-register a program's interaction.	It uses LOAD and STORE instruction in the memory-to-memory interaction of a program.
RISC has more transistors on memory registers.	CISC has transistors to store complex instructions.
The execution time of RISC is very short.	The execution time of CISC is longer.
RISC architecture can be used with high-end	CISC architecture can be used with low-end

applications like telecommunication, image processing, video processing, etc.	applications like home automation, security system, etc.
It has fixed format instruction.	It has variable format instruction.
The program written for RISC architecture needs to take more space in memory.	Program written for CISC architecture tends to take less space in memory.
Example of RISC: ARM, PA-RISC, Power Architecture, Alpha, AVR, ARC and the SPARC.	Examples of CISC: VAX, Motorola 68000 family, System/360, AMD and the Intel x86 CPUs.