Name: أحمد محسن سعد حميدة

ID : 116

G : 1

Sec : 5

GitHub Link:

GitHub page:

* Computer architecture system have 5 webpage

1. Home

 <div id="home">

        <p>

            In computer engineering, computer architecture is a set of rules and methods that describe the

            functionality, organization, and implementation of computer systems.

            Some definitions of architecture define it as describing the capabilities and programming model of a

            computer but not a particular implementation

        </p>

    </div>

1. Classification

 <div id="classification">

        <table>

            <caption>classification of Computer architecture</caption>

            <tr>

                <td>SISD</td>

                <td>single instruction stream , single data stream</td>

            </tr>

            <tr>

                <td>MISD</td>

                <td>multiply instruction stream , single data stream</td>

            </tr>

            <tr>

                <td>SIMD</td>

                <td>single instruction stream , multiply data stream</td>

            </tr>

            <tr>

                <td>MISD</td>

                <td>multiply instruction stream , multiply data stream</td>

            </tr>

        </table>

    </div>

1. Impelmentation

 <div class="Implementation">

        <p>

            An instruction set architecture (ISA) is the interface between the computer's software and hardware and also

            can be viewed as the programmer's view of the machine. Computers do not understand high-level programming

            languages such as Java, C++, or most programming languages used. A processor only understands instructions

            encoded in some numerical fashion, usually as binary numbers. Software tools, such as compilers, translate

            those high level languages into instructions that the processor can understand.<br>

            Besides instructions, the ISA defines items in the computer that are available to a program—e.g., data

            types, registers, addressing modes, and memory. Instructions locate these available items with register

            indexes (or names) and memory addressing modes.<br>

            The ISA of a computer is usually described in a small instruction manual, which describes how the

            instructions are encoded. Also, it may define short (vaguely) mnemonic names for the instructions. The names

            can be recognized by a software development tool called an assembler. An assembler is a computer program

            that translates a human-readable form of the ISA into a computer-readable form. Disassemblers are also

            widely available, usually in debuggers and software programs to isolate and correct malfunctions in binary

            Logic implementation designs the circuits required at a logic-gate level.

            Circuit implementation does transistor-level designs of basic elements (e.g., gates, multiplexers, latches)

            as well as of some larger blocks (ALUs, caches etc.) that may be implemented at the logic-gate level, or

            even at the physical level if the design calls for it.

            Physical implementation draws physical circuits. The different circuit components are placed in a chip

            floorplan or on a board and the wires connecting them are created.

            Design validation tests the computer as a whole to see if it works in all situations and all timings. Once

            the design validation process starts, the design at the logic level are tested using logic emulators.

            However, this is usually too slow to run a realistic test. So, after making corrections based on the first

            test, prototypes are constructed using Field-Programmable Gate-Arrays (FPGAs). Most hobby projects stop at

            this stage. The final step is to test prototype integrated circuits, which may require several redesigns.

    </div>

1. computer organization

 <div id="Computer organization">

        <p>

            Computer organization helps optimize performance-based products. For example, software engineers need to

            know the processing power of processors. They may need to optimize software in order to gain the most

            performance for the lowest price. <br>

            For example, in an SD card, the designers might need to arrange the card so that the most data can be

            processed in the fastest possible way.<br>

            Computer organization also helps plan the selection of a processor for a particular project. Multimedia

            projects may need very rapid data access, while virtual machines may need fast interrupts. Sometimes certain

            tasks need additional components as well. For example, a computer capable of running a virtual machine needs

            virtual memory hardware so that the memory of different virtual computers can be kept separated. <br>

            Computer organization and features also affect power consumption and processor cost.

        </p>

        <img src="https://media.geeksforgeeks.org/wp-content/uploads/vn\_cpu.png" alt="computer organization">

    </div>

1. Power effienciecy

 <div id="Power efficiency">

        <p>

            Modern computer performance is often described in instructions per cycle (IPC), which measures the

            efficiency of the architecture at any clock frequency; a faster IPC rate means the computer is faster. Older

            computers had IPC counts as low as 0.1 while modern processors easily reach near 1. <br>

            Superscalar processors may reach three to five IPC by executing several instructions per clock

            cycle.[citation needed]<br>

            Counting machine-language instructions would be misleading because they can do varying amounts of work in

            different ISAs. The "instruction" in the standard measurements is not a count of the ISA's machine-language

            instructions, but a unit of measurement, usually based on the speed of the VAX computer architecture.<br>

            Many people used to measure a computer's speed by the clock rate (usually in MHz or GHz). This refers to the

            cycles per second of the main clock of the CPU. However, this metric is somewhat misleading, as a machine

            with a higher clock rate may not necessarily have greater performance. As a result, manufacturers have moved

            away from clock speed as a measure of performance.<br>

            Other factors influence speed, such as the mix of functional units, bus speeds, available memory, and the

            type and order of instructions in the programs.<br>

            There are two main types of speed: latency and throughput. Latency is the time between the start of a

            process and its completion. Throughput is the amount of work done per unit time. Interrupt latency is the

            guaranteed maximum response time of the system to an electronic event (like when the disk drive finishes

            moving some data).<br>

            Performance is affected by a very wide range of design choices — for example, pipelining a processor usually

            makes latency worse, but makes throughput better. Computers that control machinery usually need low

            interrupt latencies. These computers operate in a real-time environment and fail if an operation is not

            completed in a specified amount of time. For example, computer-controlled anti-lock brakes must begin

            braking within a predictable and limited time period after the brake pedal is sensed or else failure of the

        </p>

    </div>

And This All pages

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Computer architecture </title>

</head>

<body>

    <p class="logo">Computer architecture</p>

    <!-- list of anchor -->

    <ul>

        <li><a href="#home">home</a> </li>

        <li><a href="#classification">classification</a></li>

        <li><a href="#Implementation">Implementation</a></li>

        <li><a href="#Computer organization">Computer organization</a></li>

        <li><a href="#Power efficiency">Power efficiency</a></li>

    </ul>

    <div id="home">

        <p>

            In computer engineering, computer architecture is a set of rules and methods that describe the

            functionality, organization, and implementation of computer systems.

            Some definitions of architecture define it as describing the capabilities and programming model of a

            computer but not a particular implementation

        </p>

    </div>

    <!-- table for Computer architecture -->

    <div id="classification">

        <table>

            <caption>classification of Computer architecture</caption>

            <tr>

                <td>SISD</td>

                <td>single instruction stream , single data stream</td>

            </tr>

            <tr>

                <td>MISD</td>

                <td>multiply instruction stream , single data stream</td>

            </tr>

            <tr>

                <td>SIMD</td>

                <td>single instruction stream , multiply data stream</td>

            </tr>

            <tr>

                <td>MISD</td>

                <td>multiply instruction stream , multiply data stream</td>

            </tr>

        </table>

    </div>

    <div class="Implementation">

        <p>

            An instruction set architecture (ISA) is the interface between the computer's software and hardware and also

            can be viewed as the programmer's view of the machine. Computers do not understand high-level programming

            languages such as Java, C++, or most programming languages used. A processor only understands instructions

            encoded in some numerical fashion, usually as binary numbers. Software tools, such as compilers, translate

            those high level languages into instructions that the processor can understand.<br>

            Besides instructions, the ISA defines items in the computer that are available to a program—e.g., data

            types, registers, addressing modes, and memory. Instructions locate these available items with register

            indexes (or names) and memory addressing modes.<br>

            The ISA of a computer is usually described in a small instruction manual, which describes how the

            instructions are encoded. Also, it may define short (vaguely) mnemonic names for the instructions. The names

            can be recognized by a software development tool called an assembler. An assembler is a computer program

            that translates a human-readable form of the ISA into a computer-readable form. Disassemblers are also

            widely available, usually in debuggers and software programs to isolate and correct malfunctions in binary

            Logic implementation designs the circuits required at a logic-gate level.

            Circuit implementation does transistor-level designs of basic elements (e.g., gates, multiplexers, latches)

            as well as of some larger blocks (ALUs, caches etc.) that may be implemented at the logic-gate level, or

            even at the physical level if the design calls for it.

            Physical implementation draws physical circuits. The different circuit components are placed in a chip

            floorplan or on a board and the wires connecting them are created.

            Design validation tests the computer as a whole to see if it works in all situations and all timings. Once

            the design validation process starts, the design at the logic level are tested using logic emulators.

            However, this is usually too slow to run a realistic test. So, after making corrections based on the first

            test, prototypes are constructed using Field-Programmable Gate-Arrays (FPGAs). Most hobby projects stop at

            this stage. The final step is to test prototype integrated circuits, which may require several redesigns.

    </div>

    <div id="Computer organization">

        <p>

            Computer organization helps optimize performance-based products. For example, software engineers need to

            know the processing power of processors. They may need to optimize software in order to gain the most

            performance for the lowest price. <br>

            For example, in an SD card, the designers might need to arrange the card so that the most data can be

            processed in the fastest possible way.<br>

            Computer organization also helps plan the selection of a processor for a particular project. Multimedia

            projects may need very rapid data access, while virtual machines may need fast interrupts. Sometimes certain

            tasks need additional components as well. For example, a computer capable of running a virtual machine needs

            virtual memory hardware so that the memory of different virtual computers can be kept separated. <br>

            Computer organization and features also affect power consumption and processor cost.

        </p>

        <img

            src="https://upload.wikimedia.org/wikipedia/commons/thumb/e/ea/MIPS\_Architecture\_%28Pipelined%29.svg/481px-MIPS\_Architecture\_%28Pipelined%29.svg.png">

    </div>

    <div id="Power efficiency">

        <p>

            Modern computer performance is often described in instructions per cycle (IPC), which measures the

            efficiency of the architecture at any clock frequency; a faster IPC rate means the computer is faster. Older

            computers had IPC counts as low as 0.1 while modern processors easily reach near 1. <br>

            Superscalar processors may reach three to five IPC by executing several instructions per clock

            cycle.[citation needed]<br>

            Counting machine-language instructions would be misleading because they can do varying amounts of work in

            different ISAs. The "instruction" in the standard measurements is not a count of the ISA's machine-language

            instructions, but a unit of measurement, usually based on the speed of the VAX computer architecture.<br>

            Many people used to measure a computer's speed by the clock rate (usually in MHz or GHz). This refers to the

            cycles per second of the main clock of the CPU. However, this metric is somewhat misleading, as a machine

            with a higher clock rate may not necessarily have greater performance. As a result, manufacturers have moved

            away from clock speed as a measure of performance.<br>

            Other factors influence speed, such as the mix of functional units, bus speeds, available memory, and the

            type and order of instructions in the programs.<br>

            There are two main types of speed: latency and throughput. Latency is the time between the start of a

            process and its completion. Throughput is the amount of work done per unit time. Interrupt latency is the

            guaranteed maximum response time of the system to an electronic event (like when the disk drive finishes

            moving some data).<br>

            Performance is affected by a very wide range of design choices — for example, pipelining a processor usually

            makes latency worse, but makes throughput better. Computers that control machinery usually need low

            interrupt latencies. These computers operate in a real-time environment and fail if an operation is not

            completed in a specified amount of time. For example, computer-controlled anti-lock brakes must begin

            braking within a predictable and limited time period after the brake pedal is sensed or else failure of the

        </p>

    </div>

</body>

</html>