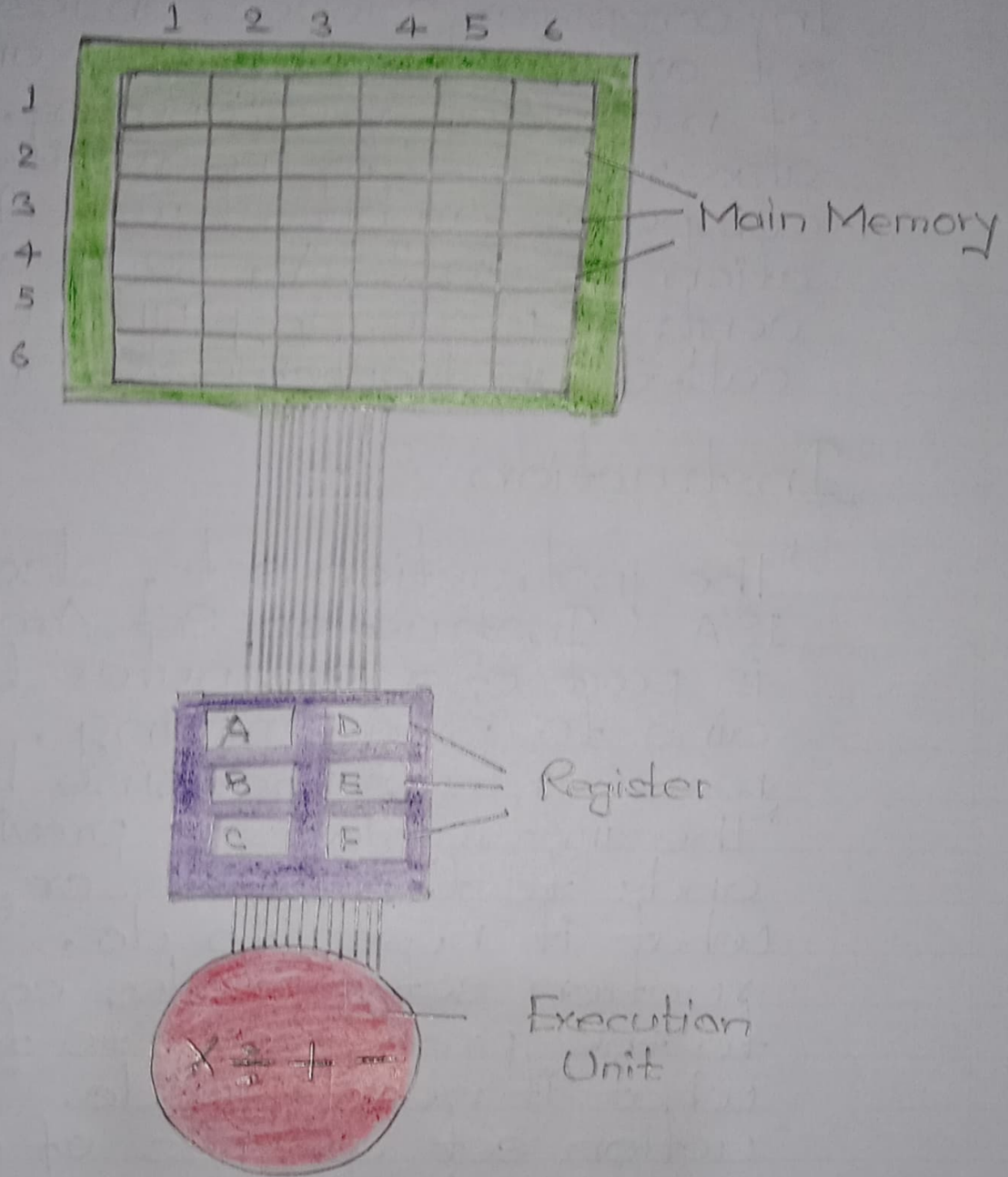
ASSIGNMENT - 2

NAME: AMRITANSHU KESHARI
BRANCH: COMP. SCIENCE & ENGG.
ROLL NUMBER: 19402060006
SUBJECT: C.H.P.
SESSION: 2019-22
SEMESTER: 4th
COLLEGE: GOVT. POLYTECHNIC COLLEGE, ADITYAPUR

In computer science, an instruction set architecture (ISA) is an abstract model of a computer. It is also referred to as architecture or computer architecture. A realization of an ISA, such as a central processing unit (CPU), is called an implementation.

Instruction Set

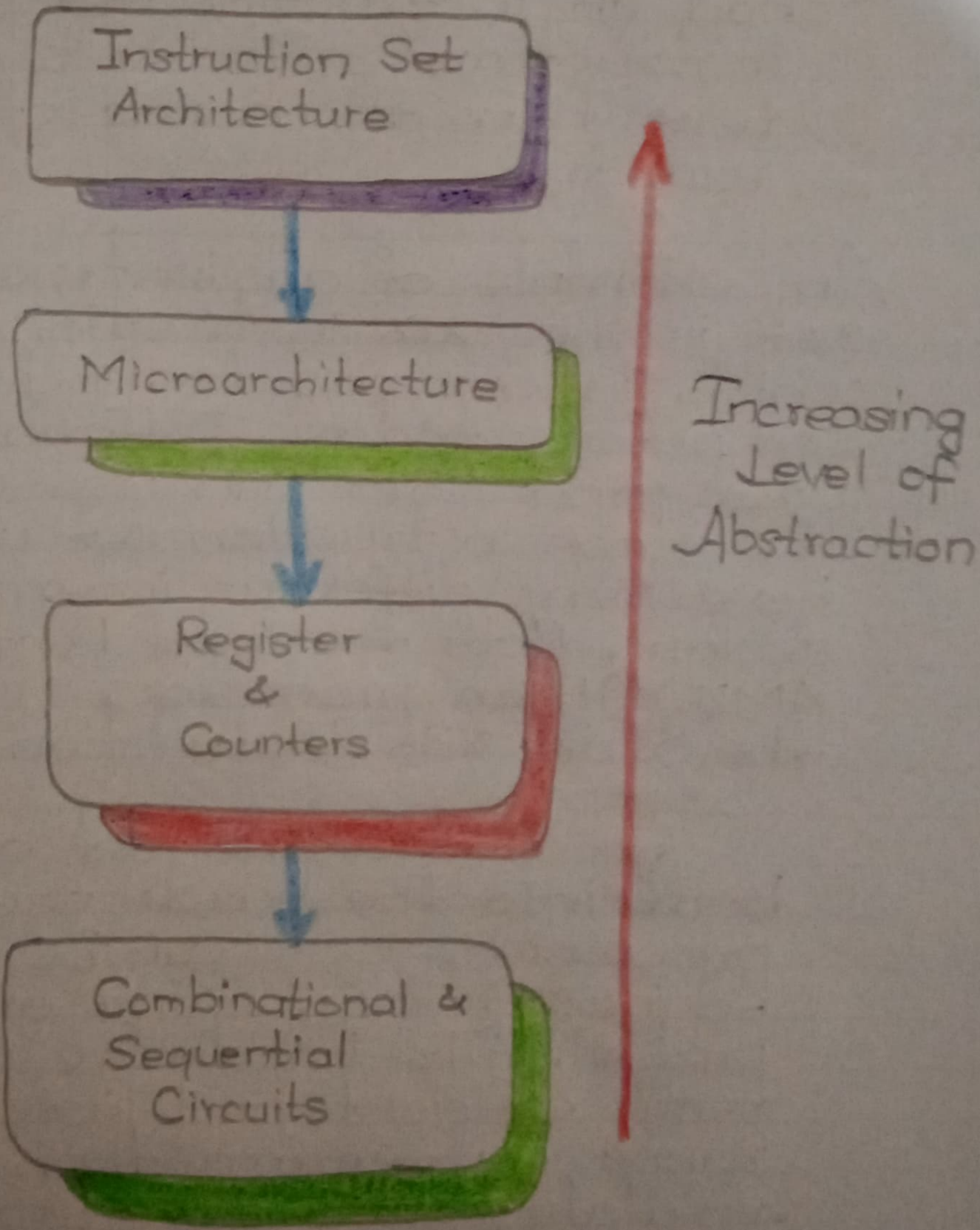
The instruction set, also called ISA (Instruction Set Architecture), is part of a computer that pertains to programming, which is more or less machine language. The instruction set provides commands to the processor, to tell it what it needs to do. The instruction set provides commands to the processor, to tell it what it needs to do. The instruction set consists of addressing modes, instructions, native data types, registers, memory architecture, interrupt and exception handling and external I/O (input/output).


1 2 3 4 5 6
1
2
3
4
5
6
Main Memory
A
D
B
E
C
F
Register
Execution Unit

The first CPU, the Intel 4004, had an instruction set of 46 instructions. Today's computers have thousands of instructions.

An example of an instruction set is the x86 instruction set, which is common to find on computers today. Different computer processors can use almost the same instruction set while still have very different internal design design. Both the Intel Pentium and AMD Athlon processors use nearly the same x86 instruction set.

An instruction set can be built into the hardware of the processor, or it can be emulated in software, using an interpreter. The hardware design is more efficient and faster for running programs than the emulated software version. Some instructions are simple read, write and move commands that direct data to different hardware.


Instruction Set Architecture
Microarchitecture
Register & Counters
Combinational & Sequential Circuits
Increasing Level of Abstraction

Examples of instruction set

(i) ADD - Add two numbers (numerical values) together.

(ii) COMPARE - To compare numbers.

(iii) IN - input information from a device, e.g., keyboard.

(iv) JUMP - jump to designated RAM (Random Access Memory) address.

(v) JUMP IF - conditional statements that jumps to a designated RAM address.

(vi) LOAD - Load information from RAM to the CPU.

(vii) OUT - Output information to device, e.g., monitor.

(viii) STORE - to store information to RAM (Random Access Memory).

An instruction set is a group of commands for a CPU in machine language. The term can refer to all possible instructions for a CPU or a subnet of instructions to enhance its performance in certain situations.

All CPUs have instruction sets that enable commands to the processor directing the CPU to switch the relevant transistors. Some instructions are simple read, write and move commands that direct data to different hardware.

In CISC (Complex Instruction Set Computer) processors there is also a microcode layer, which involves programmable instructions stored in fast memory that may be updated. The RISC (Reduced Instruction Set Computer) architecture, on the other hand, has hard-wired control; it does not require microcode but has a greater base instruction set.

Enhancement instruction sets are more familiar to users as they have often been used in marketing a given CPU. Examples of this go back to the Pentium 166 MHz with MMX technologies marketed for enhancing Intel CPU multimedia performance.

MMX stands for Multimedia extenstions and refers to the extended instruction set. Other examples include MMX+, 3DNow!, 3DNow!+, SSE, SSE2, SSSE3, SSE4.1, SSE4A, AVX, AVX2 and XOP.

If it's 1s and 0s, how do we actually communicate with the processor? Systems programmers use assembly language, which is a type of programming language. The statements are assembled into machine language -- into the instructions that the processor can understand. Assembly language is a lot like machine language with

labels -- because binary alone would quickly get hard to keep track of.

What makes up an instruction set? When we start talking about 1s and 0s, machine language, assembly language and registers, it might all become murkey. Lets start with the overall structure of an instruction and go from there.

The instruction set consists of a limited set of unique codes that let the processor know what to do next, along with some basic rules of how to express them.

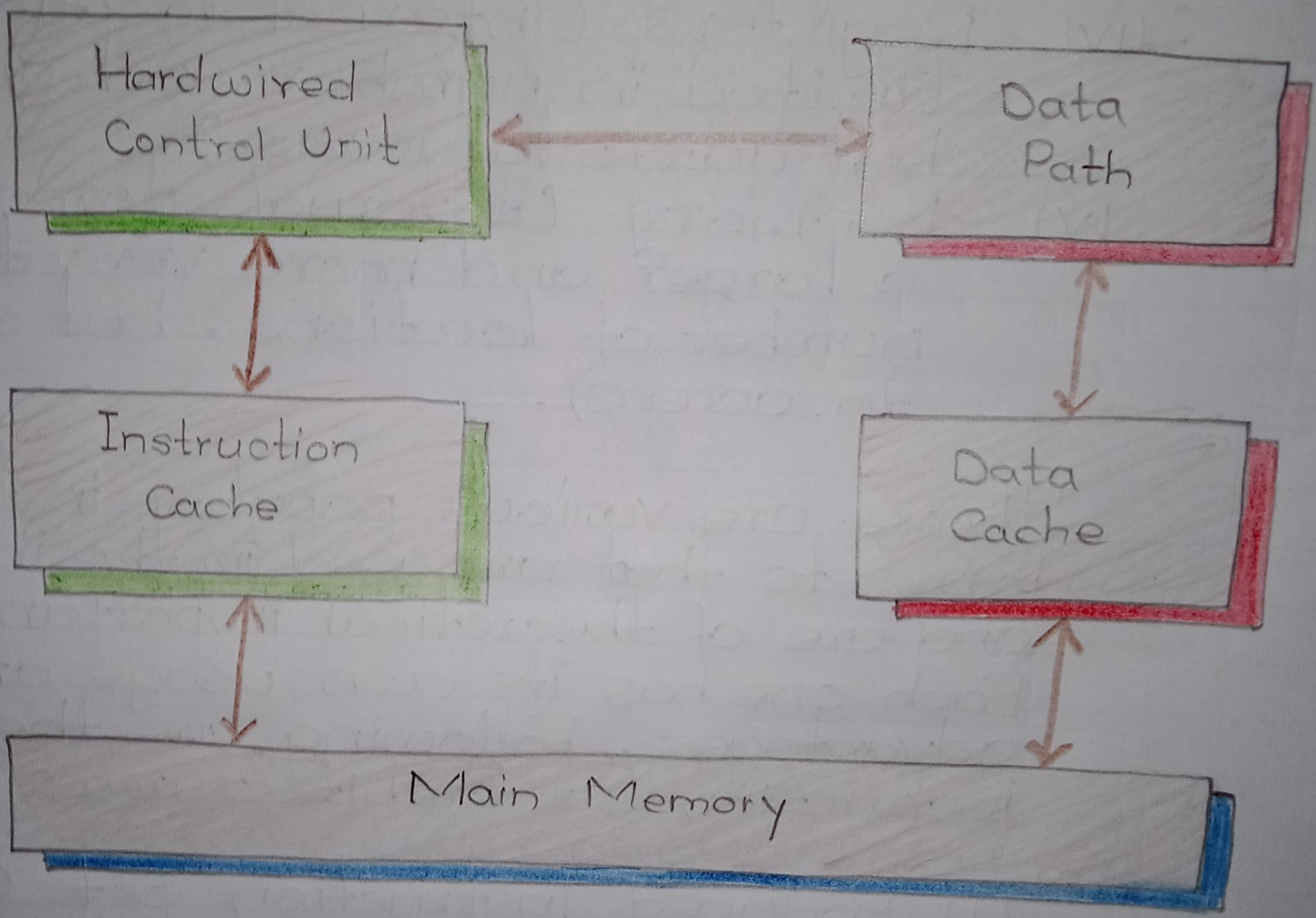
Here is some brief explaination about it using the following points are as follows as:

(i) Instruction length (this can vary).
(ii) Opcodes (the command to be carried out).

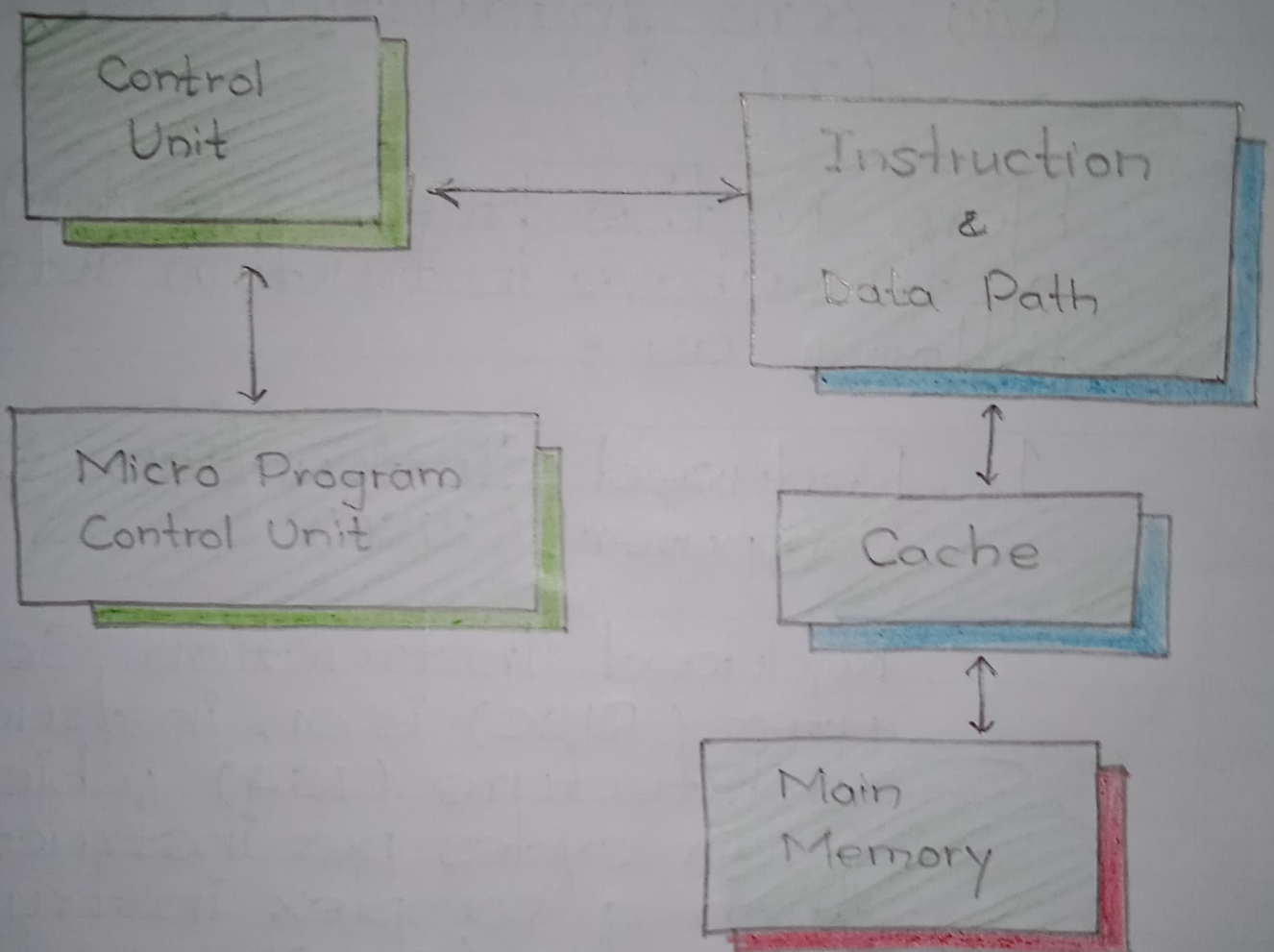
(iii) Operands (what the command will operate on)
(iv) Registers (internal locations -- limited in number and ability, but quick to access.).
(v) Memory (external storage -- a larger and more versatile number of locations, but slower to access).

There are various popular instruction sets that are used in the industry and are of theoretical importance. Each one has its own usage and advantages. Following are the instruction set architectures:

(i) Reduced Instruction Set Computer (RISC).
(ii) Complex Instruction Set Computer (CISC)
(iii) Minimal Instruction Set Computer (MISC)
(iv) Very Long Instruction Word (VLIW)
(v) Explicitly parallel Instruction Computing (EPIC)



RISC Architecture



CISC Architecture

(vi) One Instruction Set Computer (OISC)
(vii) Zero Instruction Set Computer (ZISC)

Here is the brief overview of each of the above instruction sets are as follows as :

1. Reduced Instruction Set Computer (RISC)

Reduced Instruction Set Computer (RISC) is an instruction set architecture (ISA) which has fewer cycles per instruction (CPI) than a complex instruction set computer (CISC).

2. Complex Instruction Set Computer (CISC)

Complex Instruction Set Computer (CISC) is an instruction set architecture (ISA) which has fewer instructions per program than a Reduced instruction set computer (RISC).

3. Minimal Instruction Set Computers (MISC)

Minimal instruction set computers (MISC) is a processor architecture with a very small number of basic instruction operations and corresponding opcodes.

4. Very long Instruction Word (VLIW)

Very long Instruction word (VLIW) is an instruction set architectures designed to exploit instruction level parallelism (ILP).

5. Explicitly Parallel Instruction Computing (EPIC)

Explicitly parallel instruction computing (EPIC) is an instruction set that permits microprocessors to execute software instructions in parallel by using the compiler, rather than complex

on-die circuitry, to control parallel instruction execution. This was intended to allow simple performance scaling without resorting to higher clock frequencies.

6. One Instruction Set Computer (OISC)

One instruction set computer (OISC) is an abstract machine that uses only one instruction obviating the need for a machine language opcode. OISCs have been recommended as guides in teaching computer architecture and have been used as computational models in structural computing research.

7. Zero Instruction Set Computer (ZISC)

Zero instruction set computer (ZISC) is an instruction set. It is a computer architecture based on pattern matching and absence of (micro-)

(micro-) instructions in the classical sense.

These chips are known for being throught of as comparable to the neutral networks being marked marketed for the number of "synapses" and "neurons".