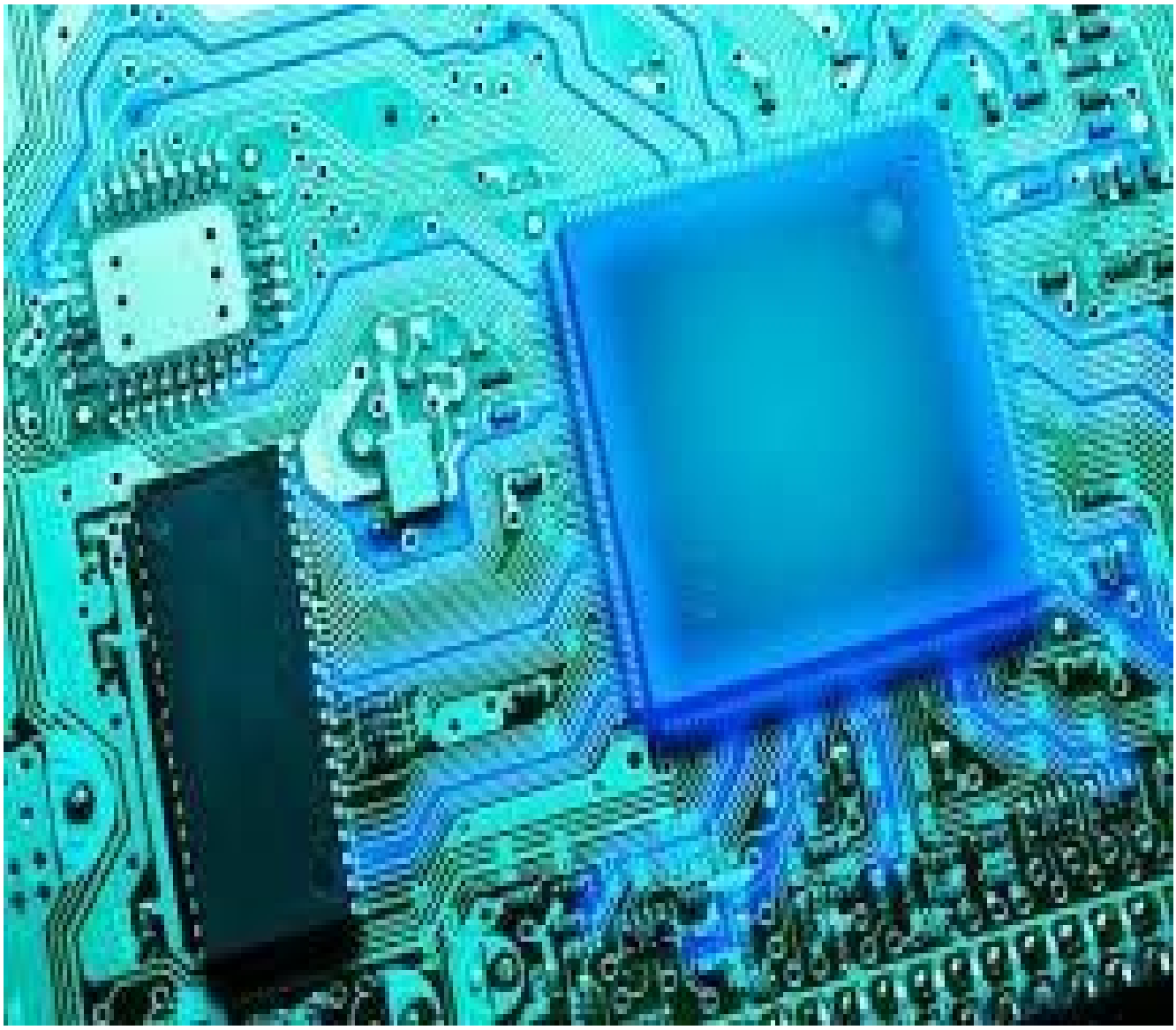


Student Name-Loveleen Kaur

Module Name-Computer Systems

Module Code-(CSY1014-STD)

Word Count- **2208**

TABLE OF CONTENT

LIST OF ABBREVIATIONS

CPU- Central Processing Unit

ALU- Arithmetic Logic Unit

IC- Instruction Decoder

I/O- Input/ Output

AD- Address Bus

CU- Control Unit

FSC- Fairchild Semiconductors

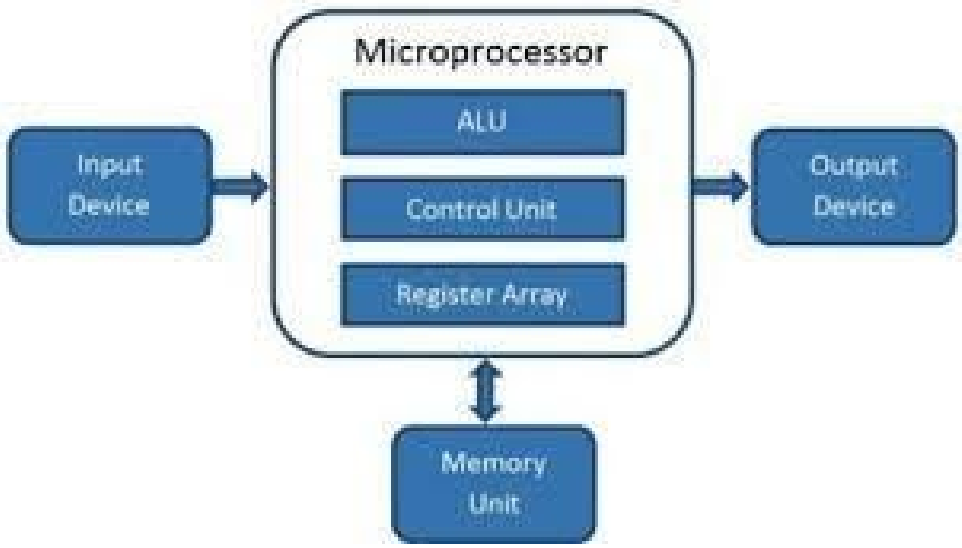
MAR- Memory Address Register

MBR- Memory Buffer Register

CIR- Current Instruction Register

RISC- Reduced Instruction Set Computer

CISC- Complex Instruction Set Computer



SUMMARY

Microprocessors are thought to be among one of the most important components of digital computers. It is also known as the brain of the computer as it accesses memory. Microprocessors incorporate arithmetic and logic functional units as well as the associated control logic, instruction processing circuitry and a portion of memory hierarchy. In this report, I have discussed Microprocessor, its history, fetch-execute cycle, pipelining technique along with advantages and disadvantages of RISC and CISC.

INTRODUCTION

Microprocessor is a programmable digital electronic component that incorporates the functions of a central processing unit on a single semiconductor integrated circuit. It is a CPU fabricated on a single small chip. It consists of a processor, Arithmetic Logic Unit, Instruction Decoder and a few registers which are used to store data for mathematical and logical operations. The necessary tools for microprocessor includes- Central Processing Unit, Input/ Output, Address bus and data bus, Timing and Control Unit, Special and general Purpose Register, L1 and L2 Cache memory, Bus Interface etc. Microprocessors have made possible inexpensive handheld electronic calculators and digital wrist watches. Nowadays multiple microprocessors work together and act as the heart of data centers, supercomputers, portable devices, communications products and other digital devices. They play supporting roles with larger companies as smart controllers for graphics display, storage devices and high speed printers. It is also used to control electric devices to operate automatic tracking and targeting systems in aircrafts and missiles.

HISTORY

The first computer was invented in the 1940's using bulky relay and vacuum tube switches. Fairchild semiconductors founded and invented the first IC in 1959. Robert Noyce, Gordan Moore and Andrew Grove in 1968 made their own company Intel. Three projects Intel's 4004, Texas Instructions' TMS 1000, and Garrett AiResearch Central Air Data Computer arguably delivered a complete microprocessor at same time. Microprocessor was born by reducing the word size of CPU from 32-bit to 4-bit so that the transistors of its logic circuit would fit only a single part. The first microprocessor invented was of 4-bit, then 8-bit, 16-bit, 32-bit and then presently used 64-bit.

S.NO.	BIT	INTEL	YEAR	TRANSISTORS
First microprocessor	4-bit	Intel 4004	1971	2300

Second microprocessor	8-bit	Intel 8008	1972	3300
Third microprocessor	16-bit	Intel 8086	1974	4500
Fourth microprocessor	32-bit	Intel 80386	1889	1.2 million

4-bit	8-bit	16-bit	32-bit	64- bit
Intel 4004	Intel 8008	Intel 8086	Intel 80386	Intel core2
Intel 4040	Intel 8080	Intel 8088	Intel 80486	Intel core i7
	Intel 8085	Intel 80186	Intel Pentium	Intel core i5
		Intel 80188	Intel Pentium2	Intel core i3
		Intel 80286	Intel Pentium 2 xeon	
			Intel Pentium 3	
			Intel Pentium 4	
			Intel dual core	

FETCH- EXECUTE CYCLE

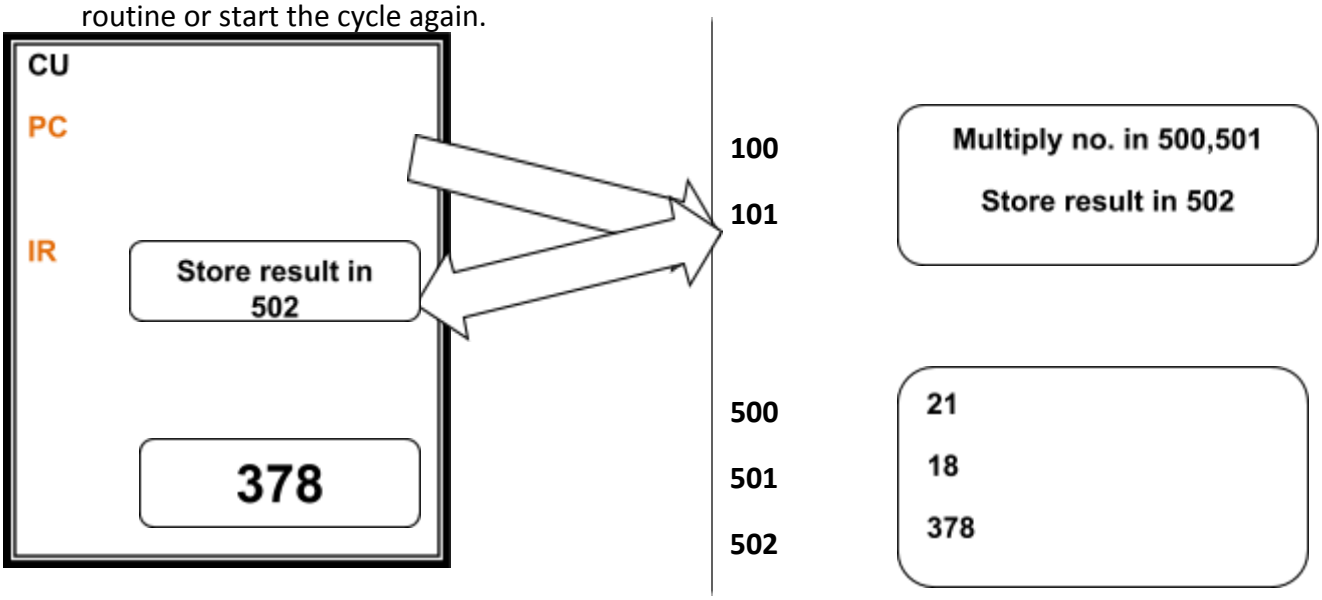
The data and program that acts upon that data are loaded into main memory by the operating system. The CPU is now ready to do some work.

Fetch Cycle

- The Program Counter contains the address of the next instruction to be fetched
- The address contained in the PC is copied to the Memory Address Register
- The instruction is copied from the memory location contained in the MAR and placed in the Memory Buffer Register
- The entire instruction is copied from the MBR and placed in the Current Instruction register
- The PC is incremented so that it points to the next instruction to be fetched

Execute cycle

- The address part of the instruction is placed in the MAR
- The instruction is decoded and excited
- The processor checks for interrupts and either branch to the relevant interrupts service routine or start the cycle again.

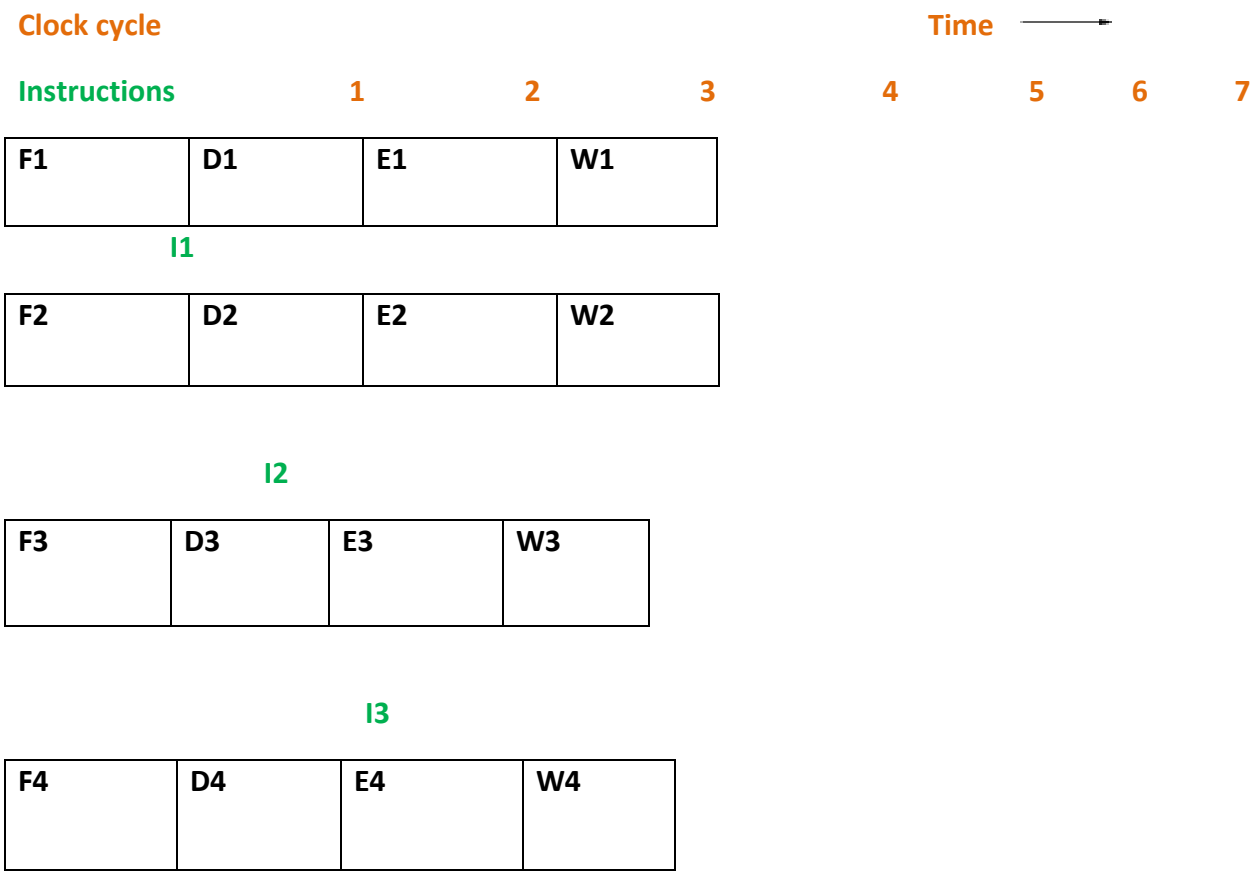


- The PC contains the address of location 101
- CU fetches instruction in location 101
- A copy of the instruction is saved in the IR
- Increment the PC
- Activated the right circuit to execute the instructions

PIPELINING

Pipelining is the process of accumulating instructions from the processor. It is widely used in modern processors. It improves system performance in terms of throughput. Pipelined organization requires sophisticated compilation techniques i.e. includes decomposing a sequential process into sub operation completed in a dedicated segment. It is commonly known as an assembly line operation. First thing is an assembly line set up a chassis, next is installing the engine, another group of workers fitting the body. The processor executes the program by fetching and executing instructions one after another.

Any condition that causes a pipeline to stall is called a hazard and includes Data hazards, structural hazards and instruction hazards. Any delay in the availability of an instruction, situation when two instructions require the use of a given hardware resource at the same time causes the pipeline to stalls. Pipelining does not result in individual instructions being executed faster rather; it is the throughput that increases. Throughput is measured by the rate at which instruction execution is completed.



I4

The time in the horizontal axis is divided into steps of equal duration

F and E refer to the fetch and execute steps for instruction I.

Fetch (F) - read the instruction from the memory

Decode (D) - Decode the instruction

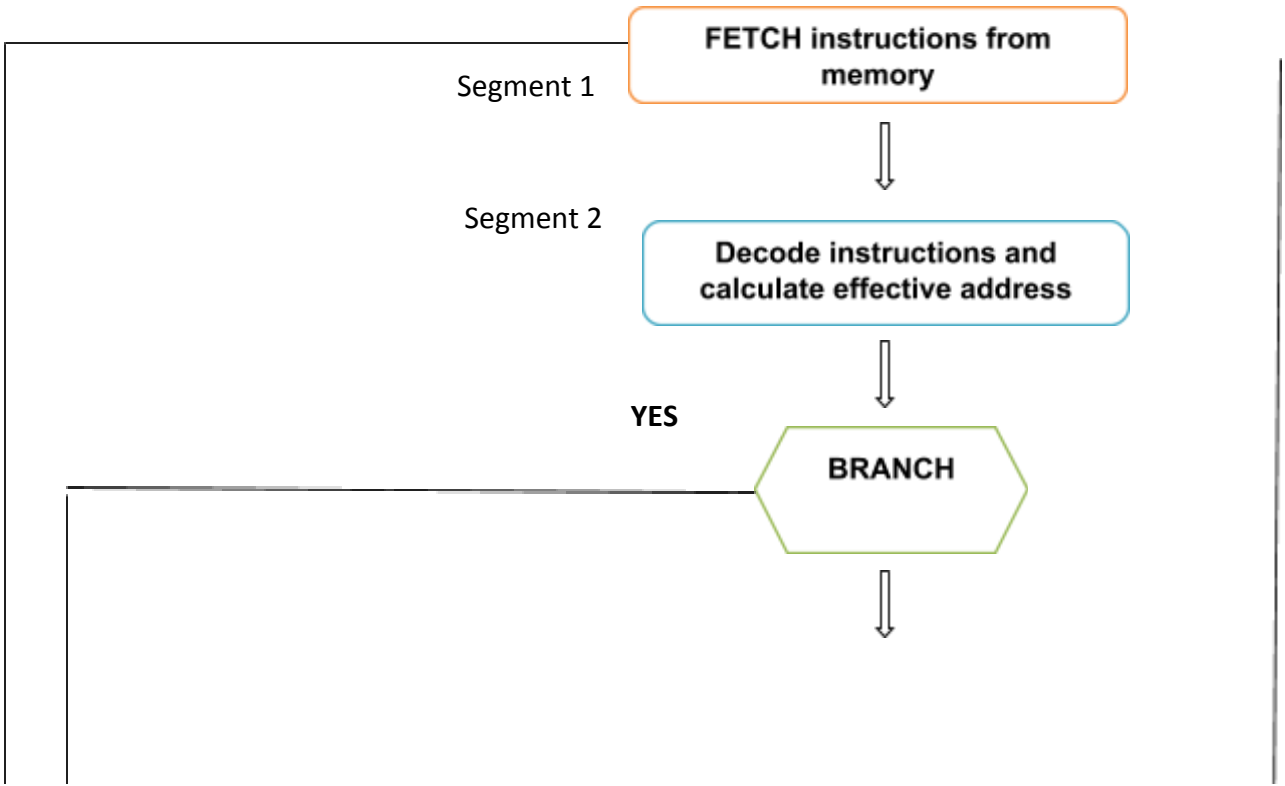
Execute (E) - perform the operation specified by the Instruction

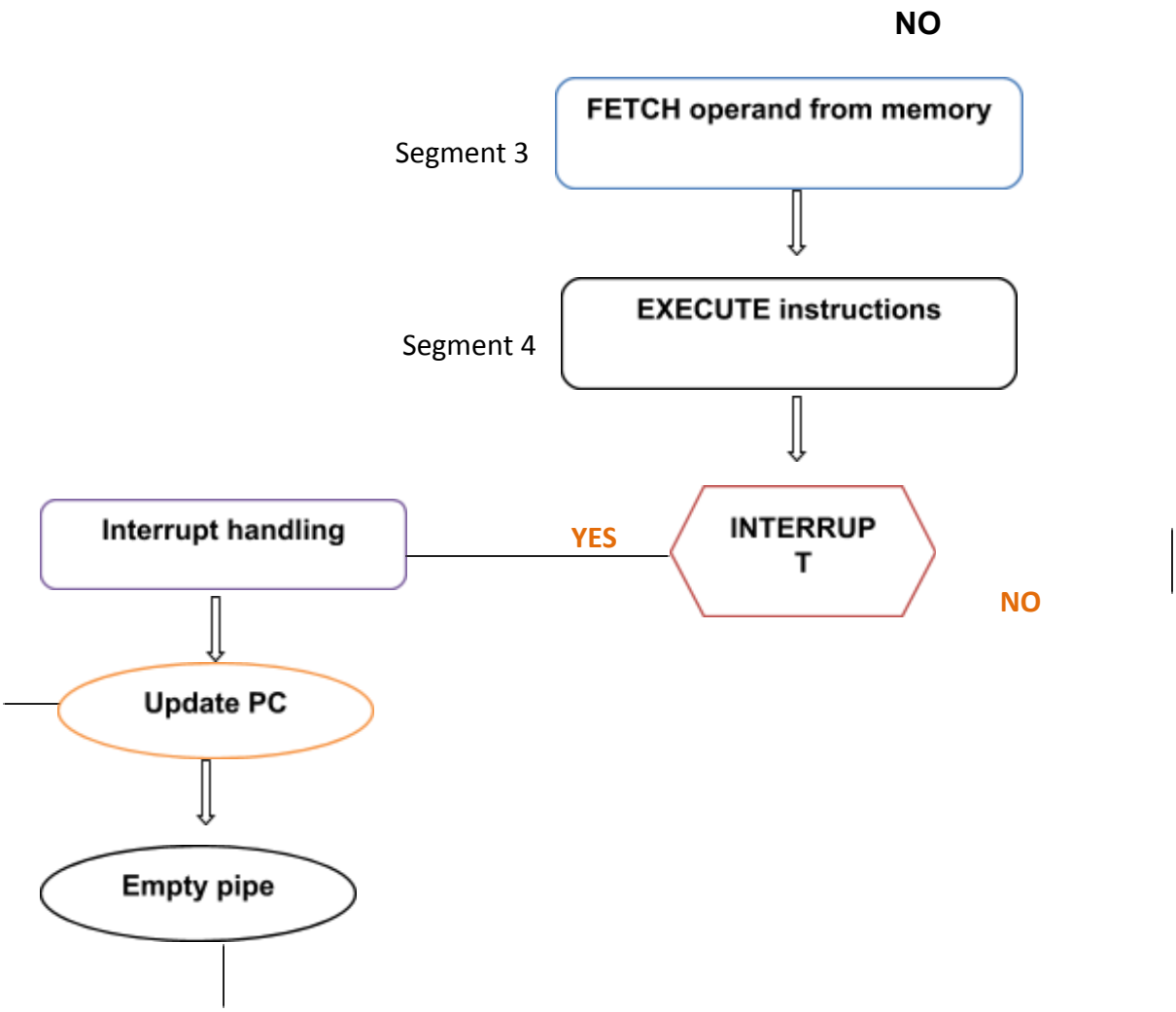
Write (W) - store the result in the destination location

Fetch Decode + Execute write

SPEED UP FACTOR

The increase in performance resulting from Pipelining depends completely on the number of pipeline stages. This increase would be achieved only if all pipeline stages require the same time to complete. In simple words it's just to split the processor into a series of small independent stages. There should not be any interruption throughout program execution unfortunately; in reality it's not possible. It increases the CPU throughput i.e. by increasing the number of instructions completed per unit time. It slightly increases the execution time of each instruction due to overhead in the pipeline. Use faster circuit technology to build the processor and the main memory. Arrange the hardware so that more than once operation can be performed at the same time. The number of operations performed per second is increased even though the elapsed time needed to perform any one operation is not changed.





RISC stands for Reduced Instruction Set Computing. It has reduced and restricted the number of instructions. It uses a lesser number of addressing modes. Instruction format is simple and uniform. Hardwired control rather than microprogrammed. External memory access time is reduced by a large number of registers. RISC processor includes AVR, ARM, PA-RISC and alpha etc. It has both advantages and disadvantages. Some of them are listed below:

Pros	Cons
It has a set of instructions so high- level language compilers can produce more efficient code	Mostly its performance depends on the programmer or compiler as the knowledge of the compiler plays a vital role while charging the CISC code to a RISC code

Because of its simplicity it allows freedom of using the space on microprocessors	Rearranging of CISC code to RISC code, called code expansion, will increase the size
It functions use only a few parameters and its processors cannot use the call instructions therefore use a fixed length instruction which is easy to pipeline	The quality of code expansion depends on the compiler and the machine's instruction set
The speed of the operation can be maximized and the execution time can be minimized	The first level cache of the RISC processors is also a disadvantage in which these processors have large quantity memory catches on the chip itself
Very less number of instructional formats A few addressing models are needed	For feeding the instructions they require very fast memory systems which is not available

ADVANTAGES AND DISADVANTAGES OF RISC

PROS	CONS
Microprogramming is easy assembly language to implement and less expensive than hard writing a control unit	The performance of the machine slows down due to the amount of clock time taken by
The ease of micro coding new instructions allowed designers to make CISC machines upwardly compatible	Only 20% of the existing instructions are used in a typical programming event even though there are various specialized instructions in reality which are not even used frequently

As each instruction became more accomplished fewer instructions could be used to implement a given task It is easy to add new commands into the clip without changing the structure of the instruction set	The conditional codes are set by the CISC instructions as a side effect of each instruction which takes time for this setting, the subsequent instruction changes the condition code bit so the compiler has to examine the condition code bits before this happens

CISC stands for complex Instruction Set Computing.It has large number of instructions variant from 100-250 instructions.It uses a variety of addressing mode varying from 12-14 modes.Instructions format have variable length.Example IBM 370/168 and Intel 80486 etc.It has both advantages and disadvantages.Some of them are listed below:

ADVANTAGES AND DISADVANTAGES OF CISC

RISC V/S CISC

- **Speed**: RISC processors often achieve 2 to 4 times the performance of CISC using comparable semiconductor technology and the same clock rates.
- **Simpler hardware**: The instruction set of the RISC processor is very simple. It uses up much less chip space and simple hardware.
- **Shorter design cycle**: RISC processors are simpler than CISC processors. They can be designed more quickly and can complete their work in 1clock cycle.
- **Code Quality**: The performance of RISC processor depends greatly on the code that it is executing. If the programmer or compiler does a poor job of instruction scheduling, the processor can spend quite a bit of time stalling and waiting for the result of one instruction before it can proceed with a subsequent instruction.
- **Code expansion**: CISC machines perform complex actions with a single instruction, where RISC machines may require multiple instructions for the same action code expansion can be a problem.
- **System Design**: RISC machines require very fast memory systems to feed them instructions. RISC based systems typically contain large memory catches, usually on the chip itself (first- level cache).

CISC Complex instructions are infrequently used by programmers and compilers. Memory references loads and stores are slow and account for a significant fraction of all instructions. Procedure and function calls are a major bottleneck passing arguments. Instruction sets and chips of new generation hardware become more complex with each generation of computers.

Reduced Instruction set Computer	Complex Instruction Set Computer
Very few instructions are present	Large no of instructions are present
Fixed length encoding of the instructions	Variable length encoding of instructions
More hardware oriented	More software oriented as the computer deals with translations.
Includes multi- clock, complex instructions	Single- clock, reduced instructions only
Memory- to- memory: "LOAD" and "STORE" incorporated in instructions.	Register to register: "LOAD" and "STORE" are independent instructions
Slower since instruction can take more than 1 cycle	Faster since instructions usually take 1 instruction cycle

Main objective is less code	Main objective is speed
Instruction size is mostly varies	Instruction size is always a set size
Addressing Models can be complex	Addressing Models are simple
No condition codes are used	Conditions codes are used

CONCLUSION

Microprogramming is as easy as assembly language to implement and much less expensive than hard writing control units. As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory. Because micro- program instruction sets can be written to match the constructs of high- level languages, the computer does not have to be as complicated. The RISC processor is often two to four times faster than CISC because of its simplicity. As CISC utilizes more cycles than RISC therefore is leading. CISC has way more complex instructions than RISC. Moreover RISC design is approximately twice as cost effective as CISC. Another main focus is Pipeline stall that causes degradation in pipeline performance. We need to identify all hazards that may cause the pipeline to stall and to find ways to minimize their impact.

REFERENCES

Borkar, Shekhar, Chien, and Andrew A. (2011) The Future of Microprocessor: Communications of the ACM, vol.54, pp.67-77.

Nichols, A.J. (1976) An overview of Microprocessor Applications: Proceedings of the IEEE, vol. 64(6), pp. 951-953.

Shriver, B.D and Smith (1998) The Anatomy of a High Performance Microprocessor: A System Perceptive with Cdrom.

Orlando, R. V. and Anderson (1981) An overview of the 9900 microprocessor family: IEEE Micra, I (3), pp.38-44

V. Kumar (2014) GPS and GSM based Passenger Tracking System: *International Journal of Computer applications*, vol.100.