

No E-Mail submissions will be accepted.  
Submission formats and file naming:

File name : firstName\_lastName\_lab\_6

File format: pdf or MS Word format

e.g. Donald\_Trump\_lab\_6.pdf

Reading materials

Use the following links and write a one page summary about the movies.

# How Does a Quantum Computer Work?

<https://www.youtube.com/watch?v=g_IaVepNDT4&ab_channel=Veritasium>



The video explains the differences between classical computers and quantum computers, noting on how quantum computing better computational power.

As we know, a classical computer processes information using bits, which can be either 0 or 1 at a time. This is the foundation of all traditional computing. A quantum computer uses qubits or quantum bits, which can have a unique property. They can exist in both 0 and 1 at the same time. This difference is what makes quantum computers able to perform certain types of calculations much faster than old computers.

The video talks about an example where researchers use the outermost electron in a phosphorous atom as a qubit. This works because electrons have a magnetic field, making them behave like tiny bar magnets. In the presence of a magnetic field, the electron will align in either a low-energy state and spin down, or 0; or a high-energy state spin up or 1.

the strange part of quantum mechanics is that before measurement, the electron can exist in both states at once. When measured, the electron changes into either 0 or 1, but before that, it holds a combination of probabilities for both states. This ability is what lets quantum computers to process information in a different way.

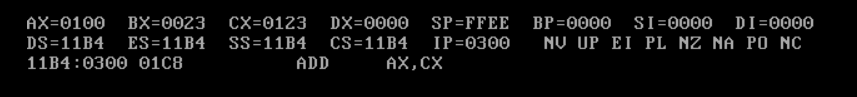
The video talks about multiple qubits where with two classical bits, there are four possible states and a classical computer can only store one of these four values. In a quantum computer, two qubits can exist in a superposition of all four states at once, meaning they can represent four values simultaneously, or 4 constants. If you add a third qubit, it can hold eight values at once, and so on until you get 2^N classical bits of information.

But there’s aproblem. Quantum computers don’t provide better performance for everything. When qubits are measured, they collapse into a single value, meaning all the superposition information is lost. This makes quantum computing useful only for specific types of problems, where the calculations take advantage of quantum parallelism before measurement. The challenge is designing quantum algorithms that extract useful answers while maintaining the quantum advantage.

The video also makes it clear that quantum computers won’t replace classical computers. They are not universally faster—they only reduce the number of operations needed for certain tasks. For stuff like if you’re watching a video, browsing the internet, or writing documents, a quantum computer won’t provide any benefit. But for where parallel calculations are used, quantum computers can be used.

Although quantum computing is a breakthrough, it is not a magic solution for all computing needs. It is another tool useful for specific problems where classical computing struggles.

**1)** The state of 8088 CPU at certain time is given by the following figure. Using this figure obtain next instruction, machine code, and the value of AX, CX, and BX before and after CPU execution.



Before

AX = 0100, BX = 0023, CX = 0123

Next instruction = ADD AX,CX

Machine code = 01C8

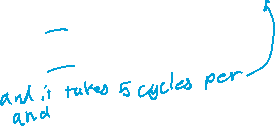
The memory address of the next instruction = 0300

After

AX = 0223, BX = 0023, CX = 0123

**2)** A program consists of 8 floating point instructions and 12 integer instructions. Floating point instructions take 5 cycles and integer instructions take 2 cycles each. Assuming that the processor has a clock rate of 200.0 MHz:

1. Obtain the total cycles.



1. Obtain the total instructions.



1. Obtain the average cycles per instruction.



1. Using the total cycles Obtain the execution time.



1. Using the average cycles per instruction obtain the execution time.



1. Compare the results obtained in parts d and e.



**3)** A program consists of 50,000 instructions as follows:

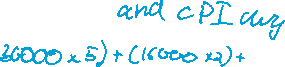
|  |  |  |
| --- | --- | --- |
| **Instruction Type** | **Instruction Count(IC)** | **Cycle per Instruction(CPI)** |
| Floating point | 30,000 | 5 |
| Integer | 16,000 | 2 |
| Data transfer | 4,000 | 6 |

* Complete the following table:

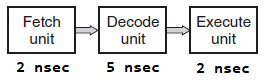
|  |  |  |  |
| --- | --- | --- | --- |
| **Processor** | **Clock Rate**  **(MHz)** | **Cycle Time(CT)**  **(nanosecond)** | **Execution Time**  **(microsecond)** |
| P1 |  | 5 |  |
| P2 | 300 |  |  |
| P3 | 500 |  |  |



* Obtain the average cycles per instruction.



**4)**A computer has a pipeline with 3 stages:



Non-pipelined processor:

1. How many instructions per second can this machine execute?



1. What is the latency of an instruction?



Pipelined processor:

1. How many instructions per second can this machine execute?

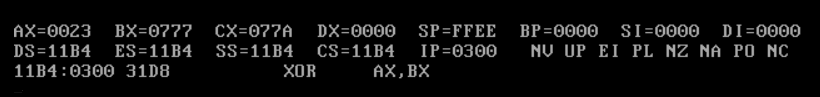


1. What is the latency of an instruction?



**5)**The state of 8088 CPU at certain time is given by the following figures. Using this figures obtain next instruction, machine code, and the value of AX, CX, and BX before and after CPU execution.Do they have same machine codes(figures 1 and 2)?

**Fig 1**



Before

AX = 0023, BX = 0777, CX = 077A

Next instruction = XOR AX,BX

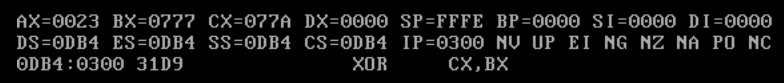
Machine code = 31D8

The location of next instruction = 0300

After

AX = 0754, BX = 0777, CX = 077A

**Fig 2**



Before

AX = 0023, BX = 0777, CX = 077A

Next instruction = XOR CX,BX

Machine code = 31D9

The location of next instruction = 0300

After

AX = 0023, BX = 0777, CX = 000D

No, they are not the same, they are different by 1.