# **Theory related**

## OpenCL is Open Computing Language which is a framework for writing program that executes across heterogenous platforms consisting of CPU, GPU and other processes or hardware acceleration.

## 1.2. It is a technology based on the fact that high quantity processors such as GPU are not in use until and unless any graphics related work is done.

## 1.3. In an experiment it has been proven that some high quantity and less complex calculations can be done concurrently by use of some technology which has high number of processors i.e. a GPU.

## 1.4. Whenever we have a huge quantity of work which has less calculations, we need parallel processing, i.e. a GPU can be used. But when the program includes very complex calculations, we don’t need parallel processing, we only need high frequency CPU.

## 1.5. FLOPS: Flops is a unit of calculation, i.e. how many calculations the device does in 1 sec. Standard today is Teraflops, i.e. 1012 Flops/sec.

## 1.6. Now next thing is why not use only GPU if it has so much functionality like processing and also can have high memory. This is because GPU lacks:

## Virtual Memory

## Means of addressing devices other than memory (e.g. Keyboard, Mouse) (A Processor can maintain communication between all of them).

## Interrupts

## 1.7. CPUs from 2011 support the OpenCL technology.

**CPU HISTORY**

| **YEAR** | **PROGRESS** | **FLOPS** |
| --- | --- | --- |
| **1997** | Intel's ASCI Red (First processor to achieve 1+ TF) | **1 TeraFlop+(1012)** |
| **2007** | IBM (First processor to achieve 1+ PF) | **1 PetaFlop+(1015)** |
| **2008** | AMD released ATI Radeon HD 4800 series, which are reported to be the first GPUs to achieve one teraflop | **1 TeraFlop** |
| **2010** | US Air Force unveiled a defense supercomputer made up of 1,760 PlayStation 3 consoles | **500 TeraFlops** |
| **2019** | Given the current speed of progress, supercomputers are projected to reach 1 exaFLOPS (EFLOPS) in 2018. | **1 ExaFlops (1018)** |

**Hardware costs**

| **YEAR** | **Approximate cost per GFLOPS** | **Approximate cost per TFLOPS** | **Platform** |
| --- | --- | --- | --- |
| **1961** | $18.7 billion | $156.8 trillion | [IBM 7030 Stretch](https://en.wikipedia.org/wiki/IBM_7030_Stretch) |
| **1997** | $30,000 | $46,000,000 | Two 16-processor [Beowulf](https://en.wikipedia.org/wiki/Beowulf_(computing)) clusters with [Pentium Pro](https://en.wikipedia.org/wiki/Pentium_Pro) micro-processors |
| **2000** | $1,000 | $1,440,000 | [Bunyip Beowulf cluster](https://en.wikipedia.org/wiki/Beowulf_cluster) |
| **2003** | $82 | $109,000 | KASY0 |
| **2012** | $0.75 | $800 | Quad AMD Radeon 7970 GHz System |
| **2017** | $0.03 | $30 | Intel Celeron G3930 & AMD RX Vega 64 |

**Future Developments**

Given the current speed of progress, supercomputers are projected to reach 1 exaFLOPS (EFLOPS). Cray, Inc. announced in December 2009 a plan to build a **1 EFLOPS** supercomputer **before 2020**.

**Setting UP OpenCL experience:**

In the starting we tried to find IDE to try and implement difference in processing power of a CPU and a GPU. We found a solution, Microsoft Visual Studio 2015 with Visual C++ support.

Next, we needed an SDK which had header files for C++ needed to assign tasks to GPU or CPU. We tried a lot with Intel SDK but didn’t succeed because it didn’t have the GL folder which was necessary if we have to work with GPU, so we had to use AMD SDK which was a really hectic job, because their official website was not working, but after hours of searching we found an SDK which had both CL and GL folders.

And then we executed 2 codes.

Intial STEP 1: Download MICROSOFT VISUAL STUDIO 2015: https://my.visualstudio.com/Downloads?q=visual%20studio%202015&wt.mc\_id=o~msft~vscom~older-downloads

Initial STEP 2: Download AMD SDK: https://drive.google.com/open?id=1K9JhcXdMRhHOXAPNxkmNk6WvGMh5m6GW

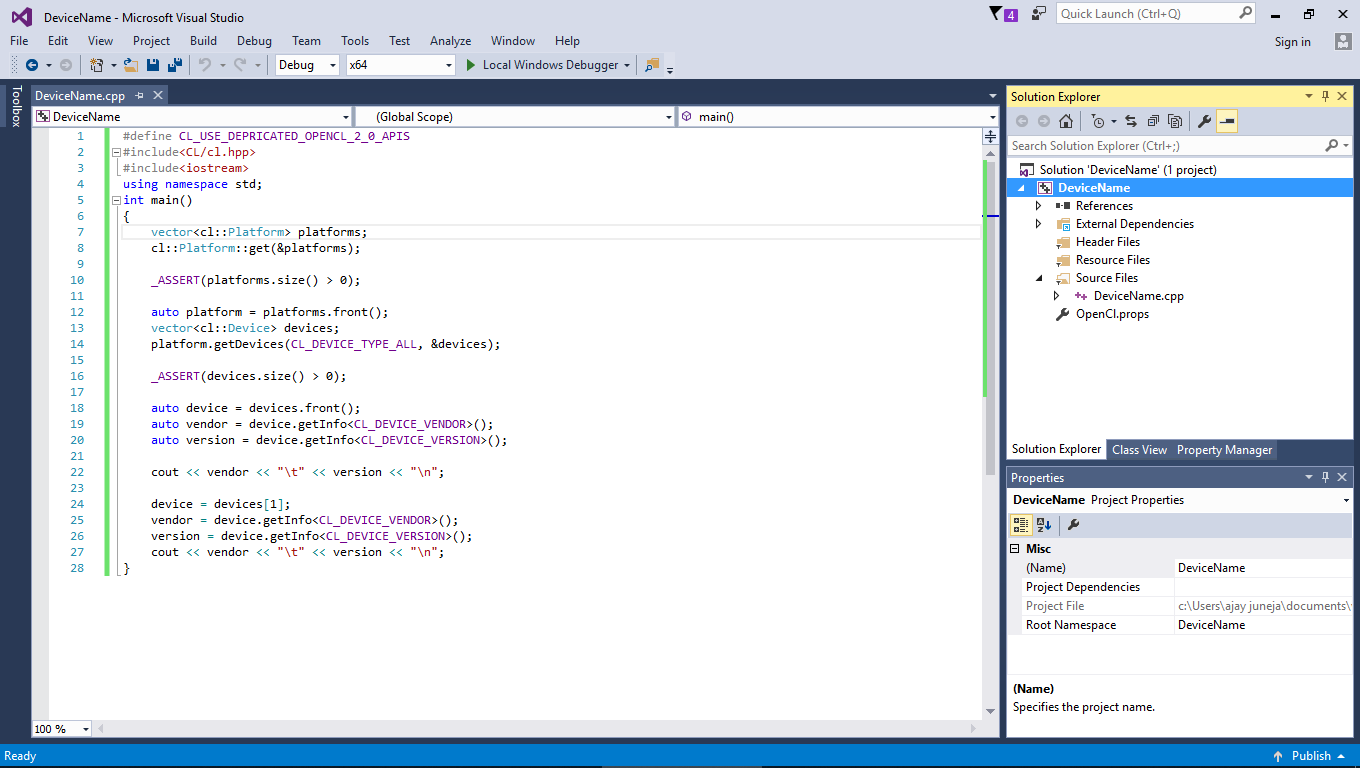
STEP 1: Start a new project -> Empty Project and name it list\_devices.

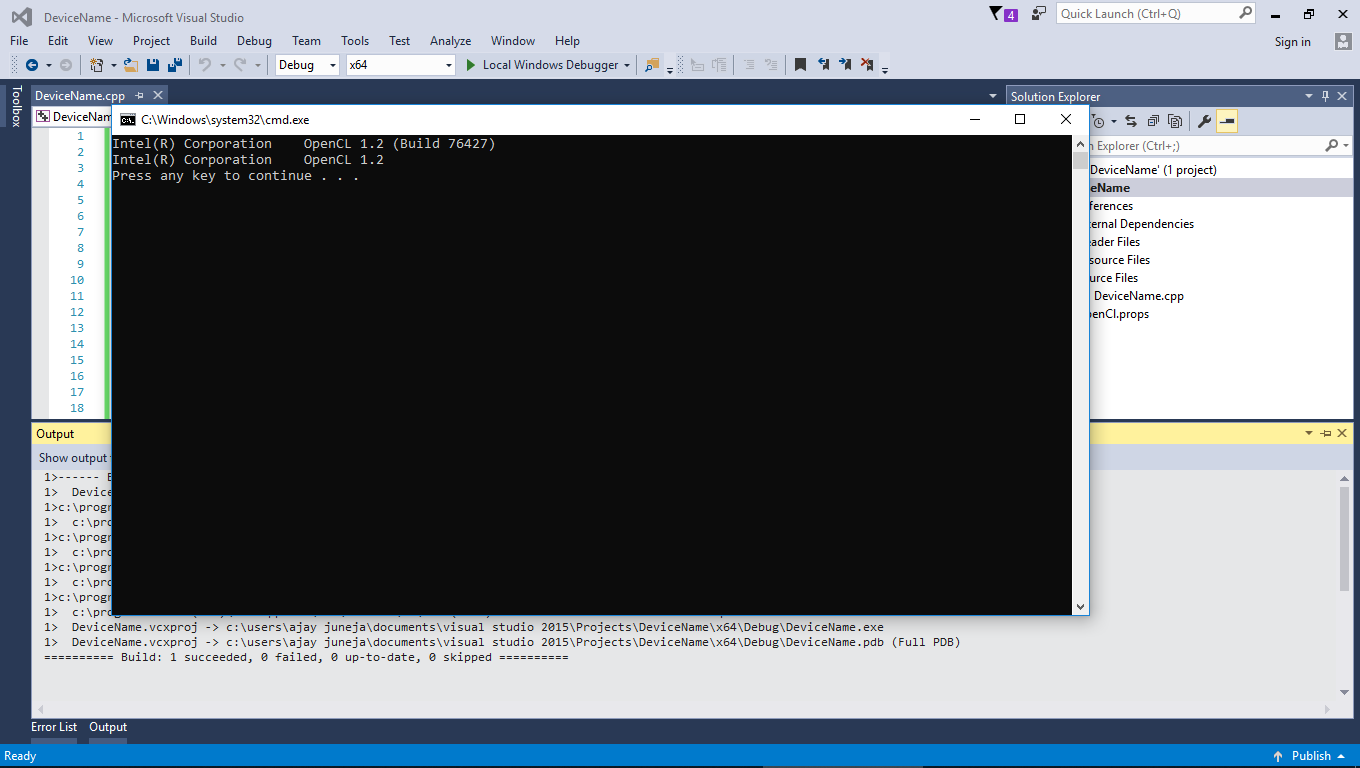
STEP 2: On the side panel, Right click on Source File -> add -> new item -> add

STEP 3: Right Click on list\_devices -> Properties -> C/C++ -> General -> Additional Include Directories -> (Add directory of the include folder in SDK) -> Go to Linker -> General -> Additional Library Directories -> (Add directory of the lib x64 folder which contains OpenCL.lib) -> Linker -> Input -> Additional Dependencies -> Type OpenCL.lib -> On C/C++ set No to SDK Checks. Then OK.

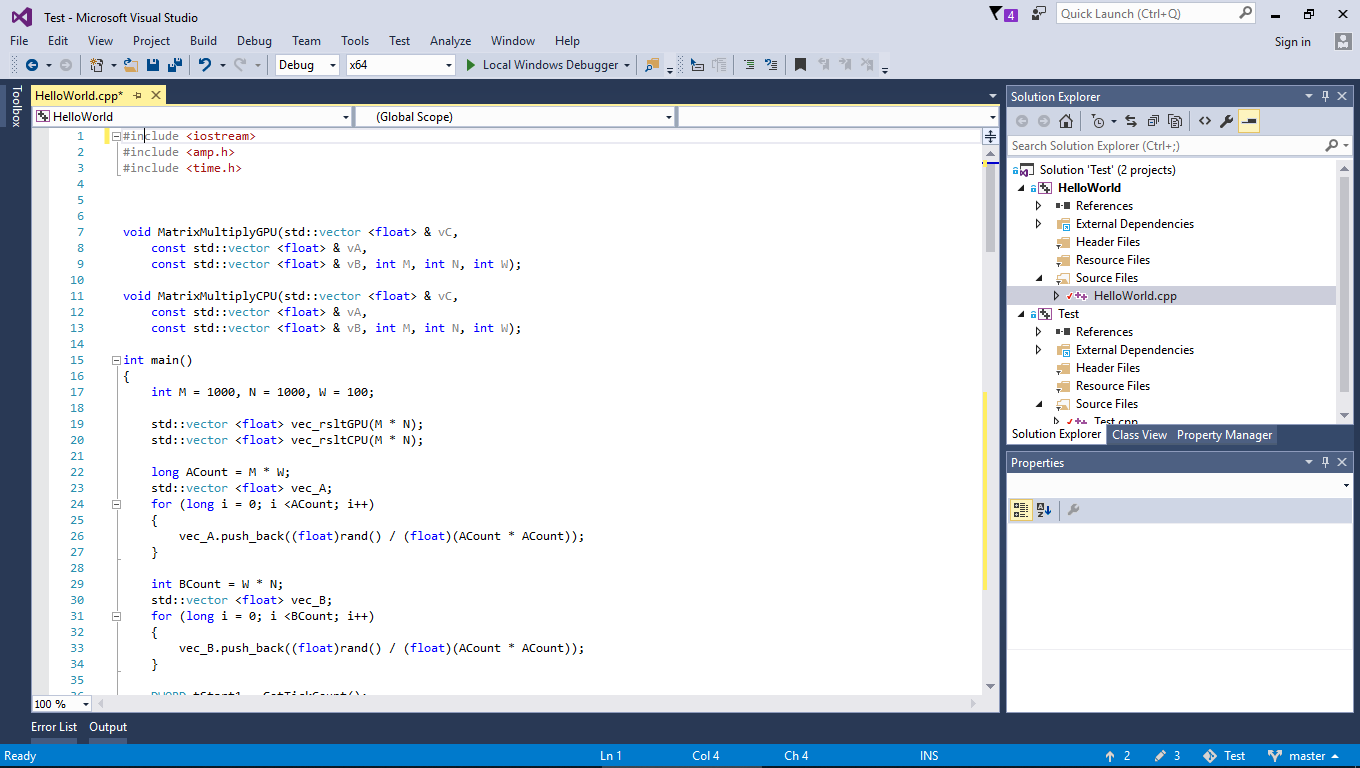
STEP 4: Start doing coding in source.cpp.

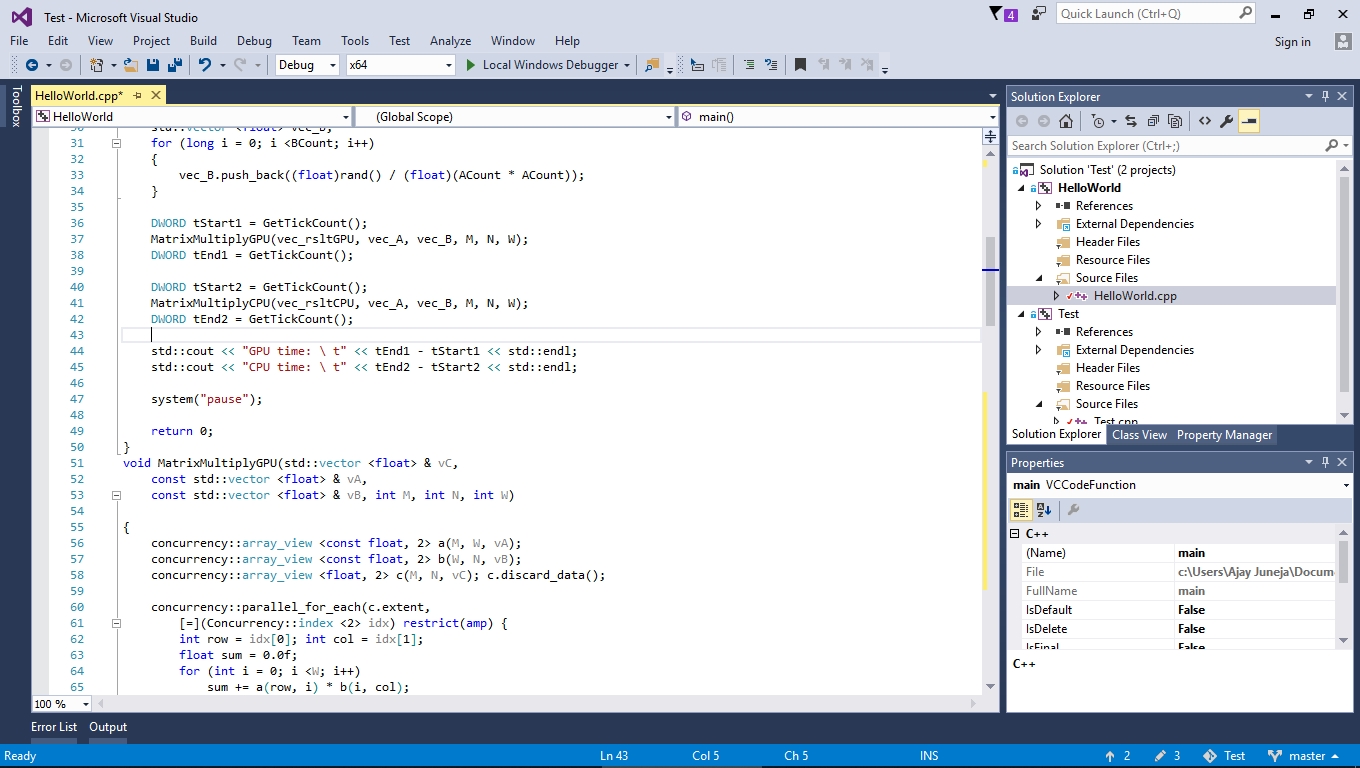
1. **Program which provides list of devices compatible with OpenCL:**

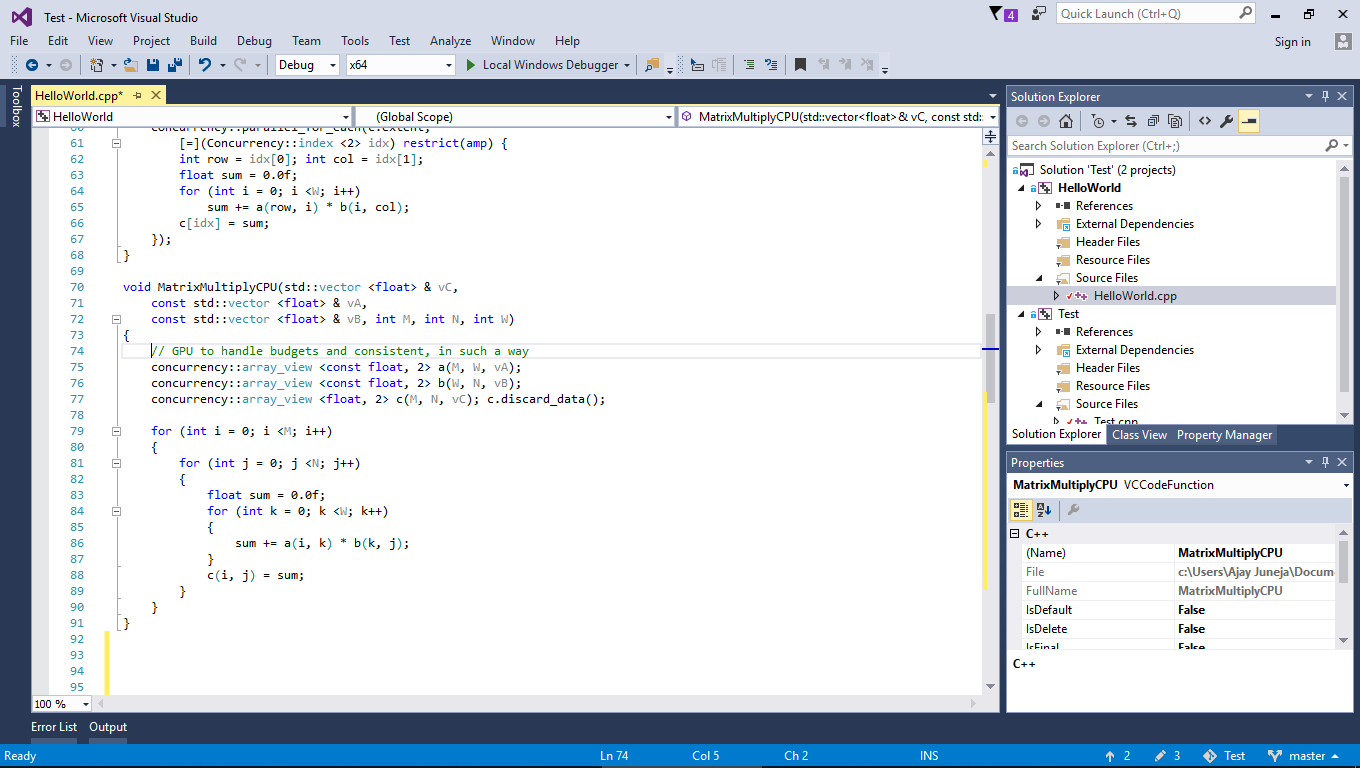


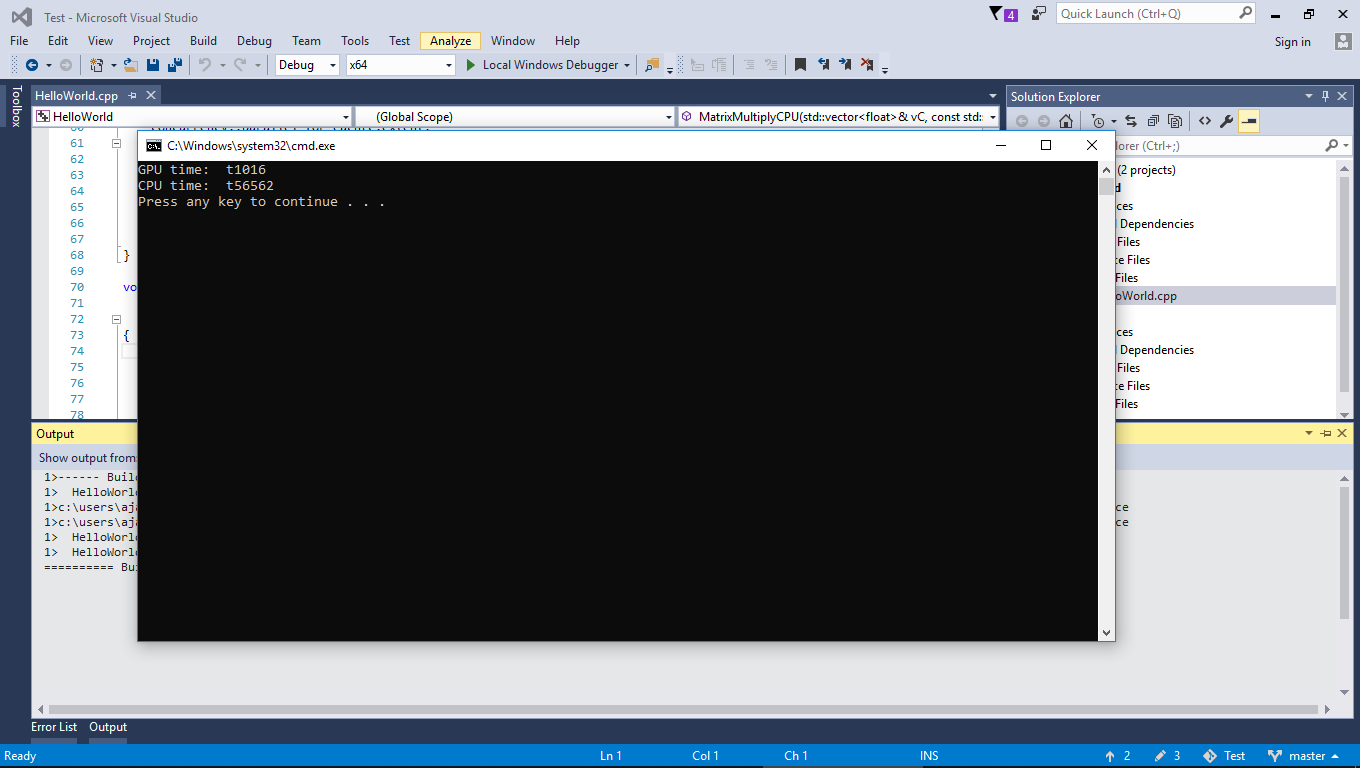


1. **Code to multiply two matrices** once using CPU and then GPU**.**









**Document Details**

|  |  |
| --- | --- |
| **Author** | Ajay Juneja, Amit Chaudhary |
| **Reviewed By** | *Who reviewed this document?* |
| **Approved By** | *Who approved this document?* |