# GUJARAT TECHNOLOGICAL UNIVERSITY

**SUBJECT NAME: Multicore & GPU based Programming**
**SUBJECT CODE: 3715504**
**Semester I**

**Type of course:**

**Prerequisite:**

1. Fundamental of Computing and Programming
2. Computer organization and architecture
3. Microprocessor and Microcontroller
4. Parallel Programming
5. High Performance Computing

**Rationale:**

**Teaching and Examination Scheme:**

| Teaching Scheme | | | Credits | Examination Marks | | | | Total Marks |
|---|---|---|---|---|---|---|---|---|
| L | T | P | C | Theory Marks | | Practical Marks | | |
| | | | | ESE(E) | PA (M) | ESE | PA(I) | |
| 3 | 2 | 0 | 4 | 70 | 30 | 30 | 20 | 150 |

L- Lectures; T- Tutorial/Teacher Guided Student Activity; P- Practical; C- Credit; ESE- End Semester Examination; PA- Progressive Assessment;

**Content:**

| Sr. No. | Content | Total Hrs | % Weightage |
|---|---|---|---|
| **1** | Multicore Programming:<br>Thread Programming constructs will be made available, Thread/OpenMP/MPI/TBB focusing on numerical computations (Numerical Linear Algebra Dense / Sparse Matrix Computations), compiler capabilities for Numerical Kernels & Benchmarks based on PGAS Memory Models ( UPC, & Global Arrays), arallel programs using Pthreads, OpenMP, TBB, Fortran90, MPI 2.0, MPI-Pthreads, MPI-OpenMP on Clusters of Multi-Core Systems, Performance tools (Intel Thread checker, Thread Vtune Analyzer, and Open Source Software tools) on Multi-Core Processor Systems, Numerical Kernels & Benchmarks based on PGAS Memory Models, Numerical Kernels based on MPI & PGAS Memory Models such as UPC, Thread/OpenMP/MPI/TBB focusing on non-numerical computations (Graph Coloring, Sorting algorithms) will be discussed | 10 | 20 |
| **2** | Multi Core Processors - An Overview of Software Multi-threading & Memory allocators & System Overview of threading; An Overview of Programming (MPI 2.0, POSIX Threads, Intel TBB, OpenMP 3.0, Java threading, Charm++), Tuning & Performance, Application Level tuning | 10 | 25 |
| **3** | GPU based Programming:<br>Complier Directives - OpenACC on GPUs, Performance of Matrix Computations, Performance of Application Kernels, CUDA Tool Chain on Multi-Core Processors, matrix computations using Concurrent Asynchronous Execution APIs of CUDA 4.1, Concurrent Asynchronous Execution, LLVM-based CUDA complier and toolkit technologies for matrix computation and application kernels; GPU Accelerator Programming Model - Compiler | 10 | 25 |

| | | | |
|---|---|---|---|
| | Optimizations, Application & System Benchmarks related to HPC GPU Cluster based on CUDA/OpenCL , Tuning & Performance | | |
| | | | |

**Reference Books:**

1. Parallel Programming: for Multicore and Cluster Systems Thomas Rauber

2. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs Shane Cook

**Course Outcome:**

After learning the course the students should be able to:

1. Demonstrate the open MPI/MP/TBB Computations
2. Design Parallel Programming done in Multiprocessing Platform
3. Demonstrate the Knowledge about the kernel Programming
4. Describes the Various application of Multi processors
5. Study on graphical Processing unit Programming