CHAPTER 7

CONCLUSION

7.1 SUMMARY

This thesis has discussed the various real-time scheduling concepts in the beginning the design of real time simulator in Labview and web-based Real time Simulator in Java has been implemented in an extensive manner. Design of Task Scheduler Co-Processor and its hardware setup is discussed extensively. The proposed real time simulator is composed of graphical front end to set task parameters and to display the simulation results capturing important run-time behavior of the computational model of a scheduling policy selected by the user. The results are represented in the form of timing diagram and statistics. This thesis has also explained design and development of real-time algorithms. In web-based real time simulator the user can verify the timing and statistics of his own algorithm using the developed simulator, the code of output modules which are compiled are presented to the user. The user can download these modules from the web architecture and integrate it into his algorithms. To protect the file of the simulator from DoS attack a special architecture is developed which hides the IP address of the systems where the simulator files are present, thereby preventing an hacker from conducting DoS attacks.
7.2 CONCLUSION AND FUTURE WORK

Visual user interface of the real-Time Scheduler Simulator enables a simple definition of simulation conditions task set characteristics, scheduling algorithms characteristics, and the form of result display. This framework is flexible for new algorithm implementation in graphical programming environment. The idea of deploying the Simulator in Web by enabling the Java server pushes the limits of the convention of using any platform-, software-, and machine-dependent simulation tools. The development of real-time scheduler hardware in the name of task-scheduler co-processor for use in conjunction with a target real-time application projected some considerable difficulties such as writing a proprietary real-time kernel in modular structure to be embedded in the target real-time system and development of the task-scheduler co-processor with communication protocols. Due to reasons outlined, the hardware implementation was confined to the development of a scheduler. The hardware setup for scheduler stands as the demonstrator for applying real-time scheduling theory to real-world embedded real-time applications and some important characteristics of real-time schedulers.

The current implementation of the real-Time Scheduler Simulator completes the framework for programming more real-time schedulers for study, research, and teaching. The framework can be extended to support more real-time periodic and aperiodic scheduling algorithms. The gaining popularity in the use of distributed and networked embedded real-time systems in the present-day-world proposes a compelling need to implement multiprocessor real-time scheduling strategies and consideration of more complex task model for simulation and display of statistics closely related to actual real-time system design perspective could improvise the scope of the tool in future. Future work includes the development of real-time
multiprocessor scheduler which is wireless co-processor and a simulator with better web security to prevent web-based attacks. The future simulator must explain all scheduling policies of multiprocessor systems and should also be able to explain the operating system characteristic and real-time characteristic in the system. Another issue that can be focused in future systems is the easy integration of proprietary algorithms into the system.