CHAPTER VII

SUMMARY AND SUGGESTIONS
7.0 Conclusions and Future Scope

7.1 Conclusions

Comprehensive testing of the embedded system is a necessity when testing of Hardware, software and both have to be undertaken using several distinct methods at different testing locations. END-TO-END testing and Regression tests help testing the embedded systems comprehensively.

The need for testing the embedded system and test case types that are required for testing the embedded system have been presented. The classification of test case types required for testing at HOST and at HOST along with Target has been presented. Various Methods are proposed for testing the embedded systems which include scaffolding, Assert Macros, Simulators, Third party tools, in circuit emulators and monitors. The Test case types that must be used with respect to each of the Test methods have been presented. The test case types that are required for a particular Embedded System application, however, have to be selected from the test case types.

Architectural frameworks and models are required for undertaking the testing of embedded systems comprehensively. Various architectural Frameworks and models are proposed which can be effectively employed for undertaking the comprehensive testing of the embedded systems using various test case types.

The following Architectural Frameworks and models have been proposed in the thesis.

1. Architectural Framework for Test Environment
2. Architectural Framework for a Semantic Model
3. Architectural Framework for Process Models
4. Architectural Framework for building test scenarios
5. Architectural Framework for END-TO-END Integration Testing
6. Architectural Framework for Regression Testing

The Architectural framework related to defining the test environment clearly defined the environment required for testing the embedded systems comprehensively.

The Architectural framework for deriving the semantic model which forms the fundamental basis for comprehensive testing of the embedded systems has been presented. The semantic model is presented to have three blocks and the interaction between the blocks has been discussed. The relationships between all the blocks have been explained through a data model. The architecture for semantic model clearly identified the Models that are used for conducting the Analysis, design and development of the Embedded Systems.

The models in the 1st block of semantic model provide the platform required for undertaking comprehensive testing of the embedded system. The models which include Use case, class, sequence and state charts and the relationships with Embedded Application have been presented. The data entities (Class, use case, attributes, functions, statements, events, thin threads, and tasks) required to relate all the models in the first block have also been explained.

The models presented in the 2nd block of semantic model provided the basis for identification of testing requirements, the END-TO-END integration testing requirements model through the definition of the Thin threads, Data models that relate the analysis, design and development of thin threads, the semantic model itself, and the Regression testing model required for undertaking the regression of the embedded systems. This block also explained the relationship of test case types with thin threads and test methods. The test case generation process has also been explained.
The models presented in the 3rd block of semantic model are related to the process models required for actual testing of the embedded systems. The block also describes the test configuration required and the processes required for undertaking the test coverage and testing Analysis.

Architectural Frameworks to build the process models required for undertaking the testing at each of the locations have clearly identified the sequence of processing undertaken right from compilation stage to the test case generation and testing of the embedded system. All integration aspects to integrate the Hardware, Software, test gadgets and Production system have also been presented.

The Scenario based architectural models required for building the Test case requirements from the end user perspective have been presented. The building of scenarios has provided the basic framework for identifying the testing requirement from the perspective of any given application. The process of construction the complex scenarios from simple scenarios have also been presented. The scenarios are linked to the thin threads as the thin threads provides for END-TO-END testing of the embedded system. The scenarios are related to each other and to thin threads and the relations are presented in the data model. The data model presented the framework for undertaking the testing of the embedded system dynamically and using the repeatable tests when ever required.

The architectural models required for undertaking the END-TO-END (E2E) Integration testing have been presented. The pre and post conditions required to undertake the E2E has been defined through patterns. Various types of patterns are used and the required input and output parameters for E2E testing of the embedded systems have also been explained. The tree structures which are used to build the thin threads and the patterns have also been explained. The linking of the thin threads with patterns through
trees has been explained. Every scenario has been mapped to a thin thread and scenarios are tested by undertaking the testing of the thin threads. The thin threads are mapped to the test methods and test case types to methods. Test case generation process for testing the thin threads has been presented.

The architectural framework required for carrying the regression testing of the embedded system has been presented. The tracing of the changes across the entire application and conducting of the ripple effect analysis has been presented. The process of identifying the thin threads that are affected due to the changes undertaken has been explained. Any thin thread that has affected components in it is proposed to be tested and this will guarantee the comprehensive testing of the embedded systems.

A Pilot project has been designed, developed, fabricated and presented. The pilot project is related to Temperature monitoring and control of Nuclear reactor systems (TMCNRS). The need for monitoring and controlling of the Nuclear reactor for the proper enrichment of the uranium has been presented. The pilot project is selected considering different real time requirements and also controlling the actions taken due to the events taking place in the external environment as consequence of changes taking place within the temperatures of the fuels contained in the Nuclear reactor tubes.

The pilot project is analyzed, designed and developed with the primary objective of testing the Embedded Application using the architectural frameworks and models.

The mechanical setup which simulates the Nuclear reactor has been presented. The development, fabrication and the arrangement of various components which include (Sensors, Pumps, Vertical tubes, coolant troughs, power supply system etc.) has been explained.
The analysis and design of embedded system Hardware has been presented. Various types of chips and their integration, the communication interfaces used to facilitate the communication with the software has been presented. All the chips used are described in terms of the features, principle of operation, and the timing diagrams wherever required.

The analysis and design of the software related to the pilot project is presented. The real time software architecture used for the development of the application has been presented. The development platform used has also been explained.

The user interface used for the HOST to interact with the Target has also been explained. The integration of Software with hardware has been presented.

Even though the analysis and design have been conducted using the Object oriented approach, the same are translated into procedural constructs due to the limitation of the development environment. For this project KEIL 2.0(An Integrated Embedded Systems Development environment) is used as development platform.

The code that resides at the target and the HOST is presented. The HOST based code helps in providing the control data to the target and the target software uses the reference data as control data to control the changes in the temperatures.

The host based application provides for the animations of the production system for the operator to visualize the operations and the process taking place at the target.
Thus, the architectural models provided the basis for comprehensive testing of the embedded systems. The models help in conducting different types of testing and help in generation of the test cases automatically.

A central repository has been defined and populated with all the relationships details. The process of automatic generation of the test cases has been explained.

All the test case types that help comprehensive testing of embedded system are mapped to the thin threads and the test cases required for undertaking testing through each of the test methods have been presented.

The repository is also populated with the analysis and design elements, such as statements, functions, tasks, data variable types, data variables, sequences of function calls etc. The relationships between all the design elements have also been built and populated into the database.

The modeling of test case scenarios for the pilot project using scenario based architectural model has been presented. The scenarios identified are populated into the repository. The thin threads for the pilot project have been identified and populated. The patterns that relate to the Pilot project are identified and mapped to the thin threads. The pattern related Pre and Post conditions for the pilot project have been identified and built into repository. The relationships of thin threads with Tasks, functions, methods that relate to the pilot project have been identified and built into the central repository.

Thus the central repository is built for the pilot project based on the architectural frame works and test generators are used to generate the test cases that should be used for testing the embedded system at different locations and using different test methods.
Testing process models for conducting the end to end testing and Regression testing of the embedded system have been presented. The process models are used to undertake the testing of the embedded systems. The test cases generated are used to carry testing of the pilot project and the test results are presented. The test results are presented in respect of each of the thin thread and a testing method separately.

7.2 Future Scope

The Architectural Frameworks and models discussed in the thesis can be extended to implement remote testing of the embedded system, where Syntactic and Semantic evolution of the embedded systems are required. Dynamic testing of embedded systems can be undertaken when changes to the embedded systems are to be dynamically implemented and online testing of the embedded systems is required without actually shutting down either the embedded system or production system.

A single tool can be developed to carry comprehensive testing of the embedded systems instead of using many tools such as Simulators, Third party tools, monitors and emulators. Such a tool may have the built-in interfaces to all the test gadgets. In this case the necessity for building too many command language interfaces can be eliminated.