MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CENG 492 – Computer Engineering Design II

Test Specifications Report

Group Name: SOSOFT

Group Members:

Hatice Beyza KIRBAŞ	1297951
Özgür Ufuk ÖZDEMİR	1347798
Oya TEZEL	1348010
Serdar TUĞCU	1298330

Project Title: Emulator and Development Environment for CENG Embedded System Card

Project Alias: PicSim

Table of Contents

1. INTRODUCTION

1.1. Goals and Objectives

The project PicSim is a simple development environment used for pic programming on PicDev Board in Ceng336 course. This simple environment has features for writing c source code, editing a written code, compiling, assembling, and debugging the code, programming the pic on the board with the hex file produced and simulating the board. We have to finish implementing all this modules in a short time period and integrate them with each other. This integration depends on the communication of each module with the others. For this reason testing and achieving a bug-free product is of great importance for our project PicSim.

1.2. Scope of the Document

In fact, the testing process has started with the first module of project we have implemented, but this document also includes some information about how the testing has been done until now. However the main aim and scope of this document is to present the testing plan for the remaining parts of the project.

It is important to emphasize that details our test plan is decided considering the specific properties of our project.

1.3. Testing Plan Scope

Our testing process includes unit testing, integration testing, performance testing, alpha and beta testing.

- **Unit testing:** We will test compiler, debugger, simulator and editor modules separately.

- **Integration testing:** Since in PicSim the main integration issue is between the user interface and main modules, will test the integration between the interface and compiler, debugger and simulator modules only. This modules are not related with each other due to our implementation, so we do not need to test integration of these modules with each other.

- **Performance testing:** This will be done mainly for the simulator module in which we try to make reasonable delay in order to achieve the same execution speed with a simple pic. Since our processors in PCs are much faster than a pic, we have to care about timing.
- **Black Box testing:** This part includes the testing of PicSim without caring about internal working of items in the project.

- **White Box testing:** This part is the testing of PicSim by project members who has implemented the corresponding part.

1.4. Major Constraints

1.4.1 Time Constraints

Time is the main constraint for us in completing PicSim with a bug-free version. We have nearly one month for last release to be finished and we have many tests as well as incomplete implementations of modules to be done. We expect at least finishing the unit and integration testing at the same time with the implementation process of main modules.

1.4.2. Efficiency of tests

Since we do not have much time and we have many small jobs in some modules, especially in the simulator module, and finding possible bugs in black box testing is easy; we will not consume much time in unit testing of this module. We need many test programs implementing which is time consuming, so we will test these small jobs all together.

1.4.3. Staff

The project group is composed of 4 people. Since the group will also be working on debugging and development, the staff is also a major constraint for the tests of our project.

2. TEST SCENARIOS

2.1. PicSim Compiler

PicSim uses the open source compiler sddc (small device c compiler) for the compiler module. Main issues to test in this module are:

- If it gives the appropriate outputs (i.e. hex, cod, asm files etc.)
- On error if it states what the error is correctly.
- If it gives the desired warnings.

2.2. PicSim Debugger

This module is related with both simulation and compiler modules as well as the main window of the user interface. So we will test this module in two phases: watch test and break point test.

2.2.1 Debugger Watch Points

PicSim Debugger watch point sub-module is used for monitoring the contents of register in pic data banks and user defined variables. Since both simulator module and debugger module uses the same register contents, it is very critical to have the right values in those registers. So the tests we will do in this part are the most important ones. We will monitor these values step by step using another debugger (MS Visual Studio 2005 C++ debugger) in order to achieve a bug-free version of this module.

2.2.2 Debugger Break Points

PicSim also has the feature of putting break points in the given line number of the code. We will also make unit testing on this sub-module which is integrated with the user interface module.

2.3. PicSim Simulator

We expect most of the bugs will be in this module. This is not because we did not care about this module, but the board we try to simulate has a pic, which is a small and simple device when compared with a desktop computer. It has a very slow processor. We have to estimate the time it works and try to show our simulator as if it works in the same speed with this device. This situation causes the implementation of this module become complex. Bug tracing in such a complex module is harder than in the other modules, so we will test this module by testing all the special purpose registers (especially the pins) as output, as input, as digital or analog signal etc. separately.

2.4. PicSim Upload Module

This module is used for programming the pic of PicDev via the parallel port of the computer with the program which is the hex file output of the PicSim compiler. This test again includes testing of the errors occurred, testing if it gives the appropriate programming messages etc.

3. TESTING TOOLS and ENVIRONMENTS

Microsoft Visual Studio .NET 2003 Debugger

Gdb

Qt and Kde library tools (for writing test codes)

4. STAFFING and RESOURCES

For testing PicSim we need a pc, a parallel port cable and the PicDev board as resources.

The staff responsible for the tests of parts of PicSim:

Compiler Module and Integration with UI	Oya Tezel
Debugger Break Points	Beyza Kırbaş
Debugger Watch Points	Serdar Tuğcu
Simulator pins	Ufuk Özdemir, Serdar Tuğcu
Simulation Timing	Oya Tezel
Upload	Beyza Kırbaş, Ufuk Özdemir

5. SCHEDULE

Jobs to Finish	**Deadline (expected)**
Test Plan Delivery:	6th of May 2007
Integration Tests:	8th of June 2007
Performance Tests:	~ 1st of June 2007 (is not certain since we have not finished implementation of simulator which is the main performance criteria)
Black Box Testing (Compiler and upload outputs)	12th of May 2007
White Box Testing (Simulation module)	8th of June 2007
Results (i.e. Bugs) Tracing and Correction:	10th of June 2007

6. APPENDIX

Test and Bug Report Form

Who made the test?
Which module was tested?
When did the test finished?
Which bugs are found?
What corrections/changes were made?
Comments:

Test Checklist

Module Name	**Test to do**	**Finishing Date**
Compiler	Integration with other modules	
	Giving errors	**Done(05.05.07)**
	Giving warnings	**Done(05.06.07)**
	Giving output files	**Done(February)**
Debugger	Integration with other modules	
	Watch points	
	Break points	
Simulator	Integration with other modules	
	Leds	
	Seven segment displays	
	LCD	
Upload	Integration with other modules	
	Giving correct error messages	
	Giving correct warnings and messages	