SOFTWARE TEST PLAN
Outline and Requirements

General Comments: *The following guidelines are given in italics and reporesent the base minimim required to complete this plan. Non-italicized text is to be included in your plan verbatim. All sections must be included in your plan, even if you do not feel it applies.*

1. Scope of Testing
The general scope of the test plan is given in this section. Summarize the specific functional, performance, and internal design characteristics that are to be tested. Briefly describe the bounds of the testing, criteria for completion of each test, and schedule constraints.

2. Test Plan
Testing is divided into phases and builds that address specific functional and behavioral characteristics of the software. Each of these builds is a group of modules which are created in a phase.

2.1. Test Phases and Builds
Describe each software development and test phase, and describe the build process to implements that phase.

2.2. Schedule
Provide estimated start and completion dates for integration, overhead software development to support test (see next section), test application periods, and test analysis periods (if applicable) for each test phase. Note that each of these development of test periods imposes an availability of unit tested modules to integrate and test.

2.3. Overhead Software
Overhead software consists of software test benches, stubs, and drivers necessary to carry out the testing. Describe these overhead items.

2.4. Environment and Resources
Define the normal test environment including hardware platforms or external resources, and software tools necessary to conduct the tests. Also define any unusual hardware configurations, exotic simulators, special test tools or techniques.

3. Test Procedure: Build *n*
This section prsents the detailed testing procedure required to accompish the test plan (Section 2 above). Separate section 3's containing the unit and integration tests are defined for each build.

3.1. Order of Integratiion
Briefly describe the order of unit and subsystem integration to be carried out to create the build.

3.1.1. Purpose
State the purpose of the build and the testing of the build.

3.1.2. Modules to be Tested
Identify the modules in this build to be tested and integrated in the order they will be tested.

3.2. Unit Tests for Modules in Build
This section describes the actual test for each module m. Briefly introduce the test here. There are m Section 3.2's, one for each module being tested.

3.2.1. Description of Tests for Module *m*

Describe each test using a formal point-by-point statement of each test.

3.2.2. Overhead Software Description

Describe all software items such as stubs, drivers, and test benches to check out this module.

3.2.3. Expected Results

Identify the expected results of the test. Write this section in a list fashion such that each test motivates one or more output descriptions.

3.3. Test Environment

3.3.1. Special Tools or Techniques

Identify all special hardware or software tools or techniques necessary to carry out this test.

3.3.2. Overhead Software Description

Describe the overhead software (.i.e., stubs, drivers, test benches) at a level detailed enough to support coding the item.

3.4. Test Case Data

Identify all test data to be used to conduct the test.

3.5. Expected Results for Build *n*

Identify the expected results for each input test data case. Note test case data and expected results can be described in a singe section mainly consisting of large table where test case data is presented on the left side of the table, and expected results shown on the right side of the table.

4. Actual Test Results

This section consists of forms with blank spaces to record the actual test results along with the date of the test, who the tester is, etc. Again, this section can be integrated with Section 3.4 and 3.5 for each build as an option.

5. References

Identify all documentation references where supporting material has been used.

6. Appendices