

# Verification and Validation Plan

## Team Groundhog Spring 2009

### 1. Introduction

#### *1.1 Purpose of this document*

The purpose of this document is to describe the process and methodology used to determine if the software is functioning as intended and designed.

#### *1.2 Scope of the development project*

The application is a Windows based .Net implementation with OpenGL. Our main UI will be developed in C#, which is a .Net language. This application will allow viewing of GDS II files, along with printing on an Epson Large Format plotter.

#### *1.3 Definitions, acronyms, and abbreviations*

- Alpha Blending - Convex combination of two colors allowing for transparency effects in computer graphics.
- GDS II: database file format which is the de facto industry standard for data exchange of integrated circuit or IC layout artwork
- OpenGL – The industry's most widely used, supported and best documented 2D/3D graphics API
- SRS - Software Requirements Specification
- UI - User Interface
- VLSI – Very Large Scale Integration

#### *1.4 References*

[http://en.wikipedia.org/wiki/Alpha\\_compositing](http://en.wikipedia.org/wiki/Alpha_compositing)

<http://en.wikipedia.org/wiki/VLSI>

<http://en.wikipedia.org/wiki/GDSII>

<http://www.opengl.org/documentation/>

#### *1.5 Overview of document*

The Verification and Validation Plan is a document outlining the testing, structure, and verification of the software. In order to further verify the software, each component of the system will be thoroughly tested on its own, and while operating as part of the larger software system.

### 2. Reviews, Walkthroughs, Inspections, and Audits

As a development team, we will meet every week. During these meetings, we will assign not only development of components to individual members, but also Quality Assurance of components of other team members. The Quality Assurance tests may be performed by a single member of the team, or together as a group.

These Quality Assurance tests will consist largely of Code Viewing, and Unit Test writing. From these Reviews, the reviewer may request additional Walkthroughs, Inspections, or Audits as necessary.

<Insert here> TWSS

### **3. Component test plans and procedures**

#### ***3.1 Component Test Strategy Overview***

Testing Process: Process by which the software will be tested.

Test frequency: How often test is required.

Components tested: Detailed description components being tested.

Test recording procedures: Description of testing records that will be maintained.

#### ***3.2 GDS Parser***

Testing Process: Verify that the objects being created are correct. Also check that all the needed fields are being parsed.

Test Frequency: Every build.

Components Tested: All subcomponents of GDS Parser

Test Recording Procedures: Document any outstanding/undocumented bugs. Add any relevant comments to existing bugs.

#### ***3.3 User Interface***

Testing Process: Verify that all menus are working correctly. Verify that controls are being created and showing appropriate information.

Test Frequency: Whenever a new feature is added to the UI.

Components Tested: All subcomponents of User Interface Test Recording Procedures: Document any outstanding/undocumented bugs. Add any relevant comments to existing bugs.

#### ***3.4 Drawing Manager***

Testing Process: Verify that the objects being drawn are correct dependent on the GDS file input. Verify alpha blending and draw order are correct.

Test Frequency: Every build.

Components Tested: All subcomponents of Drawing Manager

Test Recording Procedures: Document any outstanding/undocumented bugs. Add any relevant comments to existing bugs.

### ***3.5 Print Module***

Testing Process: Verify that the application can print to any printer given the driver. Verify that the application can print to the desired number of pixels.

Test Frequency: Every build.

Components Tested: All subcomponents of Print Module

Test Recording Procedures: Document any outstanding/undocumented bugs. Add any relevant comments to existing bugs.

## **4. System test plans and procedures**

### ***4.1 System test strategy overview***

Testing Process: Process by which the software will be tested.

Test frequency: How often test is required.

Components tested: Detailed description components being tested.

Test recording procedures: Description of testing records that will be maintained.

### ***4.2 System test description***

Testing Process: Verify that a given GDS file is parsed correctly through the file dialog provided by the UI. Verify that the drawn GDS objects are correct in relation to each other and are showing the user chosen color, alpha, and any other preferences they'd like. Make sure that the application can print the GDS that is being presented on screen.

Test frequency: Every build.

Components tested: GDS Parser, User Interface, Draw Manager, Print Module

Test recording procedures: Encountered bugs will be recorded for review/fix later.

## 5. Defect tracking plans

We will be utilizing BugZilla to track all discovered defects, which will be hosted on the same server as our source control. Scott will act as the BugZilla Admin.

Priority will range from P1 (blocker - program fails), P2 (medium - nuisances that do not affect functionality), and P3 (enhancements).

The status can be under review (logged but not yet discussed by the team), in progress (delegated to a team member who is working on it), fixed (team member has resolved the issue), or closed (fix has been verified by a team member other than he who fixed it).

The description will provide information about what the problem is and how to duplicate the issue. Additional comments will provide an area to add any information that is deemed beneficial to resolving the issue.

## 6.0 Traceability from SRS to SDS

Requirement	Specification
<b>3.2.2 GDS File Selection UI</b>	<b>3.2 GDS Parser</b>
<b>3.2.3 Main UI</b>	<b>3.3 User Interface</b>
<b>3.2.4 Color selection dialog</b>	<b>3.4 Drawing Manager</b>
<b>3.2.5 Print dialog</b>	<b>3.5 Print Module</b>

## 7.0 Test-requirements cross-reference matrix

The nature of our project is somewhat subjective because each person has a different opinion regarding what is a "user-friendly" and functional UI. Some of our test acceptance is based on our personal opinion and the opinion of our sponsor. When possible test acceptance will be based on functional vs. non-functional.

Identifier	Description	Test Means	Acceptance	System Functionality
<b>3.2.2 GDS File Selection UI</b>	Provide a simple interaction for opening GDS files	black box	Files are accessible, and .gds are available	Open File dialog
<b>3.2.3 Main UI</b>	Correctly display GDS as defined	black box	All elements in the GDS are displayed	Main UI Screen
<b>3.2.4 Color selection dialog</b>	Allow for selection of color/alpha by layer	black box	Color/Alpha selection is represented in the UI	Color/Alpha interaction

<b>3.2.5 Print dialog</b>	Allow for printing to installed printers	black box	Print dialog opens, and job is created on printer	Print dialog
---------------------------	--	-----------	---	--------------

## **8.0 Acceptance test and preparation for delivery**

### **8.1 Procedure by which the software product will be acceptance tested**

Software will be thoroughly tested by our team. Since this is primarily a User Interface, we will not be leveraging any automated test software. Our Customer will be the main acceptance tester, and will provide all feedback on features and defects.

### **8.2 Specific acceptance criteria**

All must have requirements are in place, and easily accessed within the UI.

The end customer has signed off on all must haves.

No open P1/P2 defects are present.

Less than 10% open P3s.

### **8.3 Scenario by which the software product will be installed**

Our application will be bundled into a Microsoft Installer package that can be downloaded and run on any Windows based machine. The .Net 3.5 prerequisite will also be bundled to ensure that the system has all necessary libraries present.