

# Verification & Validation Results 3

Team Groundhog

Apr 17, 2009

## 1.0 Introduction and overview

- **1.1 Purpose of this Document** The results of the VVP are presented here in the VVR, which outlines our testing results Groundhog GDSViewer.
- **1.2 Scope of the Development Project** Currently there are many GDS Viewers available, but none of them do realistic blending of different layers and therefore can not identify hierarchy. Our goal is to create a GDS Viewer that uses alpha blending rather than stippling patterns to clearly delineate hierarchy.
- **1.3 Definitions, Acronyms, and Abbreviations**
  - alpha blending – a convex combination of two colors allowing for transparency effects
  - GDSII – a database file format which is the de facto industry standard for data exchange of integrated circuit or IC Layout artwork
  - OpenGL - a standard specification defining a cross-language cross-platform API for writing applications that produce 2D and 3D Computer Graphics
  - UI - user interface.
- **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/Software\\_Requirements\\_Specification](http://www.opengl.org/documentation/Software_Requirements_Specification)
  - [http://www.opengl.org/documentation/Software\\_Design\\_Specification](http://www.opengl.org/documentation/Software_Design_Specification)
- **1.5 Overview of Document** This document describes the testing results, procedures, and fulfillment of requirements of the project.

## 2.0 Summary of Results

We thoroughly tested each added component prior to committing the code to the repository, and following the commit as well. We found in all such cases that we were able to ensure that the code was functioning just as anticipated and the changes did not cause any negative repercussions. Optimization has been the main focus following integration. Independently, all of the portions are working well, but there are some issues to iron out between the various components.

For Release 2 we have resolved all but two remaining issues. The Draw manager is still being optimized, with a deliverable of next week. For Release 3 we will have the necessary drawing speed, along with the ability to store and draw thickness on the layers. 3D rendering is still a stretch goal, and may not be possible due to the optimizations needed for the 2D drawing.

For Release 3 we have updated the Draw manager to use Display Lists, with a minor improvement in draw speed. We have spent extra time making the UI more user friendly and adding some nice to have features. There is now a default color profile that loads, and stays between files. A new Structures and SREF list will allow more flexibility in drawing and printing complex GDS objects.

### 3.0 Results from Reviews, Walk-Throughs, Inspections, and Audits

We held weekly meetings with code reviews and feature validation. Each group lead presented their progress and raised any issues that they were having. This allowed the team to provide input on the variety of issues that were raised.

Audits were performed on each of the layers.

- The UI interactions were presented and ideas exchanged on best practices and standard behaviors.
- The drawing manager was reviewed thoroughly and optimization was raised as a major concern. We quickly got through the basic functionality of the OpenGL framework.
- Printing was presented with a PDF printing plugin. We were able to quickly test and verify the quality of the image and demonstrate that the alpha blending worked correctly.

Unit tests were run against the variety of GDS files that were provided by our Customer. The main goal being to ensure that we were able to properly display each of the variety of structures that are represented in the GDS format.

Bi-Weekly walkthroughs were conducted with Prof. Brunvand, so that we were certain that we provided the correct interaction model for our end user. It's easy to get lost in how the code works and miss how the application should work.

We added a meeting with the other team that is doing a GDS viewer for their project. We also spent time with an Open GL expert to see if there was anything obvious that we can do to speed up our drawing. Both meetings were productive and gave us good feedback on what we needed to focus on. Overall our UI was more complete than the other teams', but they were able to draw the files much quicker.

### 4.0 Results from Testing

#### • 4.1 Summary of Component Test

- GDS Parser: Tested by Josh, minimum overall changes were done on the GDS Parser, and the changes that were done were tested quite easily.
  - No Open Bugs
- Drawing Manager: Tested by Tim, components were all sufficiently tested, there were no adverse effects.
  - **Bugs detailed in section 6**
- UI: Tested by Curtis, most components and features were tested and function properly.
  - No Open Bugs
- User Preferences: Tested by Adam, XML parsing and writing were tested and there are no outstanding issues in that portion.
  - No Open Bugs
- Print Module: Tested by Scott. All tests indicated normal functionality. There are no current bugs for this item
  - No Open Bugs

#### • 4.2 Summary of Integration, Test/Testing Product as a Whole

Each separate component was tested individually prior to and after integration into the application.

### 5.0 Evaluation of the Process

- **5.1 Evaluation of Test Cases**

The test cases we have been using have been quite sufficient and helpful in ensuring that our code is functioning correctly.

- **5.2 Results from Bugzilla**

To this point in our project using Bugzilla has served primary as a tool for keeping track of current issues, and keeping everything organized.

- **5.3 Lessons Learned**

Working on a project with multiple contributors requires constant communication and careful planning. Frequent updates and meetings with the customer is very useful in keeping the group focused, and ensuring a quality product.

## **6.0 Outcome of Acceptance Tests and Delivery**

All must have requirements have been presented to Prof. Brunvand. No major issues have been raised, and additional nice to have requirements have been added. The majority of feedback received related to how he would like to see the tool behave for people that are more familiar with GDS.

- Must have requirements
  - Parse GDS files provided, with the goal of handling any GDS (Implemented)
  - Draw GDS structures in correct layers with colors specified by end user (Implemented)
  - Apply Alpha Blending on layer overlap (Implemented)
  - Allow for printing to any printer, mainly a large format plotter (Implemented)
  - Zoom/Move on UI for GDS object (Implemented)
  - User Preferences for assigning color, name, and position of layers (Implemented)
- Nice to haves
  - Track layer thickness and implement in drawing (Implemented)
  - Allow for 3D view of GDS object (postponed)
  - Structure List in UI(Implemented)
  - SREF Tree in UI (Implemented)
- Bug Tracking
  - P1 – Rendering Large GDS Structures cause memory issues (Resolution in progress)
    - Investigation of using Vertex Object Buffers for more speedy loading
  - P1 – SREFs not drawing (Resolved)
  - P1 – AREFs not drawing (Resolved)
  - P2 – SREF/AREF rotation needs review (Resolution in progress)
    - Bounding box is not calculated correctly
    - Rotation in 8x8 SRAM GDS needs to be corrected
  - P2 – Preferences UI doesn't updated with main UI (**Resolved**)
    - Shared data model between UI elements
  - P3 – Preferences UI needs OK/Cancel functionality (**Resolved**)

- Acceptance test bug
  - P3 – Drag/Drop functionality not updating drawing (Resolved)
  - P2 – Print preview not based on resolution of printer (Resolved)

P1 – 1 remaining/2 closed

P2 – 1 remaining/2 closed

P3 – 0 remaining/3 closed

## **7.0 Summary of open issues**

Rendering large-scale chips needs to be optimized.

No open feature requests